D O C T O R A L T H E S I S

Mathematical Models and Optimization Algorithms for Sustainable Operations in Manufacturing and Services

University of Modena and Reggio Emilia
Department of Economics "Marco Biagi" - Marco Biagi Foundation
Doctoral course in Labour, Development, and Innovation
coordinated by Prof. Ylenia Curzi

Cycle XXXVI
SSD: MAT/09 - RICERCA OPERATIVA

Candidate
Giulia Caselli

Supervisor
Prof. Manuel Iori

Co-Supervisor
Prof. Carlo Alberto Magni

Academic year 2022/2023

Abstract

This PhD thesis addresses complex decision-making and management problems emerging in some real-world manufacturing and service business contexts. It considers economic, environmental, and social aspects of optimization, with the objective to devise effective solution methods for the problems at hand and apply them to solve real-world instances. The aim is to contribute to the advancement of the state of the art in Sustainable Operations, a recent interdisciplinary field that employs quantitative methods from Operations Research to achieve economic, environmental, and social sustainability in business, by solving real-world applications of increasing complexity, starting from mono-objective problems, then moving to bi-objective and finally bilevel problems.

The first part of the thesis explores applications within the traditional setting of a single decision maker with a single optimization goal. Mixed Integer Linear Programming (MILP) is applied to formulate and solve in an exact way two optimization problems in the educational and healthcare sectors. Additionally, exact decomposition approaches are used to solve a scheduling problem in a test laboratory environment, and machine learning methods are employed to forecast demand in an environmentally impactful production planning application. The second part of the thesis is devoted to bi-objective optimization, which allows to make a trade-off between conflicting goals of a single decision maker. Facility location for waste management and equipment replacement problems for energy transition are studied under the bi-objective setting, incorporating both economic and environmental goals. By the iterated use of MILP models, it is shown how decision makers can simultaneously consider costs and carbon emissions. In the third part of the thesis, bilevel optimization is explored. Bilevel optimization allows to model optimization problems with multiple decision makers in a hierarchical two-level relationship. This appears in many realistic situations involving, for instance, government and private businesses, or suppliers and manufacturers of the same supply chain. We aim to comprehensively review the existing literature on bilevel optimization with sustainable perspectives. Overall, this thesis showcases the potential of Operations Research in achieving sustainability even in very complex decision-making settings, solving several real-world applications with a variety of optimization methods.

Keywords: Sustainability; Mixed Integer Linear Programming; Machine Learning; Optimization; Real-World Applications

Abstract (in italiano)

Questa tesi di dottorato affronta problemi decisionali e gestionali complessi che emergono in contesti reali di imprese manifatturiere e di servizi. La tesi esamina aspetti economici, ambientali e sociali dell'ottimizzazione, con l'obiettivo di sviluppare metodi di risoluzione efficaci per tali problemi e applicarli per risolvere istanze reali. Lo scopo è contribuire all'avanzamento dello stato dell'arte nell'area di ricerca cosiddetta delle "Operazioni Sostenibili", un recente campo interdisciplinare che impiega metodi quantitativi della Ricerca Operativa per perseguire la sostenibilità economica, ambientale e sociale nelle imprese, risolvendo applicazioni reali di crescente complessità, partendo da problemi mono-obiettivo, passando ai problemi bi-obiettivo e infine a quelli bi-livello.

La prima parte della tesi esplora diverse applicazioni nel contesto tradizionale di un singolo decisore con un singolo obiettivo di ottimizzazione. La Programmazione Lineare Intera Mista (PLIM) è utilizzata per formulare e risolvere in modo esatto due problemi di assegnamento nei settori dell'educazione e della sanità. Inoltre, si applicano algoritmi esatti di decomposizione per risolvere un problema di schedulazione in un laboratorio di testing e metodi di machine learning per prevedere la domanda in un'applicazione di pianificazione della produzione con elevato impatto ambientale. La seconda parte di questa tesi è dedicata all'ottimizzazione bi-obiettivo, che consente di bilanciare obiettivi contrastanti di un singolo decisore. Si studiano problemi di localizzazione di impianti per la gestione dei rifiuti e di sostituzione di veicoli per la transizione energetica nel contesto bi-obiettivo, incorporando obiettivi sia economici che ambientali. Attraverso l'uso iterato di modelli PLIM, si dimostra come i decisori possano considerare contemporaneamente costi ed emissioni di carbonio. Nella terza parte di questa tesi, si esplora l'ottimizzazione bi-livello. L'ottimizzazione bi-livello consente di modellare problemi di ottimizzazione con molteplici decisori coinvolti in una relazione gerarchica a due livelli. Questo si verifica in molte situazioni realistiche, che coinvolgono, ad esempio, entità governative e imprese private, o fornitori e produttori della stessa catena di approvvigionamento. L'obiettivo è esaminare in modo esaustivo la letteratura esistente sull'ottimizzazione bi-livello con prospettive di sostenibilità. Nel complesso, questa tesi mostra il potenziale della Ricerca Operativa nel perseguire la sostenibilità anche in contesti decisionali molto complessi, risolvendo diverse applicazioni reali con una varietà di metodi di ottimizzazione.

Parole chiave: Sostenibilità; Programmazione Lineare Intera Mista; Machine Learning; Ottimizzazione; Applicazioni Reali

Contents

List of Figures

List of Tables

Acknowledgements

My deepest gratitude goes to my thesis advisor, Professor Manuel Iori. He introduced me to the field of Operations Research and transferred to me his great passion for it. During this three-year journey, he gave me lots of opportunities for growing my academic knowledge as well as for developing my own research interests and ideas. I want to sincerely thank him for his trust, encouragement, insightful suggestions, and appreciation, without which this thesis would not have been possible.

I also want to sincerely thank my co-supervisor, Professor Carlo Alberto Magni. He introduced me to the subject of Engineering Economics and taught me the art of caring for the details. During my last year of PhD, he provided me with precious teaching opportunities in his courses at the Department of Economics “Marco Biagi”.

Throughout my PhD, I have had the privilege of interacting with exceptional professors and researchers who have enriched my knowledge and research interests. I want to thank Maxence Delorme, who was the very first one to talk to me about the PhD and academic career. His guidance on Combinatorial Optimization and coding have been invaluable. He followed me during the very first research projects with patience and support, and he was always very kind.

I am immensely grateful to Professor Ivana Ljubić, who hosted me in the kindest way in Paris. During my five-month visiting period at ESSEC Business School, I had the great opportunity to collaborate with Ivana and being introduced to the topic of Bilevel Optimization, of which she is an expert. I am grateful for the opportunity to discuss with her on research, learn how to build a solid network in the academia, and climb to the top of Puy-de-Dôme in Clermont-Ferrand. She is the best female researcher I know and a great source of inspiration.

Furthermore, a sincere thank is due to Professor Joseph Hartman for his warm hospitality during my visit to the University of Massachusetts Lowell. He dedicated to me his time for discussing on Dynamic Programming. I felt very honored.

I also want to thank Professor Juan José Miranda Bront for our recent discussions on exact optimization algorithms and his encouraging attention to my academic research.

Special thanks to my colleagues at the Department of Sciences and Methods for Engineering, especially Mirko, Dario, Mirko, Giomaria, Giorgio, Francesco, and Alberto. To Alice and Martina, my beautiful Italian friends in Paris. To the Marco Biagi Foundation and, very specially, my friends Federica, Ilaria, Rolando, and Fabio. They have been the best part of every day. Finally, I want to thank my sister Chiara for being there, and especially my parents, Antonio and Cristina, who always inspired my work and supported me in my decisions.

Chapter 1

Introduction

This PhD thesis studies increasingly complex decision-making and management problems arising in the field of Sustainable Operations (SO), starting from mono-objective, moving to bi-objective, and finally to bilevel optimization problems. The thesis aims to contribute to the advancement of Operations Research (OR) by employing various optimization techniques, by addressing economic, environmental, and social sustainability, and by solving problems encountered in real-world manufacturing and service contexts. Given that the chapters are self-contained and explore a range of analytical methods from OR and related disciplines, this introduction section first provides a brief outline of the main methodologies applied within each chapter of the thesis, and then provides a summary of the key contributions of each chapter and their corresponding outcomes as documented in peer-reviewed journal publications and conference proceedings.

Operations Research

OR is the science of driving better decisions through the application of advanced analytical methods (Hillier and Lieberman, 2021). Mathematical programming, among other methodologies, is applied to complex decision-making problems typically faced by enterprises of any kind, but also public institutions and Non-Governmental Organizations to find optimal or near-optimal solutions. OR has various fields of application, among which the most famous ones include the transportation (e.g., to optimize the routing plans of vehicles for goods transportation), telecommunication (e.g., to define the best location for concentrators in the network to reach demand points), logistics (e.g., to design the optimal network of the supply chain), project management (e.g., to allocate resources to project tasks), and healthcare (e.g., to manage hospital patients flow) sectors. Over time, OR has continuously grown in terms of both methods and applications (Petropoulos et al., 2023). This growth has been driven by the well-known complexity of OR problems, which has been stimulating new branches of research to address the challenging task of solving these problems, and the great success in many applications since the early moments of the existence of OR. Moreover, thanks to the advancements in computer technology, data management, optimization software, and machine learning tools, greater theoretical and practical achievements have been made by the OR community in the last decades (see, e.g., Agrawal et al., 2010). Sustainability and sustainable development have been addressed by OR giving rise to novel branches within the discipline, such as SO, thereby presenting new problems and solution methods.

Mathematical programming

Given a set of discrete variables, a Combinatorial Optimization (CO) problem is to find a subset of variables whose values identify an optimal solution with respect to a given (minimization or maximization) objective function subject to a set of specific combinatorial constraints. Enumerating all feasible solutions is very often impossible, thus most of the CO problems are known in the complexity theory to be $\mathcal{NP}$-hard (Garey and Johnson, 1979). On the other hand, CO problems arise in many industrial sectors, involving a mix of continuous and integer-valued decision variables. Given vectors b, c, d and matrices A, B, a Mixed Integer Linear Programming (MILP) problem (Conforti et al., 2014) can be mathematically formulated in matrix notation as

$$\min \quad z = c^T x + d^T t \tag{1.1}$$

$$\text{s.t.} \quad Ax + By = b \tag{1.2}$$

$$x, y \geq 0 \tag{1.3}$$

$$x \in \mathbb{Z} \tag{1.4}$$

Equation (1.1) represents the objective function, with any maximization possibly rewritten as minimization. Equation (1.2) includes the linear constraints that must be satisfied by a feasible solution together with equations (1.3)-(1.4), stating the non-negative domain of x and y variables, with x limited to integer values. When integrality is relaxed, we obtain a pure Linear Programming (LP) problem formulated as $\min\{c^T x : Ax = b, x \geq 0\}$. When all variables are integer-valued, MILP reduces to ILP.

The variables represent the decision of the problem. To solve an MILP problem to optimality means to find the best feasible solution, that is a set of decisions that give the best objective function value while satisfying all the constraints. Real-world decision-making problems modelled with MILP are countless and many of them are notably difficult to solve to optimality. Enumerating all possible solutions is not possible in practice (i.e., computationally in reasonable time) even for simple problems like assigning n jobs to n workers to minimize the overall completion time of all the jobs. Therefore, since back to the 1960s, several exact and heuristic algorithms have been proposed in the literature to solve MILP problems to optimality or near-optimality, respectively (Hillier and Lieberman, 2021). Most exact algorithms are based on implicit (i.e., intelligent) enumeration of solutions and on simplifications of the original problem. In the famous Branch&Bound (B&B) method for MILPs, a branch-decision tree is built by dividing the original problem, at the root node of the tree, into subproblems ("branching") and estimating a bound on the optimal solution for each subproblem, then pruning tree branches that cannot lead to an optimal solution based on these bounds and exploring the ones with good bounds ("bounding"). State-of-the-art commercial MILP solvers, such as `Gurobi, IBM ILOG CPLEX` and `FICO Xpress`, include built-in exact algorithms based on B&B. A great effort has been addressed to improving more and more the computational performance of optimization solvers. All modern solvers enhance the B&B framework with the extensive use of additional algorithmic components, preprocessing and primal heuristics. Today we can model a variety of decision-making problems in different programming languages and obtain optimal or good solutions in short times. Further approaches based on B&B have been theoretically designed and implemented in the literature, including Branch&Cut (B&C), Branch&Price (B&P), which basically reduce the search space of solutions by tailored mathematical programming techniques. Decomposition is another optimization approach that allows to simplify the original problems by breaking it down into, typically, a master problem that is easier to solve and subproblems solved iteratively until the optimal solution for the original problem is found. Several decomposition methods have been studied for MILPs, based on the characteristics of the problem (i.e., the natural divisibility of primary and secondary decisions). Famous decompositions are the Benders' decomposition and the column generation method, that is typically integrated with B&B into B&C. When the complexity of the problem is too high to be treated with exact approaches, that is, when too many branching steps or decomposition iterations are necessary to reach the optimal solutions, heuristic algorithms can help to obtain solutions of good quality (i.e., close to the optimal).

Mathematical programming has soon evolved to consider multiple conflicting objectives of a single decision maker. A multi-objective LP (Ehrgott, 2005) is formulated as

$$\min \quad (z_1, z_2, \dots, z_p) = (c_1^T x, c_2^T x, \dots, c_p^T x) \tag{1.5}$$

$$\text{s.t.} \quad Ax = b \tag{1.6}$$

$$x \geq 0 \tag{1.7}$$

with p objectives in equation (1.5), under the assumption that there is not a feasible solution that minimizes all the objectives at the same time. Therefore, finding the optimal solution for each single objective is easier than finding the overall optimum. Restricting some (all) decision variables to be integer-valued, we get a general multi-objective MILP (ILP). Solving multi-objective programming problems to optimality means to find one or more Pareto-optimal or non dominated solutions, that are feasible solutions in the criteria space of all the objectives such that no other feasible solution exist that strictly improve one of the objectives without worsening (or not improving) the others. The *Pareto set* is the complete set of Pareto-optimal solutions. Notable solution methods for multi-objective programming are the weighted sum and the ϵ-constraint approaches. The weighted sum method consists in defining a single objective function as a linear combination of the p objectives defined by

$$z_\lambda = \lambda^T (z_1, z_2, \dots, z_p) = \sum_{i=1}^{p} \lambda_i c_i^T x \tag{1.8}$$

and then solving the resulting single-objective problem "$\min z_\lambda$". This method provides a single Pareto-optimal solution. The decision maker guides the method in deciding *a priori* the set of weights $\lambda_1, \lambda_2, \dots, \lambda_p$ that mathematically express the importance of each objective. The ϵ-constraint method generates a set of bi-objective solutions by converting all objectives apart from one into additional constraints that restrict the objective values to be lower than an arbitrarily small ϵ, and iteratively solve the single-objective problem with updated ϵ values at each iteration. In the bi-objective case ($p = 2$), the resulting mono-objective problem at the h-th iteration is formulated as

$$\min \quad z_1 = c_1^T x \tag{1.9}$$
$$\text{s.t.} \quad Ax = b \tag{1.10}$$
$$c_2^T x \leq \epsilon_h \tag{1.11}$$
$$x \geq 0. \tag{1.12}$$

This method allows to generate a set of solutions, containing the full Pareto front of solutions spread in the criteria space between the best solutions for one objective and the other. One can thus observe the trade-off between the objectives and select *a posteriori* the best solution with the desired balance of objectives. Although less attractive for the OR community from a theoretical point of view (or at least until very recently), multi-objective optimization allows to model a variety of real-world problems where two or more objectives must be balanced.

Going further in the complexity of decision-making problems, one can consider realistic scenarios with multiple actors with multiple and different goals and different levels of decision power/influence. While single- and multi-objective optimization model a single level of decision with one or more goals, multi-level optimization extends the ILP theory to deal with multiple decision levels. In particular, bilevel optimization allows to formalize decision processes where multiple decision makers are organized within a two-level hierarchy (Fischetti et al., 2017). A general optimistic bilevel programming problem (Dempe, 2020) modeling a sequential game with two players, called the *leader* and the *follower*, is formulated as

$$\min_{x \in X, y} \quad F(x, y) \tag{1.13}$$
$$\text{s.t.} \quad G(x, y) \geq 0 \tag{1.14}$$
$$y \in S(x), \tag{1.15}$$

where $S(x)$ is the set of optimal solutions of the x-parameterized problem

$$\min_{y \in Y} \quad f(x, y) \tag{1.16}$$
$$\text{s.t.} \quad g(x, y) \geq 0. \tag{1.17}$$

Models (1.13)-(1.15) and (1.16)-(1.17) define, respectively, the so-called upper-level (UL) and lower-level (LL) problems. The UL variables $x \in \mathbb{R}^{n_x}$ represent the decisions of the leader, and the LL variables $y \in \mathbb{R}^{n_y}$ represent the decisions of the follower. The leader, positioned higher in the hierarchy, decides first anticipating the follower's decision outcome. The follower, lower in the hierarchy, reacts to the leader's decision afterwards. In the optimistic scenario, that is the most studied so far, the leader controls the variables of the follower such that the follower makes the best solution for the leader among the multiple LL optimal solutions. More complex bilevel frameworks allows to include multiple leaders and followers in cooperation or competition. Bilevel problems are proven to be extremely complex due to their nested structure. Exact approaches are based on single-level reformulations, which usually generate Mixed Non-Linear Integer Programs (MINLP), that in turn require advanced mathematical programming techniques to be treated in theory and practice. The literature on exact (Kleinert et al., 2021) and heuristic (Camacho-Vallejo et al., 2024) methods is growing rapidly.

Forecasting

Forecasting refers to the systematic process of predicting future values or outcomes related to specific decision variables that are essential for decision-making within an organization. This predictive analysis is often based on historical data and mathematical models. OR practitioners typically use forecasting to anticipate demand for products or services, optimize resource allocation, and enhance inventory management by predicting demand trends and inventory levels to allow operational decision makers making informed decisions and ultimately improve the overall operational efficiency. Although production contexts are the most common fields of application, forecasting is widely spread in all human systems, including project management, traffic management, retail, and finance (Petropoulos et al., 2022). Forecasting is one approach among many others developed in OR for dealing with uncertainty, by providing reliable estimates of future events or conditions. This approach is typically embedded within *predict-then-optimize* approaches (Elmachtoub and Grigas, 2022), where the forecasting outputs become the input of the following optimization step. On the contrary, *predict-and-optimize* approaches directly integrate uncertainty within mathematical programming models and methods. These approaches include stochastic and robust optimization, as well as new Machine Learning (ML) methods.

When predicted values are gathered over time, particularly at regular intervals, one refers to time-series forecasting. Forecasting models provide not only point estimates but also probabilistic forecasts, prediction intervals, or path forecasts. Several statistical and ML tools have been designed for time-series forecasting. Statistical methods (Hyndman and Athanasopoulos, 2018) rely directly on mathematical relationships between data (e.g., by estimating a linear regression model on the data) and can be generally expressed as

$$y_{t+1} = f(y_1, y_2, \ldots, y_t) + \gamma_t \tag{1.18}$$

where y_{t+1} is the predicted value for the next time period, $f(y_1, y_2, \ldots, y_t)$ is a mathematical function that captures the relationship between past observations and the predicted value, and γ_t is the forecasting error term. The goal is to estimate the best-fitting relationship function. Forecasting ML methods (Sarker, 2021), in contrast, aim to capture complex patterns and non-linear relationships. In this case, a complex function maps a set of input features (potentially not only historical data) to the predicted value. ML methods leverage algorithms such as neural networks, support vector machines, or ensemble methods (e.g., random forest) to identify intricate patterns within the data. Often, a hybrid approach combining both classes is employed for comprehensive forecasting solutions.

Sustainable Operations

SO employs quantitative methods from OR to achieve economic, environmental, and social sustainability in business (Jaehn, 2016). This emerging interdisciplinary field can develop to be considered the main tool for OR researchers and practitioners to face the urgent climate crisis. The concept of sustainable development, as introduced by the United Nations (UN) in 1987 and further developed in the “2030 Agenda” for sustainable development (United Nations, 2015), has significantly influenced the global understanding of sustainability. Additionally, the adoption of the triple bottom line accounting framework in business (Elkington, 2002) has contributed to establish the definition of sustainability as a three-dimensional principle, based on which economic development should be pursued by companies and societies in a manner that benefits both the well-being of people and the health of the planet. OR plays a central role in sustainable development and, more specifically, in pursuing UN SDGs. Consider for instance Goal 1 ("No Poverty") and Goal 2 ("Zero Hunger") that are addressed, among many, by humanitarian logistics for food chains (Peters et al., 2021), but also Goal 7 ("Affordable and Clean Energy") and Goal 11 ("Sustainable Cities and Communities") tackled, for instance, by routing with emissions reduction (Bektaş and Laporte, 2011) , and more generally Goal 8 ("Decent Work and Economic Growth"), Goal 9 ("Industry, Innovation, and Infrastructure") and Goal 12 ("Responsible Consumption and Production") addressed by product design, manufacturing, technology selection, and reverse logistics (Tang and Zhou, 2012). In addition to classical OR problems, whose aim is to improve efficiency and profitability of firms, multi-objective programming and multi-level programming allow, respectively, to balance economic, environmental, and social goals (e.g., not only maximizing a profit function/minimizing costs but also minimizing the environmental footprint and maximizing social inequality) and involve multiple stakeholders' decisions (e.g., the authority sets carbon public policies and companies adapt their production plan to reduce the financial impact). The new field of SO brings these efforts together and, in this sense, sustainability enriches OR. In turn, working on sustainability problems, which involve decision-making, optimization, uncertainty, and ML, enriches OR by generating new challenging sustainability problems.

Engineering Economics

Engineering Economics (EE) applies economic principles to the analysis and solution of engineering problems (Hartman, 2007; Blank and Tarquin, 2014). It involves evaluating the costs and benefits associated with engineering projects, investments, and decisions. The primary goal is to optimize resource allocation and decision making within engineering contexts to maximize efficiency and achieve desired financial and strategic outcomes. Applications of EE include project valuation and selection (i.e., evaluating different project alternatives and selecting the most economically viable option with respect to organizational objectives), capital budgeting (i.e., prioritizing projects based on their potential to generate value and contribute to long-term profitability), project cost estimation and control, asset management, and replacement analysis. Corporate finance models typically seek to derive economic value from projects (Brealey et al., 2023), whereas OR concentrates on optimizing a given operational objective through the determination of optimal decision variables. EE integrates both aspects, addressing both the financial considerations and operational facets of managerial decision-making. For instance, it considers trade-offs among economic, environmental, and ethical considerations in replacement analysis (see, e.g., Riechi et al., 2017).

Contributions

The overall motivation of the present thesis is to explore the potential of OR in achieving sustainability in less and more complex decision-making contexts of industrial and service businesses through the formulation of original models and methods and the review of advanced techniques. Each chapter of the thesis is self-contained and can be read independently of the others.

Methodology Chapters 2-4 and Chapters 6-7 deal with different CO problems in the classes of assignment, scheduling, facility location, and asset replacement inspired by real-world decision-making contexts of, respectively, education and healthcare, industrial production, waste management, and transportation. The common methodology involves the definition of original and effective MILP formulations. In Chapter 5, ML methods are employed for a forecasting problem in the food industry. Tailored optimization algorithms are proposed to find optimal solutions for real-world instances of such problems. Extensive computational experiments are also performed to test the validity and scalability of the algorithms and provide valuable results within the realm of OR scientific research. On the other hand, Chapters 8-9 contain surveys. More than 120 optimization problems are classified and analyzed to provide original and comprehensive overviews of the selected topics.

Chapters outline We can consider the present thesis divided in three parts, dealing with different OR problems of increasing complexity with respect to mathematical programming aspects and decision-making elements related to sustainability perspectives. The first part of the thesis (Chapters 2-5) deals with MILP models, tailored exact approaches, and machine learning algorithms for real-world single-objective problems involving one decision maker.

In Chapter 2, the Tutor Allocation Problem, that is assigning tutors to workshops of academic courses to maximize tutors' preferences, is formulated and solved by an ILP model to optimality. The model is tested on one real instance of a British university, obtaining a significant improvement with respect to the manual assignment in use at the time. Further computational experiments prove that such improvement could be maintained while optimizing other key performance indicators for workers' satisfaction such as load balance among groups of tutors and the total number of courses assigned. This work has been published as a paper in Caselli, Delorme, Iori (2022).

In Chapter 3, a different assignment problem faced by an Italian hospital during the COVID-19 emergency is studied for the reduction of overcrowding in the waiting rooms dedicated to outpatient services. Overcrowding is a measure of infection risk at the pandemic time, where strict social distancing measures were introduced by governments. An ILP model is formulated to identify the weekly optimal layout and applied to real data from the hospital. Several tests are performed on the model to find the best trade-off between the reduction of overcrowding and the changes across the weeks to provide an optimal solution applicable in practice. This work has been published as proceedings of the 2021 IEEE International Conference on Industrial Engineering and Engineering Management in Caselli et al. (2021).

Chapter 4 addresses a real-world scheduling problem occurring in the engineering test laboratory of a multinational company producing hydraulic components for motion systems, where the objective is to allocate a set of tasks to a set of machines and to a set of workers in such a way that the total weighted tardiness is minimized. The problem is original in the scheduling literature. The chapter presents two MILP formulations, a Constraint Programming (CP) model, and an original combinatorial Benders' decomposition and demonstrates their effectiveness through extensive computational experiments on both real-world and random instances. This work has been published as a paper in Caselli, Delorme, Iori, Magni (2024).

Chapter 5 investigates a forecasting problem faced by an Italian company in the perishable food industry and, more specifically, in the highly pollutant business of processed meat. The demand of a variety of products from supermarkets and food retailers to a production facility must be predicted day by day as accurately as possible (i.e., to minimize the forecasting error). Three ML methods are applied to a real dataset provided by the company: random forests, support vector regression, and artificial neural networks. Their performance is compared with classical baseline models, among which the method currently used by the company is included. Computational experiments measure the outperformance of the proposed combination of ML methods on the baselines, and a simulation test shows the practical implications of the use of forecasting models on the company's inventory management. In the latter, three key performance indicators of productivity and inventory are simulated, that are the daily manual adjustments on forecasting error, the size of the warehouse, and the daily number of expired products, all resulting in reduced values under the proposed forecasting scenarios. The results of this chapter have been submitted for publication to an international journal (Mucciarini et al., 2024).

The second part of the thesis (Chapters 6-7) studies two bi-objective problems and provide exact and approximated bi-objective optimization approaches with the aim of balancing economic efficiency and environmental sustainability goals of a single-decision maker. Chapter 6 deals with a bi-objective real-world facility location problem faced by an Italian multi-utility company operating in the sector of waste management. The flow of different classes of recyclable waste must be optimized by deciding whether and where to open additional intermediate transfer facilities among a set of dedicated points in a regional waste transfer network. The aim is to minimize the total costs and the CO_2 emissions involved in the process. A detailed description of the cost and emissions parameters estimation is provided. Two MILP models are formulated and applied within three multi-objective optimization methods: the exact and heuristic weighted sum and an approximated ϵ-constraint methods. Two real-world case studies are solved, for which approximated Pareto fronts of solutions are given with significant reductions of costs and emissions with respect to the status quo of the network. Further computational experiments on random instances show that the models can be employed for analogous real-world applications. A preliminary work was published as proceedings of the 2022 International Network Optimization Conference in Caselli, Columbu, Iori, Magni (2022). The extended version of this work has been submitted for publication to an international journal and is currently under review (Caselli, Columbu, Iori, Magni, Olveira, 2024).

Chapter 7 is the result of an open work which started from the collaboration with Professor Joseph Hartman of the University of Massachussetts Lowell (US). During the third year of my PhD, I visited UMass Lowell for a one-month research period. Research on original formulations, theoretical results, and computational tests have been conducted and are still ongoing. The topic of the chapter is a bi-objective vehicle replacement problem with budget constraint, where the goals are the minimization of the total discounted costs and the minimization of the total CO_2 emissions generated by the use of the fleet. The problem is inspired by two real-world applications of Italian companies that every year must evaluate the state of the vehicles fleet used for transportation, considering efficiency and technology selection (e.g., fleet electrification). Two alternative ILP formulations and a forward Dynamic Programming (DP) model are presented with preliminary computational results. Principles and modeling approaches from EE are integrated in the proposed methods for evaluating the economic sustainability of the vehicle replacement problem within the objective function. Preliminary computational experiments, which were presented at the 2023 POMS International Conference (Caselli, Hartman, et al., 2024), show that ILP is a promising approach for trading off financial and environmental sustainability in the considered vehicle replacement problem. The chapter is closed by presenting the future work, that will be mainly address to further improve the proposed algorithms, derive valuable results on the bi-objective replacement literature, and ultimately solve the real case studies.

The third part of the thesis (Chapters 8-9) explores the field of bilevel optimization. This is the result of two open works (see Caselli, Iori, Ljubić, 2024a and Caselli, Iori, Ljubić, 2024b) which started from the collaboration with Prof. Ivana Ljubić from ESSEC Business School of Paris (France), where I did a five-month research stay during the third year of my PhD.

Chapter 8 contains a survey on bilevel optimization problems specifically addressed to applications with sustainability perspectives, where multiple stakeholders with multiple economic, environmental, and social objectives are involved. The survey starts by outlining a complete overview on the structure of the bilevel problems, the type of leaders and followers involved, the different goals, and the common solution methods for the selected bilevel applications. A following section presents the typical mathematical formulation for basic bilevel problems as a starting point for generalization. Then, five sections summarize the selected work. The classification is based on the application sectors. The considered sectors are: transportation and logistics; production planning and manufacturing; waste, material, and environment management; supply chains; disaster prevention and response. Sections are divided in subsections defined by specific topics or famous OR problems, and for each subsection a summary table is provided giving information on bilevel games, UL and LL players and goals, and solution methods for each reference. The survey, which is still under authors' revision, is concluded by identifying the interrelation between bilevel optimization and sustainability perspectives and gives insights on possible directions for future research in the field.

Chapter 9 contains a second survey dedicated to bilevel optimization problems exclusively in the green energy sector. Given the importance of the sector and the numerous works already present in the literature, we analyze the energy sector as a distinct topic. Again, the focus is on energy-related bilevel problems with sustainability perspectives of the leader and the followers. The survey is structured similarly to that contained in Chapter 8 and is still under authors' revision.

List of contributions The content of this thesis is the result of two peer-reviewed journal articles, three conference proceedings, and five technical reports (two currently submitted for publications). All the works are the results of past and present collaborations with several co-authors and involve a company, a foreign university, or both. Table 1.1 summarizes the main contributions related to each chapter, including academic and industrial partners involved in the works.

Table 1.1: Summary of contributions.

Chapter	Reference	Partners
2	Caselli, G., Delorme, M., Iori, M. (2022). Integer Linear Programming for the Tutor Allocation Problem: A Practical Case in a British University. *Expert Systems with Applications*, 187, 115967.	The University of Edinburgh
3	Caselli, G., De Santis, D., Delorme, M., Iori, M. (2021). A Mathematical Formulation for Reducing Overcrowding in Hospitals' Waiting Rooms. In *Proceedings of 2021 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)* (pp. 297-301). IEEE.	MAPS Tilburg University
4	Caselli G., Delorme M., Iori M., Magni C. A. (2022). Mixed Integer Linear Programming for a Real-World Parallel Machine Scheduling Problem with Workforce and Precedence Constraints. In: Amorosi, L., Dell'Olmo, P., Lari, I. (Eds.) *Optimization in Artificial Intelligence and Data Sciences*, AIRO Springer Series, vol 8., pp. 61-71, Springer, Cham. Caselli, G., Delorme, M., Iori, M., Magni, C. A. (2024). Exact algorithms for a Parallel Machine Scheduling Problem with workforce and contiguity constraints. Computers & Operations Research, 163, 106484.	DANA Tilburg University
5	Mucciarini, M., Caselli, G., Iori, M., Lippi, M. (2024). A Real-World Application of Demand Forecasting in the Perishable Food Industry (under review).	INALCA
6	Caselli G., Columbu G., Iori M., Magni C.A. (2022). Mixed Integer Linear Programming for CO2 emissions minimization in a Waste Transfer Facility Location Problem. In *Proceedings of the 10th International Network Optimization Conference (INOC)*, Aachen, Germany, June 7–10, 2022 (pp. 51-56). Caselli, G., Columbu, G., Iori, M., Magni, C. A., Oliveira, M. (2024). Minimizing costs and CO_2 emissions in a waste transfer facility location problem (under review).	iren UFV
7	Caselli, G., Hartman, J., Iori, M., Magni, C. A., Zucchi, G. (2024). Optimal vehicle replacement with budget constraints and CO_2 emissions minimization (technical report presented at *POMS2023 International Conference*, Paris, France, July 18–20, 2023).	coopservice UMASS LOWELL
8	Caselli, G., Iori, M., Ljubic, I. (2024). Bilevel optimization with sustainability perspective: a survey on applications (technical report).	ESSEC Business School
9	Caselli, G., Iori, M., Ljubic, I. (2024). Bilevel optimization for green energy problems (technical report).	ESSEC Business School

References

Agrawal, S., Subramanian, K., & Kapoor, S. (2010). Operations research-contemporary role in managerial decision making. *International Journal of Research and Reviews in Applied Sciences*, *3*(2), 200–208.

Bektaş, T., & Laporte, G. (2011). The pollution-routing problem. *Transportation Research Part B: Methodological*, *45*(8), 1232–1250.

Blank, L., & Tarquin, A. (2014). *Basics of engineering economy*. McGraw-Hill.

Brealey, R. A., Myers, S. C., Allen, F., & Edmans, A. (2023). *Corporate finance* (14th ed.). McGraw-Hill.

Camacho-Vallejo, J.-F., Corpus, C., & Villegas, J. G. (2024). Metaheuristics for bilevel optimization: A comprehensive review. *Computers & Operations Research*, *161*, 106410.

Caselli, G., Columbu, G., Iori, M., & Magni, C. A. (2022). A mixed integer linear programming model for co2 emissions minimization in a waste transfer facility location problem. *Proceedings of the 10th International Network Optimization Conference (INOC), Aachen, Germany, June 7–10, 2022*, 51–56.

Caselli, G., Columbu, G., Iori, M., Magni, C. A., & Olveira, M. (2024). Minimizing costs and co2 emissions in a waste transfer facility location problem (manuscript submitted for publication).

Caselli, G., De Santis, D., Delorme, M., & Iori, M. (2021). A mathematical formulation for reducing overcrowding in hospitals' waiting rooms. *Proceedings of the 2021 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 297–301.

Caselli, G., Delorme, M., & Iori, M. (2022). Integer linear programming for the tutor allocation problem: A practical case in a british university. *Expert Systems with Applications*, *187*, 115967.

Caselli, G., Delorme, M., Iori, M., & Magni, C. A. (2024). Exact algorithms for a parallel machine scheduling problem with workforce and contiguity constraints. *Computers & Operations Research*, *163*, 106484.

Caselli, G., Hartman, J., Iori, M., Magni, C. A., & Zucchi, G. (2024). Optimal vehicle replacement with budget constraints and co2 emissions minimization (technical report).

Caselli, G., Iori, M., & Ljubić, I. (2024a). Bilevel optimization for green energy problems (technical report).

Caselli, G., Iori, M., & Ljubić, I. (2024b). Bilevel optimization with sustainability perspective: A survey on applications (technical report).

Conforti, M., Cornuéjols, G., Zambelli, G., Conforti, M., Cornuéjols, G., & Zambelli, G. (2014). *Integer programming models*. Springer.

Dempe, S. (2020). *Bilevel optimization: Advances and next challenges*. Springer International Publishing.

Ehrgott, M. (2005). *Multicriteria optimization* (Vol. 491). Springer.

Elkington, J. (2002). *Cannilbals with forks: The triple bottom line of the 21st century*. Oxford Press.

Elmachtoub, A. N., & Grigas, P. (2022). Smart "predict, then optimize". *Management Science*, *68*(1), 9–26.

Fischetti, M., Ljubić, I., Monaci, M., & Sinnl, M. (2017). A new general-purpose algorithm for mixed-integer bilevel linear programs. *Operations Research*, *65*(6), 1615–1637.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of np-completeness*. W. H. Freeman.

Hartman, J. (2007). *Engineering economy and the decision-making process*. Pearson.

Hillier, F. S., & Lieberman, G. J. (2021). *Introduction to Operations Research* (11th). McGraw-Hill Education.

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and practice*. OTexts.

Jaehn, F. (2016). Sustainable operations. *European Journal of Operational Research*, *253*(2), 243–264.

Kleinert, T., Labbé, M., Ljubic, I., & Schmidt, M. (2021). A survey on mixed-integer programming techniques in bilevel optimization. *EURO Journal on Computational Optimization*, *9*, 100007.

Mucciarini, M., Caselli, G., Iori, M., & Lippi, M. (2024). A real-world application of demand forecasting in the perishable food industry (manuscript submitted for publication).

Peters, K., Silva, S., Gonçalves, R., Kavelj, M., Fleuren, H., den Hertog, D., Ergun, O., & Freeman, M. (2021). The nutritious supply chain: Optimizing humanitarian food assistance. *INFORMS Journal on Optimization*, *3*(2), 200–226.

Petropoulos, F., Apiletti, D., Assimakopoulos, V., Babai, M. Z., Barrow, D. K., Taieb, S. B., Bergmeir, C., Bessa, R. J., Bijak, J., Boylan, J. E., et al. (2022). Forecasting: Theory and practice. *International Journal of Forecasting*, *38*(3), 705–871.

Petropoulos, F., Laporte, G., Aktas, E., Alumur, S. A., Archetti, C., Ayhan, H., Battarra, M., Bennell, J. A., Bourjolly, J.-M., Boylan, J. E., Breton, M., Canca, D., Charlin, L., Chen, B., Cicek, C. T., Jr, L. A. C., Currie, C. S., Demeulemeester, E., Ding, L., . . . Zhao, X. (2023). Operational research: Methods and applications. *Journal of the Operational Research Society*, 1–195.

Riechi, J., Mácian, V., Tormos, B., & Avila, C. (2017). Optimal fleet replacement: A case study on a spanish urban transport fleet. *Journal of the Operational Research Society*, *68*(8), 886–894.

Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, *2*(3), 1–21.

Tang, C. S., & Zhou, S. (2012). Research advances in environmentally and socially sustainable operations. *European Journal of Operational Research*, *223*(3), 585–594.

United Nations. (2015). Transforming Our World: The 2030 Agenda for Sustainable Development [Retrieved Feb. 3, 2024]. https://undocs.org/en/A/RES/70/1

Chapter 2

Integer Linear Programming for the Tutor Allocation Problem: A Practical Case in a British University[1]

Abstract

In the Tutor Allocation Problem, the objective is to assign a set of tutors to a set of workshops in order to maximize tutors' preferences. The problem is solved every year by many universities, each having its own specific set of constraints. In this work, we study the tutor allocation in the School of Mathematics at the University of Edinburgh, and solve it with an integer linear programming model. We tested the model on the 2019/2020 case, obtaining a significant improvement with respect to the manual assignment in use and we showed that such improvement could be maintained while optimizing other key metrics such as load balance among groups of tutors and total number of courses assigned. Further tests on randomly created instances show that the model can be used to address cases of broad interest. We also provide meaningful insights on how input parameters, such as the number of workshop locations and the length of the tutors' preference list, might affect the performance of the model and the average number of preferences satisfied.

2.1 Introduction

As in many other countries, British academic courses are divided into lectures and workshops. While lectures bring the necessary theoretical background to the students, workshops allow them to put their newly acquired knowledge in practice. Lectures are delivered by *lecturers* and workshops are delivered by *tutors*. In most courses, only one lecturer is responsible for the lectures during the academic year. As most lecturers have open-ended contracts with their universities, once a lecturer is assigned to a course, there is an interest in keeping it over the years: the preparation time becomes shorter and the teaching abilities increase. Such consistency is unfortunately not possible for the

[1]The results of this chapter appear in: Caselli, G., Delorme, M., Iori, M. (2022). Integer Linear Programming for the Tutor Allocation Problem: A Practical Case in a British University. *Expert Systems with Applications*, 187, 115967.

workshops: indeed, as the majority of the tutoring staff is composed of PhD students and post-doctoral researchers, it changes every year. In addition, as workshops require more than one tutor (depending on the ratio tutors/students wished by the university), the problem of assigning tutors to workshops, also called the *Tutor Allocation Problem* (TAP), is far from trivial and has to be solved every year, representing a significant workload for the administrative team.

In the School of Mathematics at the University of Edinburgh, the TAP is solved manually by the administrative staff in August, before the beginning of every academic year. An update taking into account the staff changes is also performed in November. Even though the TAP is solved manually, the decision makers have to take into account several parameters. First, tutors have different sets of skills, and high quality workshops require tutors with the appropriate skills. Second, different kinds of constraints have to be respected: for example, some tutors have a maximum number of tutoring hours specified in their working contracts and there are incompatibility constraints between two simultaneous workshops. Third, some tutors have a preference list (not ranked in our case), that consists of a subset of courses that they wish to tutor. In general, a tutor is happier and gives better tutoring if she is assigned to a workshop in her preference list. Another key aspect of the TAP is the fact that the university is a dynamic environment: indeed, the members of staff may change unexpectedly, or the number of students in a given workshop may increase, requiring an additional tutor. The slightest update in the data makes the assignment obsolete and requires a new solution.

Currently, the School of Mathematics first assigns lecturers to lectures, then schedules the lectures and workshops (together with the room allocation), and then uses an iterative process to solve the TAP. Based on the tutors' preferences, skills, and availabilities, an initial assignment is proposed to the faculty members. After collecting the requests for changes, another solution with minor modifications is proposed. The process may be iterated a few times before the final assignment is reached.

In this work, our goal is to ease the work by the administrative staff by proposing an *Integer Linear Programming* (ILP) model whose goal is to provide the initial assignment. The ILP model has several advantages. First, it ensures that all the constraints are satisfied (we observed that a manual assignment sometimes allocates a tutor to two simultaneous workshops, resulting in a request for changes). Second, as our objective function is to maximize the tutors' satisfaction, it provides better allocations for the tutors. Third, it can be easily generalized to include further constraints, thus making it a powerful tool for addressing several real-world situations.

The rest of this work is organized as follows. A concise description of the related literature is given in Section2.2. In Section 2.3, we provide a detailed description of the problem. In Section 2.4, we formally present our mathematical model. Section 2.5 reports the computational experiments, including a set of tests on our real-world case in which we explore the use of additional constraints to make the allocation fairer and the use of ordered preference lists, as well as a number of additional tests performed on randomly created instances with the aim of assessing the model behavior and evaluating the impact of some key input parameters. Concluding remarks are finally provided in Section 2.6.

2.2 Concise literature review

The TAP and its related problems have been studied since the early seventies (see Andrew and Collins, 1971). We refer the reader to the survey of Carter and Laporte (1998), who reviewed in details the papers related to real-world allocation problems faced by universities. In the following, we mention some of the most important mathematical models and heuristics proposed in the literature for the TAP and closely related problems. To ease the comprehension, we always use the TAP terminology (i.e., tutors and workshops) to describe the models. We also provide a recapitulative table summarizing the main features of each problem at the end of the section (Table 2.1).

In the late seventies, Dyer and Mulvey (1976) proposed an ILP model to assign workshops to tutors. The model maximizes the tutors' preferences, while taking into account a minimum/maximum number of tutors per workshop, and a minimum/maximum number of workshops per tutor. The model also takes into account a large time period (such as a quarter or a semester) in order to balance the tutors' workload. They showed that the problem could be transformed into a network flow problem, but did not provide computational experiments. A few years later, a similar model was used by Hill et al. (1983) for solving a more general class of problems. Their model was tested on two real-world problems: assigning students to job interview slots and assigning students to project teams in an MBA field project course. The model could also take into account the tutors' skills and make sure that a minimum number of tutors with the appropriate skills were assigned to the workshops. Dinkel et al. (1989) also proposed a network flow formulation for an extension of the TAP in which the room and the schedule is determined by the model.

As noticeable variations of the previous models, we mention the work of Breslaw (1976), who proposed a model based on continuous variables used to assign percentages of workshops to tutors. The model still maximizes the tutors' preferences, however, it also takes into account the notion of time slots, making sure that a tutor is not assigned to two simultaneous workshops. We also mention the work of Tillett (1975), who introduced a zero-one programming model in which the objective function combines tutors' preferences and effectiveness. In the work of Yang and Pineno (1989), the tutors' effectiveness is measured through 15 different scores performed by a survey given to the students. Harwood and Lawless (1975), Lee and Schniederjans (1983), and Al-Husain et al. (2011) also used an advanced objective function involving positive and negative deviations from the tutors' targeted numbers of hours.

Shih and Sullivan (1977) presented a two-stage optimization approach: in the first stage, tutors are assigned to workshops maximizing their preferences; in the second stage, workshops are assigned to time slots. Hultberg and Cardoso (1997) proposed a model for the TAP that aimed at minimizing the number of different courses the tutors are assigned to. In other words, the model prefers assigning a tutor to many workshops of few courses rather than few workshops of many courses. The authors also pointed out the resemblance between their version of the TAP and the fixed charge transportation problem.

McClure and Wells (1984) introduced the first model in which tutors are assigned to a pattern of workshops. As every pattern respects the tutoring load, the model has very few constraints. However, it has a very large number of variables, as the number of tutoring patterns is exponential. They later extended their work in McClure and Wells

(1987) to allow a more advanced objective function involving positive and negative deviations from the tutors' targeted numbers of hours. Theoretical investigations on such models were provided in Brucker et al. (2011).

An ILP model for the TAP with ad hoc constraints was proposed by Al-Yakoob and Sherali (2006). Their model considers incompatibility constraints for simultaneous and successive workshops, and gender-based policies. Another model for the TAP, specifically targeted for tutors and workshops, was proposed by Al-Yakoob and Sherali (2017). Due to the exponential number of variables the latter model involved, they proposed a column generation procedure. Over the last two decades, many other mathematical and heuristic approaches have been proposed, each of them involving specific constraints related to the authors' institution. We mention in particular the work of Wang (2002) in Taiwan, Gunawan et al. (2008) in Indonesia, Güler et al. (2015) in Turkey, da Cunha Jr and de Souza (2018) in Brazil, and Domenech and Lusa (2016) in Spain.

Very recently, other problems closely related to the TAP have been studied. Ghiani et al. (2017) addressed the issue of remedial education for underprepared students that need additional support by teachers. They focused on high school remedial education, although they explicitly underlined the problem applicability to university systems. Assignment of teachers to courses and timetabling were modeled into a unique ILP formulation, and a heuristic approach was implemented to solve large size instances.

The aim of our work is to provide an ILP model able to solve exactly the TAP with constraints that are specific to our case study. Table 2.1 gives a summary of the TAP models previously mentioned, with indicators on whether or not they consider specific features. The last line is specific to our case. The following notation is used in the table:

- *Nature of the objective function*: O1: Multi-objective function solved through goal programming (hierarchical optimization); O2: Multi-objective criterion solved as a standard ILP (where each objective function is given a weighted factor); O3: Single objective function (e.g., tutors' preferences, tutors' productivity, cost of allocating a tutor to a workshop);

- *Sets of constraints*: C1: Workshops need a minimum/exact number of tutors (or tutoring hours); C2: Tutors can be assigned to a minimum and/or maximum number of workshops (and/or tutoring hours); C3: Tutors can be assigned to maximum one (pre-determined) configuration of workshops for each course; C4: Tutors are eligible for workshops (for instance in terms of knowledge, computer skills, academic position and so on); C5: Tutors cannot be assigned to two concurrent workshops; C6: Tutors cannot be assigned to two following workshops located in different sites if they don't have the time required for travel; C7: Some tutors-workshops assignments are forced by the model; C8: Some tutors-workshops assignments are forbidden by the model.

- *Domain of the decision variables*: V1: Binary decision variables; V2: Integer decision variables; V3: Non negative continuous decision variables.

Table 2.1: Literature review of the main models for the personnel assignment problem.

Model	Obj. func.			Constraints								Variables		
	O1	O2	O3	C1	C2	C3	C4	C5	C6	C7	C8	V1	V2	V3
Al-Husain et al. (2011)	✓		✓	✓	✓							✓		✓
Al-Yakoob and Sherali (2006)		✓	✓	✓	✓			✓				✓		
Al-Yakoob and Sherali (2017)			✓	✓	✓		✓	✓	✓			✓	✓	
Breslaw (1976)			✓	✓	✓			✓						✓
Brucker et al. (2011)			✓	✓		✓						✓		
da Cunha Jr and de Souza (2018)		✓		✓	✓							✓		✓
Dinkel et al. (1989)	✓		✓	✓	✓			✓						✓
Domenech and Lusa (2016)		✓	✓	✓	✓			✓				✓		✓
Dyer and Mulvey (1976)			✓	✓	✓								✓	
Ghiani et al. (2017)		✓		✓			✓	✓				✓		
Güler et al. (2015)	✓		✓	✓	✓			✓				✓		✓
Gunawan et al. (2008)		✓		✓	✓					✓		✓		
Harwood and Lawless (1975)	✓				✓			✓				✓		✓
Hill et al. (1983)			✓	✓	✓		✓					✓		
Hultberg and Cardoso (1997)			✓	✓	✓							✓		✓
Lee and Schniederjans (1983)	✓		✓	✓	✓							✓		✓
McClure and Wells (1984)			✓	✓		✓						✓		
McClure and Wells (1987)	✓		✓	✓		✓						✓		
Shih and Sullivan (1977)			✓	✓	✓							✓		
Tillett (1975)		✓	✓	✓	✓	✓						✓		
Wang (2002)			✓	✓	✓		✓					✓		
Yang and Pineno (1989)			✓	✓	✓							✓		
Our case			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		

2.3 Problem description

In this section, we describe each component of the TAP and give an exhaustive list of the constraints that need to be considered.

2.3.1 Definitions

Courses A course is a combination of lectures and workshops. Each course has a set of research groups to which it is associated and is delivered at a specific academic year (e.g., first year, second year, post-graduate). On the students' side, a course is composed of at most one lecture and one workshop per week. As the amphitheatres are large enough to accommodate hundreds of students, lectures are given only once. However, due to certain restriction such as the room size or the high number of participants, it is possible that the same workshop has to be repeated several times during the week. We identified three types of courses: small courses, with one lecture and one workshop section per week; medium courses, with one lecture and two or three workshop sections per week, generally in different days of the week; and large courses, one lecture and four or more consecutive workshop sections in the same day of the week.

As far as scheduling is concerned, some courses are proposed twice during the year (once per semester) while some others are semester-specific. Lectures and workshops are delivered either every week, every even week, or every odd week.

Lectures Each lecture is given by one lecturer and has a location, a day, and a time slot. All these data are fixed (i.e., determined before solving the TAP), so we consider them as inputs. They are used to define scheduling incompatibilities. For example, a member of staff cannot lecture a course and tutor another at the same time.

Workshops Each workshop has a *supertutor*, a location, a day, a time slot, and a set of computer skills (e.g., softwares Maple or Matlab) that are fixed, and a set of tutors that needs to be determined.

Supertutors Each workshop section has a supertutor that is fixed. The main task of the supertutor is to manage the other tutors, and possibly answer the questions tutors are not able to answer. In most cases, the lecturer of a given course is also the supertutor for the workshops of that course. Supertutoring hours are included in the tutoring hours.

Tutors Tutors refer to all the faculty members of the department (academics, postdocs, PhD students) and external members (hired as consultants or members of other departments). Tutoring is mandatory for some of the academic members (e.g., when the working contracts involve a minimum number of tutoring hours to do), while it is based on volunteering for some others (e.g., some postdocs wish to tutor to get some teaching experience).

A minimum and a maximum number of hours of tutoring are assigned to each tutor every semester. This number depends on the category in which the tutor belongs and her specific requests (e.g., some members prefer to do all their tutoring hours in one semester). In addition, the tutors have a maximum number of courses they are allowed to be involved with each semester to prevent someone to be assigned to many different small courses. For a standard tutor, with a balanced workload between semesters, the limit is set to three different courses per semester.

Each tutor belongs to one or several research groups, may have one or several computer skills, and is allowed to tutor courses from a specific set of years. In addition, tutors have a preference list of courses they wish to tutor. The preferences are expressed through the annual tutor survey, together with all the special requests in terms of availability, including sabbatical semesters and extra commitments. A tutor can only be assigned to a workshop if she has all the appropriate skills, if she has at least one of the appropriate research group, and if she is allowed to tutor at the level (year) of the course.

2.3.2 System requirements

The specific requirements of the system described in this problem are the following:

1. a tutor can be assigned to a workshop only if her research group, the years she is allowed to tutor, and her skills are consistent with those required by the workshop. Note that, as the skills depend on the workshop and not on the course, a tutor may be assigned to only certain workshops of a given course;

2. the number of required tutors for each workshop section must be respected;

3. the minimum and maximum number of hours and courses per semester for each tutor must be respected; the number of hours spent in a given workshop is the multiplication between the actual number of hours spent in that workshop and a parameter called the *Tutor Marking Multiplier (TMM)*, which is assigned to each course as a scaling factor so that certain activities are taken into account (e.g., marking);

4. each tutor can be assigned to multiple tutoring sections of the same course, but no more than three in a row;

5. workshops are delivered in two locations, i.e., two campuses; each daily movement from one campus to the other requires at least one hour to travel;

6. an academic should not be supertutor and tutor of the same course, even on different workshops;

7. the allocation should satisfy tutors' preferences as much as possible.

The administrative staff in charge of the tutor allocation is also able to give additional preferences based on historical data, remove preferences they believe are not adequate, or force/forbid a tutor to be assigned to a given course.

2.3.3 Model setting

In order to clarify the model formulation reported in the following section, we introduce the concept of *configuration* and *incompatibilities*.

Configurations A configuration is a subset of workshop sections scheduled for one specific course that respect the system requirements: it can be a single section, two or three following sections in the same day, or multiple sections in different days. In practice, (i) small-sized courses (e.g., course (S) in Table 2.2) have a unique configuration consisting in a single workshop section, (ii) medium-sized courses (e.g., course (M) in the table) have several configurations consisting in either a single workshop or a combination of workshops occurring in different days, and (iii) large-sized courses (e.g., course (B) in the table) have several configurations consisting of either two or three consecutive workshops. Feasible configurations are determined in a separate procedure called before the model. We say that a configuration is *active* at a given time slot if one workshop included in the configuration occurs at this time slot. Each tutor can be assigned to at most one configuration of each course. If a tutor cannot be assigned to a given course, then we prevent her to be assigned to any configuration of that course.

Incompatibilities We identified two types of incompatibilities: (i) for each time slot, no tutor can be assigned to more than one configuration that is active in that time slot (e.g., in Table 2.2, configuration 1 of course (S) and configurations 1 and 2 of course (M) are incompatible); (ii) a tutor cannot be assigned to two configurations that are active in consecutive time slots if they are in two different locations (e.g., in Table 2.2, configuration 1 of course (S) and configurations 1 and 2 of course (B) are incompatible if the workshops of courses (S) and (B) take place in different locations).

Table 2.2: Example of configurations generation for small (S), medium (M), and big (B) courses.

Course	Workshops	Configurations
(S)	Mon 09.00-10.00	1) Mon 09.00-10.00
(M)	Mon 09.00-10.00 Fri 14.00-15.00	1) Mon 09.00-10.00 2) Mon 09.00-10.00, Fri 14.00-15.00 3) Fri 14.00-15.00
(B)	Mon 10.00-11.00 Mon 11.00-12.00 Mon 12.00-13.00 Mon 13.00-14.00	1) Mon: 10.00-11.00, 11.00-12.00 2) Mon: 10.00-11.00, 11.00-12.00, 12.00-13.00 3) Mon: 11.00-12.00, 12.00-13.00 4) Mon: 11.00-12.00, 12.00-13.00, 13.00-14.00 5) Mon: 12.00-13.00, 13.00-14.00

Forced and forbidden configurations Some specific system requirements have to be included in the model as forced configurations. For instance, externals not working in the department are included in the list of tutors because they are needed for a specific course. Therefore, we have to make sure that they are assigned to the appropriate configuration of the course they are hired for. Similarly, we use forbidden configurations to prevent them to be assigned to any configuration of a course they were not hired to tutor. We also forbid all the configurations a tutor cannot do, either because she is lecturing during one of the time slot in which the configuration is active, or because she does not have the appropriate skills, research group, or year.

2.4 Mathematical model

In this section, an ILP model is formulated to address the TAP. The needed mathematical notation is defined in Table 2.3.

We introduce a set of two-index binary variables x_{ij} that take the value 1 if tutor i is assigned to configuration j, and 0 otherwise ($i \in I, j \in J$).

The TAP is modelled as follows:

$$\max \quad \sum_{i \in I} \sum_{j \in J} \frac{1}{\min\{m_i, U_i\}} x_{ij} \tag{2.1}$$

$$\text{s.t.} \quad \sum_{i \in I} \sum_{j \in J_s} x_{ij} = N_s \qquad s \in S \tag{2.2}$$

$$l_i \leq \sum_{j \in J} h_j t_j x_{ij} \leq u_i \qquad i \in I \tag{2.3}$$

$$L_i \leq \sum_{j \in J} x_{ij} \leq U_i \qquad i \in I \tag{2.4}$$

$$\sum_{j \in Jk} x_{ij} \leq 1 \qquad i \in I \quad k \in K \tag{2.5}$$

$$\sum_{l \in \{1,2\}} \sum_{j \in C(d,t,l)} x_{ij} \leq 1 \qquad i \in I, \quad d \in D, \quad t \in T \tag{2.6}$$

Table 2.3: Mathematical notation.

Notation	Definition
I	Set of available tutors in the semester
K	Set of courses scheduled in the semester
S	Set of courses' sections requiring tutors
J	Set of courses' configurations to be assigned to tutors
J_K	Set of possible configurations for course k
J_S	Set of configurations including section s
D	Set of teaching days in the considered semester ($D = 1, 2, \dots$)
T_{max}	Last time slot of a teaching day
T	Set of time slots in a teaching day ($T = 1, 2, \dots, T_{\max}$)
T'	Set T without the last time slot ($T' = T \setminus \{T_{\max}\}$)
T''	Set T without the last two time slots ($T'' = T \setminus \{\{T_{\max}\} \cup \{T_{\max} - 1\}\}$)
$C(d,t,l)$	Set of configurations active at time slot t of day d at location l
P_i	Set of preferences for tutor i (configurations of preferred courses)
F_i	Set of forced configurations which tutor i is forced to tutor
F_i'	Set of forbidden configurations that cannot be tutored by tutor i
l_i, u_i	Min/max number of tutoring hours per semester for tutor $i \in I$
L_i, U_i	Min/max number of allocated courses per semester for tutor $i \in I$
h_j	Total number of tutoring hours per semester for configuration $j \in J$
t_j	TMM of course which configuration $j \in J$ belongs to
N_s	Number of tutors required by each course section s
m_i	Number of preferences (courses) expressed by tutor i

$$\sum_{j \in C(d,t,1)} x_{ij} + \sum_{j \in C(d,t+1,2)} x_{ij} \leq 1 \qquad i \in I, \quad d \in D, \quad t \in T' \tag{2.7}$$

$$\sum_{j \in C(d,t,2)} x_{ij} + \sum_{j \in C(d,t+1,1)} x_{ij} \leq 1 \qquad i \in I, \quad d \in D, \quad t \in T' \tag{2.8}$$

$$\sum_{j \in C(d,t,1)} x_{ij} + \sum_{j \in C(d,t+2,2)} x_{ij} \leq 1 \qquad i \in I, \quad d \in D, \quad t \in T'' \tag{2.9}$$

$$\sum_{j \in C(d,t,2)} x_{ij} + \sum_{j \in C(d,t+2,1)} x_{ij} \leq 1 \qquad i \in I, \quad d \in D, \quad t \in T'' \tag{2.10}$$

$$x_{ij} = 1 \qquad i \in I \quad j \in F \tag{2.11}$$

$$x_{ij} = 0 \qquad i \in I \quad j \in F' \tag{2.12}$$

$$x_{ij} \in \{0,1\} \qquad i \in I \quad j \in J \tag{2.13}$$

The objective function (2.1) maximizes tutors' preferences. Tutors are free to express as many preferences as they want in the annual survey and their preference lists may be shortened or made longer by the administrative staff. Coefficients $\frac{1}{\min\{m_i, U_i\}}$ make sure that every tutor has the same weight, whether they listed a number of preferences greater than, equal to, or lower than U_i. For example, if tutor i expresses five preferences (i.e., $m_i = 5$) for a semester while her maximum number of courses for that semester is equal to three (i.e., $U_i = 3$), every satisfied preference counts for $\frac{1}{3}$ units in the objective function, so that tutor i contributes to at most 1 in the objective function. If that tutor had only expressed two preferences, every satisfied preference would have counted for $\frac{1}{2}$ in the objective function. This policy is used in order to avoid favoring a tutor based on the number of expressed preferences. Note that if tutor i had expressed zero preferences, she would not appear in the objective function as the sum over all her preferred configurations would be equal to zero (i.e., no elements to sum). Constraints (2.2) ensure that the required number of tutors is allocated to each workshop section. Constraints (2.3) and (2.4) ensure that the number of tutoring hours and courses allocated to each tutor respect the given limits. Constraints (2.5) are necessary to guarantee that each tutor is assigned to at most one configuration for each course. Constraints (2.6) represent incompatibility constraints for concurrent sections: each tutor can be at most in one location in each time slot. Constraints (2.7)-(2.10) are location incompatibility constraints. Courses are distributed in two locations in the presented model and a fixed number of time slots (2 in our case) is required for tutors to move from one location to another so they cannot be assigned to any activity during those time slots. Constraints (2.11) and (2.12) refer to forced and forbidden configurations respectively. Note that the preprocessing of most mathematical solvers automatically removes the variables set to 0 in (2.12). Finally, constraints (2.13) state that variables x_{ij} are binary.

For the sake of conciseness, the model only considers one semester, but can easily be extended to handle both semesters simultaneously by making a copy of constraints (2.3) and (2.4) per semester. Constraints related to the skills, the research group, and the academic years are included in the forbidden configurations constraints (2.12). These three aspects are related to tutors' teaching competences and play a key role in our model to provide a valuable assignment solution. Indeed, research puts a significant effort in the study of competence analytics and good results have been obtained in the literature with the use of similar mathematical approaches, as shown by, e.g., Bohlouli et al., 2017.

2.5 Computational experiments

In this section, we study the performances of model (2.1)-(2.13) on a real TAP instance (from the University of Edinburgh, School of Mathematics, academic year 2019-2020), and on randomly generated instances with larger sizes. Our model has been coded in C++ and solved with Gurobi 8.1.1. on an Intel Core i7, 1.80 GHz, with 16 GB of RAM memory, running under Windows 10 64 bits. A time limit of 3600 seconds per instance has been imposed to the solver.

2.5.1 Case study

We ran our model on two semesters, for a total of 115 days (11 weeks in the first semester and 12 weeks in the second semester). Activities are scheduled from Monday to Friday, between 9 am and 6 pm, so there are 18 time slots of 30 minutes every day. Lectures and workshops of the School of Mathematics are distributed in two campuses, one hour distant from each other (around 95% of the workshops take place in the first location, 5% in the second). Around one hundred courses are proposed in the 2019-2020 academic year, equally split among the two semesters. The TMM is usually equal to 2, but can take values between 1 and 2.5. Almost 300 tutors are available for the considered academic year. The maximum number of assignable courses per semester is equal to 3 for most tutors with rare exceptions. The maximum number of tutoring hours per semester varies in general between 80 hours and 120 hours. There are 9

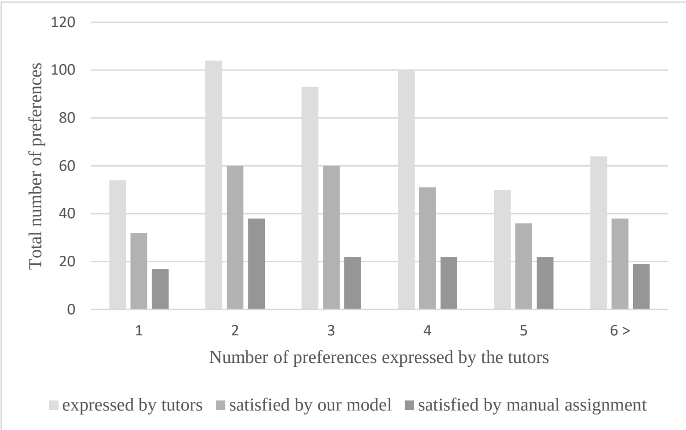


Figure 2.1: Total sum of preferences satisfied by the model solution and the manual assignment.

research groups in the School of Mathematics and each tutor belongs to one group or more. Courses are typically compatible with more than one research group. On average, a tutor is compatible with 37% of the courses and expresses 1.4 preferences.

An optimal solution was found by the model within ninety seconds. On average, tutors obtained 60% of their preferences, which is an improvement with respect to the manual solution obtained by the administrative team that only gave them 31% of their preferences. Note that only tutors who expressed at least one preference are considered when performing the indicator on proportion of satisfied preferences. Figure 2.1 shows a detailed comparison between the two solutions. Tutors are classified with respect to the number of preferences they expressed in the horizontal axis. For each group, the graph shows in the y axis the total number of preferences expressed by the tutors, the number of preferences satisfied by the solution of our model, and the number of preferences satisfied by the manual assignment. We observe an interesting pattern:

- for the groups of tutors with few preferences (1 and 2), the manual assignment is slightly worse than our model;
- for the groups of tutors with a large number of preferences (3 and 4), the solution given by the manual assignment is significantly worse than our model;
- for the groups of tutors with a very large number of preferences (5 and 6), the solution given by the manual assignment is still worse than our model, but at a lesser extent.

One possible explanation is that the administrative team tries to satisfy as much as possible the tutors expressing 1 or 2 preferences. For tutors expressing 3 or 4 preferences, the administrative team tries to satisfy at least one of their preferences, if it is possible. Tutors with 5 preferences or more seem to have been in the system for several years, and it may be possible that the administrative staff tries to give a higher priority to these "experienced" tutors.

When we compared our solution with the manual solution, we observed that only a third of the allocations were similar, indicating that there is a high interest in using an

ILP model to generate the initial solution of tutors to workshops, as this can lead to solutions that are very difficult to obtain manually. We want to underline that even if our results are very promising, the two solutions can only be compared to some extent: on one hand, the manual solution violated some constraints of the model (increasing the solution space, which gives it an advantage with respect to the ILP model), but on the other hand, the manual allocation had less tutors available as the workshops assigned to tutors who left the university after the allocation was published had to be redistributed among the remaining tutors (reducing the solution space, which gives the ILP model an advantage).

As it maximises the number of satisfied preferences, a clear downside of our approach is that the tutors who did not express any preferences do not have any impact on the objective function. As a result, we observed that tutors who expressed some preferences were allocated more tutoring hours on average than the tutors who did not express any preference (1.38 times more hours for the tutors who expressed one or more preferences with respect to those who did not). Such feature was also noticed in the manual allocation (1.14 times more hours for the tutors who expressed one or more preferences with respect to those who did not) but at a lesser extent. A similar trend was observed with the average number of allocated courses. As this is not a desirable feature (we do not want to encourage tutors to not express their preferences with the hope that they can reduce their tutoring loads), we added a set of two constraints in our next set of experiments to enforce a certain level of fairness between the two groups of tutors and we measured their impact on the total number of preferences satisfied. If we consider n_p as the number of tutors with preferences belonging to set I_p, $n_{\bar{p}}$ as the number of tutors without preferences belonging to set $I_{\bar{p}}$, and δ as the level of fairness, our additional constraints to model (2.1)-(2.13) are the following:

$$(1-\delta)n_p \sum_{i \in I_{\bar{p}}} \sum_{j \in J} x_{ij} \leq n_{\bar{p}} \sum_{i \in I_p} \sum_{j \in J} x_{ij} \leq (1+\delta) n_p \sum_{i \in I_{\bar{p}}} \sum_{j \in J} x_{ij}. \tag{2.14}$$

These constraints makes sure that the average number of hours per tutor in group I_p (i.e., the tutors who expressed 1 preference or more) does not deviate from the average number of hours per tutor in group $I_{\bar{p}}$ (i.e., the tutors who did not express any preferences) by more than a factor of δ.

Different values of δ were tested between 0.1 and 0.5, in order to reduce the disparities among the two groups of tutors (measured as $\frac{n_{\bar{p}} \sum_{i \in I_p} \sum_{j \in J} x_{ij}}{n_p \sum_{i \in I_{\bar{p}}} \sum_{j \in J} x_{ij}}$) as the ratio was 1.38 in the solution obtained by our model while it was 1.14 in the manual allocation. As observed in Table 2.4, the additional constraints were able to provide more balanced solutions among the two groups of tutors without deteriorating the objective function, but at the expense of an increase in the computing time (+96s for $\delta = 0.13$). The first column of the table indicates the value of δ, columns "time" and "obj" give the computing time and its optimal solution value, the three following columns report the average number of hours for the tutors who expressed at least one preference, the average number of hours for the tutors who did not express any preferences, and the ratio between the two values. The last three columns give the same information for the average number of courses assigned. The last row of the table (row "mod.") reports the results of the model without constraint (2.14). We point out that lower δ values were also tested, but the resulting models were either infeasible (e.g., for $\delta = 0$) or undecided

Table 2.4: Model performance measures depending on δ.

δ	time (s)	obj.	avg. nb. hours p	avg. nb. hours $\bar{p}$	avg. nb. hours $p/\bar{p}$	avg. nb. courses p	avg. nb. courses $\bar{p}$	avg. nb. courses $p/\bar{p}$
0.50	93.38	108.95	108.15	75.68	1.43	3.78	2.44	1.55
0.40	96.92	108.95	106.88	77.22	1.38	3.52	2.63	1.34
0.30	94.91	108.95	104.36	80.29	1.30	3.45	2.77	1.25
0.20	101.22	108.95	101.06	84.28	1.20	3.45	2.81	1.23
0.15	111.44	108.95	99.33	86.39	1.15	3.31	2.97	1.11
0.13	183.84	108.95	98.61	87.27	1.13	3.35	2.88	1.16
mod.	87.72	108.95	106.86	77.25	1.38	3.61	2.59	1.39

(i.e., the model was not proven to be infeasible, but no feasible solution was found) after 8 hours of computing (e.g., for $\delta = 0.1$).

2.5.2 Further analysis with the case study

When given the choice, a tutor who listed more than one course in her preference list might prefer to be assigned to one course over the others. Such feature can be captured by using ordered preference lists. In the next set of experiments, we tested the impact of ordered preference lists by replacing the weights $\frac{1}{\min\{m_i,U_i\}}$ associated with every preference from tutor i with a set of decreasing weights. For this set of experiments, we assumed that a tutor ranked the courses in her preference list from the most favorite to the least favorite. If tutor i expresses m_i preferences, the set of weights is defined as $(1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots, \frac{1}{2^{m_i-1}})$. The new objective function becomes:

$$max \sum_{i\in I}\sum_{j\in J} \frac{w_{i,c(j)}}{\sum_{k=1}^{\min\{m_i,U_i\}} \frac{1}{2^{k-1}}} x_{ij}, \tag{2.15}$$

where $w_{i,c(j)}$ is the weight given by tutor i to the course c associated with configuration j (we recall that constraints (2.5) prevent a tutor to be assigned to two configurations of the same course). Similarly to our initial model, the weights are scaled to give the same importance to every tutor who expressed at least one preference. For example, if a tutor expresses three preferences, the weights are $1, \frac{1}{2}, \frac{1}{4}$, and the scaled weights are $\frac{4}{7}, \frac{2}{7}, \frac{1}{7}$. By using objective function (2.15), we obtained an optimal solution value equals to 111.12 in 91.33 seconds of computing time. This indicates that the model can also handle ordered preference lists if necessary and that it could potentially lead to better assignments for the tutors (the overall satisfaction when the preference lists were not ordered was 108.95). We could not compare this value to the manual assignment as it is not a feature the administrative team is currently using.

One potential downside of model (2.1)-(2.13) is that, if a course is really popular for the tutors, the solution will tend to assign a small configuration of that course to many tutors in order to increase the number of satisfied preferences. As this is not a desirable feature, in the next set of experiments, we minimize first the number of courses assigned to tutors (n_c) with the following objective function:

$$\min n_c = \sum_{i\in I}\sum_{j\in J} x_{ij}. \tag{2.16}$$

Table 2.5: Model performance measures depending on $\hat{\delta}$.

$\hat{\delta}$	time (s)	obj.
0	88.33	96.18
10	92.63	103.95
20	103.03	106.53
30	104.94	108.35
40*	88.30	108.95
mod.	87.72	108.95

* The original solution value was obtained with 37 additional courses.

and then we measure how the maximum number of satisfied preferences change if the number of courses assigned to the tutors is constrained to be at most $\hat{\delta}$ away from its minimum value with the following constraint:

$$\sum_{i \in I} \sum_{j \in J} x_{ij} \leq n_c + \hat{\delta} \tag{2.17}$$

Table 2.5 shows how the computing time and the solution value vary when the number of additional courses ($\hat{\delta}$) increases. The last row reports the results of the original model without constraint (2.17). If the total number of courses assigned to tutors is kept at its minimum ($n_c = 826$ in our case), the objective value is deteriorated by roughly 12% to reach 96.18 (which corresponds to 244 satisfied preferences in the solution). If we allow the total number of courses to be $\hat{\delta}$ away from its minimum, the objective value is only deteriorated by 5% for $\hat{\delta} = 10$, 2% for $\hat{\delta} = 20$, less than 1% for $\hat{\delta} = 30$, and is unchanged when $\hat{\delta} = 37$ (which corresponds to 279 satisfied preferences in the solution). This is particularly interesting considering that the total number of courses in the unconstrained model was 1046, indicating that it is beneficial to constrain the total number of courses to stay at a reasonable distance from its minimum value, especially considering that the computing time required when this constraint is added remains acceptable (+8s on average, with a maximum of +17s for $\hat{\delta} = 30$).

Finally, we reran the model using both constraints (2.14) and (2.17) with values $\delta = 0.13$ and $\hat{\delta} = 37$ in order to obtain a solution that is relatively fair for the tutors who do not express preferences and in which we keep the total number of courses assigned to tutors close to its minimum. We obtained an optimal solution with value 108.55 (a decrease of 0.38% with respect to our initial experiment) in 2709 seconds of computing time.

2.5.3 Experiment with random instances

To gain some deeper insight on the performance and scalability of the mathematical model, we also used it to solve randomly generated instances. We designed an ad-hoc data-generator that could reproduce the structure of our real-world instance, but in which we could also vary the number of tutors and courses. With the generator, we created 50 instances of 5 different sizes (number of tutors, number of courses), namely: (250, 100), (500, 200), (750, 300), (1000, 400), and (1500, 600).

A summary of the results we obtained is reported in Table 2.6. The first three columns of the table contain the instance size (number of tutors and courses) and the

Table 2.6: Results obtained with the ILP model.

nb. tutors	nb. courses	nb. instances	nb. feasible solutions	nb. optimal solutions	avg. time (s)	avg. solution value
250	100	50	46	46	58.16	102.01
500	200	50	47	47	200.23	205.05
750	300	50	40	40	436.85	303.67
1000	400	50	39	39	746.08	408.36
1500	600	50	28	28	1749.49	614.30

number of instances, the two following columns display the number of instances for which a feasible solution (first column) or an optimal solution (second column) was found, and the two last columns indicate the average computing time (in seconds) and the average solution value computed with objective function (2.1).

We observe that: some instances are infeasible, meaning that the problem can be over-constrained sometimes; all instances were solved, some of them with a proof of infeasibility, and the remaining ones with a proven optimal solution; the model can cope even with large size instances (up to 1500 tutors and 600 courses), however, the average computing time grows rapidly with the number of tutors; on average, tutors obtain around 50% of their preferences, which is consistent with our real instance.

We note that as the number of tutors grow, the number of infeasible instances grow as well. Infeasibility is usually caused by the lower and upper limits on the number of tutoring hours per tutor, indicating that there can be an incentive to put constraints (2.3) in the objective function, as done in other works such as Al-Husain et al. (2011) and Güler et al. (2015).

We also carried out a sensitivity analysis to evaluate model behavior and robustness with respect to small changes in the values of input parameters, observing the corresponding variations in the outputs of the model.

First of all, we studied the differences in solution values by varying the maximum number of preferences expressed by the tutors. In the first experiment explained in Section 2.5.3, the number of preferences expressed by a tutor was an integer number randomly generated between 0 and 3 per semester, which corresponds to the maximum number of courses a tutor can be assigned to in each semester. Two additional experiments were performed, by generating the same instances as the first experiment, but only reducing and enlarging the maximum number of preferences per semester to two and four, respectively. The results we obtained are compared to the previous experiment as reported in Table 2.7. The solution value decreases on average by 3% with a maximum of two preferences expressed by tutors and increases on average by 4% with a maximum of four preferences.

A second analysis was executed on the number of locations, in order to detect the importance of this parameter in the model, especially focusing on how it affects the computing time. In all previous computational experiments, the number of locations was equal to two as we aimed at reproducing the real case presented in Section 2.5.1. Two additional experiments were carried out by considering one and three locations, respectively. For the instances with 3 locations, 90% of the workshops are in location 1, 5% in location 2, and 5% in location 3. Results are shown in Table 2.8, where they are compared with the results of the original experiment. While the solution value remains

Table 2.7: Sensitivity analysis on tutors' preferences.

max nb. preferences	nb. tutors	nb. courses	nb. instances	nb. feasible solutions	nb. optimal solutions	avg. time (s)	avg. solution value
2	250	100	50	46	46	60.60	98.21
	500	200	50	47	47	208.71	198.56
	750	300	50	40	40	467.75	295.56
	1000	400	50	39	39	772.44	396.62
	1500	600	50	28	28	1877.61	598.88
3	250	100	50	46	46	58.16	102.01
	500	200	50	47	47	200.23	205.05
	750	300	50	40	40	436.85	303.67
	1000	400	50	39	39	746.08	408.36
	1500	600	50	28	28	1749.49	614.30
4	250	100	50	46	46	61.51	105.88
	500	200	50	47	47	209.26	212.73
	750	300	50	40	40	441.94	316.84
	1000	400	50	39	39	839.58	425.05
	1500	600	50	28	28	1790.91	636.59

Table 2.8: Results obtained with the sensitivity analysis of number of locations.

nb. locations	nb. tutors	nb. courses	nb. instances	nb. feasible solutions	nb. optimal solutions	avg. time (s)	avg. solution value
1	250	100	50	44	44	49.74	100.25
	500	200	50	43	43	186.49	201.67
	750	300	50	41	41	389.50	305.00
	1000	400	50	36	36	676.61	404.63
	1500	600	50	31	31	1518.22	608.28
2	250	100	50	46	46	58.16	102.01
	500	200	50	47	47	200.23	205.05
	750	300	50	40	40	436.85	303.67
	1000	400	50	39	39	746.08	408.36
	1500	600	50	28	28	1749.49	614.30
3	250	100	50	46	46	83.53	101.95
	500	200	50	47	47	268.94	204.90
	750	300	50	40	40	575.87	303.48
	1000	400	50	39	39	990.40	408.08
	1500	600	50	28	28	2275.65	613.91

stable in the three experiments, the computing time varies significantly. In particular, instances with only one location are solved with an average reduction of 11% in the computing time. When switching from two to three locations, we noticed an average increment of 35% in the computing time.

2.6 Conclusion

This work deals with the Tutor Allocation Problem, which aims at assigning tutors to workshops and is a key problem faced by many universities every year. This problem is important as tutors tend to give high-quality workshops when they are assigned to courses they wish to tutor. We proposed an Integer Linear Programming model that maximizes tutors' preferences for workshops, while satisfying a large set of constraints such as location, min/max number of hours, min/max number of courses, number of required tutors per workshop, scheduling, and incompatibility constraints.

To validate the model, three kinds of computational experiments were carried out. First, the 2019/2020 problem for the School of Mathematics at the University of Edinburgh was solved, and we showed that we could satisfy 60% of the tutors' choices while respecting all constraints. This is a clear improvement from the hand-based assignment that only satisfied 31% of the tutors' choices. However, we observed some disparities in the number of hours allocated between the tutors who expressed some preferences and the tutors who did not. In order to avoid unfairness issues, we constrained the level of disparity and observed that we could still reach the same optimal solution with a fairer balance between the two groups. We also tested the impact of ordered preference lists and how restricting the total number of courses assigned to tutors deteriorate the number of satisfied preferences. For the former experiments, we observed that the model could still reach an optimal solution within a reasonable time, but it is harder to evaluate the improvements with respect to a manual assignment as it is not a feature the administrative team is using. For the latter experiments, we observed that, as one would expect, the total number of satisfied preferences and the total number of courses assigned to tutors are two conflicting objectives, and we could find a set of Pareto optimal solutions. We also combined some of our experiments in order to find a fair solution that keeps the total number of courses relatively low, and we observed that the model could still reach an optimal solution that is very close to the one obtained by the initial model, but in a much larger computing time. Then, a more complete analysis based on randomly generated data showed that our model could potentially solve instances with up to 1500 tutors and 600 courses in reasonable time. We also observed that the difficulty of the problem mainly comes from the high number of variables and constraints, so that any combinatorial optimization technique aimed at reducing the model size could be beneficial. Finally, a sensitivity analysis was performed to evaluate the model behavior with respect to two important parameters of the model. We observed a variation around 4% in the solution value adding or removing one unit to the maximum number of preferences a tutor can express per semester. A different analysis on the number of locations considered in the problem showed a good stability in the solution value yet a significant variation in the computing time. Indeed, a decrease of 11% was measured removing one location while an average increase of 35% was shown adding one location.

We conclude this work by reminding that, as the university is a dynamic environment, changes in the tutors' availability or in one or more of the course specifications could make the solution obsolete. Therefore, if a change occurred before the Integer Linear Programming solution is made public, we could simply rerun the model. If a change occurred instead after, readjustment would be more problematic as it is not an option to rerun the model and obtain a completely different solution if the tutors have been communicated a first assignment. One option could be to rerun our model with an alternative objective function with the aim of minimizing the differences between the new and the old assignments. However, the number of changes might still be significant because the model prevents any constraint violation. Thus, it is important that the administrative members continue to actively participate in the process, so that they are able to monitor any change required after the solution publication. Indeed, they have the knowledge about what constraints could possibly be relaxed in order to fix a solution without adding large perturbations, if the model was not able to do so. This represents an interesting future research direction. It would also be interesting to study multi-objective optimization techniques more in depths to find, for example, a set of solutions with an appropriate balance between the number of satisfied preferences, the total number of courses assigned to tutors, the total number of trips between the two locations that tutors must perform, and the load disparities among tutors.

Another interesting future research direction is the study of metaheuristic algorithms to be applied to the Tutor Allocation Problem as it has been done for similar problems in the literature. We mention in particular the work of Leite et al. (2019) and Hossain et al. (2019) who studied advanced heuristic techniques for the optimization of university timetabling and scheduling problems. These types of techniques could be beneficial to solve very large instances or problems characterized by additional complicating constraints.

References

Al-Husain, R., Hasan, M. K., & Al-Qaheri, H. (2011). A sequential three-stage integer goal programming (IGP) model for faculty-course-time-classroom assignments. *Informatica*, *35*(2).

Al-Yakoob, S. M., & Sherali, H. D. (2006). Mathematical programming models and algorithms for a class–faculty assignment problem. *European Journal of Operational Research*, *173*(2), 488–507.

Al-Yakoob, S. M., & Sherali, H. D. (2017). A column generation mathematical model for a teaching assistant workload assignment problem. *Informatica*, *28*(4), 583–608.

Andrew, G. M., & Collins, R. (1971). Matching faculty to courses. *College and University*.

Bohlouli, M., Mittas, N., Kakarontzas, G., Theodosiou, T., Angelis, L., & Fathi, M. (2017). Competence assessment as an expert system for human resource management: A mathematical approach. *Expert Systems with Applications*, *70*, 83–102.

Breslaw, J. A. (1976). A linear programming solution to the faculty assignment problem. *Socio-Economic Planning Sciences*, *10*(6), 227–230.

Brucker, P., Qu, R., & Burke, E. (2011). Personnel scheduling: Models and complexity. *European Journal of Operational Research*, *210*(3), 467–473.

Carter, M. W., & Laporte, G. (1998). Recent developments in practical course timetabling. In E. Burke & M. Carter (Eds.), *Practice and Theory of Automated Timetabling II* (pp. 3–19). Springer Berlin Heidelberg.

da Cunha Jr, J. J., & de Souza, M. C. (2018). A linearized model for academic staff assignment in a Brazilian university focusing on performance gain in quality indicators. *International Journal of Production Economics*, *197*, 43–51.

Dinkel, J. J., Mote, J., & Venkataramanan, M. (1989). OR practice—An efficient decision support system for academic course scheduling. *Operations Research*, *37*(6), 853–864.

Domenech, B., & Lusa, A. (2016). A MILP model for the teacher assignment problem considering teachers' preferences. *European Journal of Operational Research*, *249*(3), 1153–1160.

Dyer, J. S., & Mulvey, J. M. (1976). An integrated optimization/information system for academic departmental planning. *Management Science*, *22*(12), 1332–1341.

Ghiani, G., Manni, E., & Romano, A. (2017). Training offer selection and course timetabling for remedial education. *Computers & Industrial Engineering*, *111*, 282–288.

Güler, M. G., Keskin, M. E., Döyen, A., & Akyer, H. (2015). On teaching assistant-task assignment problem: A case study. *Computers & Industrial Engineering*, *79*, 18–26.

Gunawan, A., Ng, K., & Ong, H. (2008). A genetic algorithm for the teacher assignment problem for a university in Indonesia. *Information and Management Sciences*, *19*(1), 1–16.

Harwood, G. B., & Lawless, R. W. (1975). Optimizing organizational goals in assigning faculty teaching schedules. *Decision Sciences*, *6*(3), 513–524.

Hill, A. V., Naumann, J. D., & Chervany, N. L. (1983). SCAT and SPAT: Large-scale computer-based optimization systems for the personnel assignment problem. *Decision Sciences*, *14*(2), 207–220.

Hossain, S. I., Akhand, M., Shuvo, M., Siddique, N., & Adeli, H. (2019). Optimization of university course scheduling problem using particle swarm optimization with selective search. *Expert Systems with Applications*, *127*, 9–24.

Hultberg, T. H., & Cardoso, D. M. (1997). The teacher assignment problem: A special case of the fixed charge transportation problem. *European Journal of Operational Research*, *101*(3), 463–473.

Lee, S. M., & Schniederjans, M. J. (1983). A multicriteria assignment problem: A goal programming approach. *Interfaces*, *13*(4), 75–81.

Leite, N., Melício, F., & Rosa, A. C. (2019). A fast simulated annealing algorithm for the examination timetabling problem. *Expert Systems with Applications*, *122*, 137–151.

McClure, R. H., & Wells, C. E. (1984). A mathematical programming model for faculty course assignments. *Decision Sciences*, *15*(3), 409–420.

McClure, R. H., & Wells, C. E. (1987). Modeling multiple criteria in the faculty assignment problem. *Socio-Economic Planning Sciences*, *21*(6), 389–394.

Shih, W., & Sullivan, J. A. (1977). Dynamic course scheduling for college faculty via zero-one programming. *Decision Sciences*, *8*(4), 711–721.

Tillett, P. I. (1975). An operations research approach to the assignment of teachers to courses. *Socio-Economic Planning Sciences*, *9*(3-4), 101–104.

Wang, Y.-Z. (2002). An application of genetic algorithm methods for teacher assignment problems. *Expert Systems with Applications*, *22*(4), 295–302.

Yang, C. W., & Pineno, C. J. (1989). An improved approach to solution of the faculty assignment problem. *Socio-Economic Planning Sciences*, *23*(3), 169–177.

Chapter 3

A Mathematical Formulation for Reducing Overcrowding in Hospitals' Waiting Rooms[1]

Abstract

The COVID-19 pandemic has triggered several new measures in public and private companies to limit the spread of the virus. One of the most effective measures was shown to be social distancing, but such measure is not easy to implement for every entity, especially for hospitals. In this work, we study the case of an Italian hospital whose goal is to find the best layout of outpatient services to reduce overcrowding in the waiting rooms. We propose an Integer Linear Programming model to identify the weekly optimal layout and we test it on a set of real data from the year 2019. The results obtained by our model reduce the overcrowding by 80% on average with respect to the results obtained with the configuration used by the hospital, but such results can only be obtained if the layout is allowed to change every week. We then study the case in which we force the layout to be fixed for two or three consecutive weeks and outline that both the computational time and the solution quality worsen significantly.

3.1 Introduction

Among the measures introduced to prevent the spread of COVID-19, social distancing is probably one of the most effective but comes with many logistical challenges. While companies can often enforce the 1.5-meters rule by reducing the number of people (workers and customers) in their premises, public entities such as hospitals must find other ways to limit physical interactions. In this work, we study the layout design of an Italian hospital (i.e., the room allocation) in order to avoid over-capacitated waiting rooms.

[1]The results of this chapter appear in: Caselli, G., De Santis, D., Delorme, M., Iori, M. (2021). A Mathematical Formulation for Reducing Overcrowding in Hospitals' Waiting Rooms. In *Proceedings of 2021 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)* (pp. 297-301). IEEE.

In the considered problem, each doctor delivers a set of visits during the week, which is also called an *agenda*. Every visit takes place at a given *time slot* of a given *day* and may require a specific machine (e.g., an X-ray machine for an agenda in radiology). Each visit takes place in a *clinic room*, which is characterized by its location in the hospital and its equipment. Patients wait for their appointment in a *waiting room*, which is characterized by its location and its capacity. The aim of the problem is to find an assignment of agendas to clinic rooms and clinic rooms to waiting rooms that minimizes overcrowding (i.e., the number of patients over the waiting room capacity) while satisfying a set of logistical constraints.

The logistical constraints considered in this work are the following: (i) an agenda must be assigned to exactly one clinic room for the entire time horizon, (ii) a clinic room can be assigned to at most one waiting room for the entire time horizon, (iii) an agenda can only be assigned to a subset of clinic rooms (determined by the equipment and the location of the room), and (iv) a clinic room can only be assigned to a subset of waiting rooms (determined by the location of the rooms).

Even though our problem has some similarities with well-known combinatorial optimization problems (e.g., the vector packing problem) and with existing healthcare optimization problems (e.g., operating room planning and outpatient appointment system), to the best of our knowledge, it was never studied in the literature. Indeed, its objective function (minimizing overcrowding in hospital common spaces) originates from the recent COVID-19 emergency and its ad hoc constraints are hospital-specific (the same way as the constraints of a teacher allocation problem are university-specific, see Caselli et al., 2022).

Considering these remarks, we provide in the following a new Integer Linear Programming (ILP) formulation to solve our problem. We test the model with a real-world data set provided by the hospital which contains the agendas of the 53 weeks of year 2019 and we solve the problem with different time horizon lengths.

3.2 Literature review

The literature dedicated to optimization in healthcare is large. We mention the survey by Rais and Viana (2011) who identified and categorized the scientific literature into four categories, namely healthcare planning (e.g., for determining the optimal location of healthcare centers), healthcare management and logistics (e.g., for determining the best allocation of shifts to nurses), treatment planning (e.g, for designing the optimal radiation therapy), and specialized and preventive healthcare (e.g., for maximizing the number of kidney transplants). The problem we study belongs to the category "healthcare and logistics". In the following, we outline optimization problems that are related to our room allocation problem.

Our problem has some similarities with operating room (OR) planning and scheduling (see Cardoen et al., 2010 for a recent survey). For example, Santibáñez et al. (2007) developed an ILP model to schedule surgical blocks for each specialty, considering OR time availability and resource constraints. Each surgical block had to be assigned to an OR that was compatible with its specialty and the assignment was fixed for the entire time horizon. Unlike our work however, capacity was considered as a "hard constraint" and could not be exceeded. In Denton et al. (2010), the authors introduced stochastic optimization models for the allocation of surgery agendas to ORs. Their objective was to minimize the costs that comprised a fixed component to open an OR and a variable component to use the OR longer than its available time. In our case, we do not consider that using a room (either a waiting room or a clinic room) has a cost.

Our problem has also some similarities with the Outpatient Appointment Systems (OASs) for which we refer the interested reader to the recent survey by Ahmadi-Javid et al. (2017). For example, Wang and Gupta (2011) designed an OAS able to learn the patient's preferences to improve booking decisions. They used an ILP model to assign appointments to clinic room, with the objective of maximizing the overall patient satisfaction. Another relevant study was proposed by Green et al. (2006) who optimized patient management in a hospital diagnostic facility. They identified three categories of patients: (i) the outpatients, whose appointments are scheduled weeks in advance, (ii) the inpatients, whose appointments are scheduled the same day, and (iii) the emergency patients, who do not have any appointments, but must be taken care of as soon as possible. The authors studied the design of dynamic priority rules for admitting patients into service in real time.

Finally, we refer to the vector packing problem (see Brandao and Pedroso, 2016) where one wants to pack a set of items, each of them having a set of weights, into the minimum number of capacitated bins. The agendas are the items, the weights are the agenda occupancy per time slots, the waiting rooms are the bins, and the clinic rooms serve as an intermediary level that is both an item and a bin (an agenda is "packed" into a clinic room which itself is packed into a waiting room).

3.3 Problem description and mathematical model

We consider a set of agendas $i \in I$, a set of clinic rooms $j \in J$, and a set of waiting rooms $k \in K$ with capacity C_k. Each agenda must be assigned to exactly one clinic room, and each clinic room must be assigned to exactly one waiting room. The assignments are fixed through the entire time horizon. This time horizon is divided into days $d \in D$, which are themselves divided into time slots $t \in T$.

For each agenda i, we define an integer matrix p_{idt} that indicates the number of people that are expected in a waiting room in each time slot t of each day d. Note that this matrix is not necessarily binary as some patients (e.g., children or elders) must be accompanied. We also introduce a binary matrix b_{idt} indicating whether agenda i is active or not at time slot t of day d. Note that in our case, if b_{idt} is equal to 0, then p_{idt} is also equal to 0. The opposite is not necessarily true: for example, if an agenda is active at time slots $t-1$ and $t+1$, then it is also considered active at time slot t, even if no patients are expected. Indeed, it is not practical for a doctor to leave the clinic room to another doctor for a single time slot.

Because of logistical constraints, we distinguish three types of agendas: (i) the *fixed* agendas $i \in I_F$ requiring heavy machinery (e.g., X-rays) that can only be assigned to a small subset of clinic rooms, (ii) the *semi-fixed* agendas requiring non-heavy machinery (i.e., that can be moved occasionally from one clinic room to another such as gynecologist tables) that must be assigned to a clinic room in a specific hospital floor, and (iii) the *free* agendas requiring no machinery at all (e.g., general practitioner consultancy) that can be assigned to any clinic room. Fixed and semi-fixed agendas cannot be assigned to the same clinic room as fixed agendas. We use binary compatibility matrix Q_{ij} taking value 1 if agenda i is compatible with clinic room j (i.e., if the clinic room has the suitable machinery and is located in the suitable floor). We use binary compatibility matrix Q'_{jk} taking value 1 if clinic room j is compatible with waiting room k (i.e., if the waiting room is located in the same floor and at a reasonable distance from the clinic room). In our model, we use five sets of variables: (i) binary variable x^J_{ij} taking value 1 if agenda i is assigned to clinic room j, (ii) binary variable x^K_{ik} taking value 1 if agenda i is assigned to waiting room k, (iii) binary variable y_{jk} taking value 1 if clinic room j is assigned to waiting room k, (iv) integer variable Δ_{kdt} indicating the overcrowding of waiting room k at time slot t of day d, and (v) binary variable z_j taking value 1 if clinic room j is used for fixed agendas and value 0 if it is used for semi-fixed and free agendas. Our problem can be modeled as follows:

$$\min \quad \sum_{k\in K}\sum_{d\in D}\sum_{t\in T}\Delta_{kdt} \tag{3.1}$$

$$\text{s.t.} \quad \sum_{i\in I} p_{idt}x^K_{ik} \le C_k + \Delta_{kdt} \qquad k\in K, d\in D, t\in T \tag{3.2}$$

$$\sum_{j\in J} x^J_{ij} = 1 \qquad i\in I \tag{3.3}$$

$$\sum_{k\in K} x^K_{ik} = 1 \qquad i\in I \tag{3.4}$$

$$\sum_{k\in K} y_{jk} \le 1 \qquad j\in J \tag{3.5}$$

$$\sum_{i\in I} b_{idt}x^J_{ij} \le 1 \qquad j\in J, d\in D, t\in T \tag{3.6}$$

$$x^J_{ij} \le Q_{ij} \qquad i\in I,\quad j\in J \tag{3.7}$$

$$y_{jk} \le Q'_{jk} \qquad j\in J,\quad k\in K \tag{3.8}$$

$$x^J_{ij} \le z_j \qquad i\in I_F,\quad j\in J \tag{3.9}$$

$$x^J_{ij} \le 1 - z_j \qquad i\in I/I_F,\quad j\in J \tag{3.10}$$

$$x^J_{ij} + x^K_{ik} - 1 = y_{jk} \qquad i\in I,\quad j\in J,\quad k\in K \tag{3.11}$$

$$x^J_{ij} \in \{0,1\} \qquad i\in I,\quad j\in J \tag{3.12}$$

$$x^K_{ik} \in \{0,1\} \qquad i\in I,\quad k\in K \tag{3.13}$$

$$y_{jk} \in \{0,1\} \qquad j\in J,\quad k\in K \tag{3.14}$$

$$z_j \in \{0,1\} \qquad j\in J \tag{3.15}$$

$$\Delta_{kdt} \ge 0, integer \qquad k\in K,\quad d\in D,\quad t\in T \tag{3.16}$$

The objective function (3.1) minimizes the total overcrowding. Constraints (3.2) compute the overcrowding Δ_{kdt} for each waiting room at each time slot of each day. Constraints (3.3) and (3.4) make sure that each agenda is assigned to exactly one clinic room and one waiting room, constraints (3.5) ensure that each clinic room is assigned to at most one waiting room. Constraints (3.6) prevent two conflicting agendas (i.e., two agendas which are both active at the same time slot of the same day) to be assigned to the same clinic room. Constraints (3.7) make sure that each agenda is assigned to a compatible clinic room and constraints (3.8) make sure that each clinic room is assigned to a compatible waiting room. Constraints (3.9) and (3.10) prevent free and semi-fixed agendas to be assigned to a clinic room j if it is used for fixed agendas (i.e., if z_j is equal to 1). Constraints (3.11) are used to relate the three families of variables x^J_{ij}, x^K_{ik} and y_{jk}. Constraints (3.12)-(3.15) state that the variables $x^J_{ij}, x^K_{ik}, y_{jk}, z_j$ are binary. Finally, constraints (3.16) state that the variables Δ_{kdt} are non-negative integers.

We point out that a preliminary version of model (3.1)-(3.16) using a set of 3 index variables x_{ijk} (taking value 1 if agenda i is assigned to clinic room j and waiting room k) instead of three sets of 2-index variables x^J_{ij}, x^K_{ik} and y_{jk} was tested but exhibited a worse computational behavior. We also underline that constraints (3.7) and (3.8) belong to the model for the sake of clarity: in practice, the variables set to 0 by the two constraints were not generated in our implementation (they would have been removed by the preprocessing of any state-of-the-art ILP solver anyways).

3.4 Computational experiments

We implemented model (3.1)-(3.16) using Python 3.9 as programming language and Gurobi 9.1.2. as ILP solver. The experiments were run on a virtual machine Intel(R) Xeon(R) Gold 6130 with 2.1 GHz and 16 GB of RAM memory, with Windows Server 2019 Standard as operating system.

In our first set of computational experiments, we studied the performance of the model on real-world instances from an Italian hospital. We ran the model for each of the 53 weeks of 2019, imposing a time limit of 3600s. Each instance is composed of 483 agendas spread over 34 specialties (e.g., angiology, cardiology, dermatology, radiology), 119 clinic rooms, and 31 waiting rooms. The integer matrix indicating the number of patients for each week, each day, each time slot, and each agenda and the corresponding activity matrix b were given by the hospital. In the following, we provide relevant statistics from our real-world data:

- On average, an agenda has 16 patients per week and it is active 21 time slots per week (we recall that an agenda can be active at a time slot even if no patients are expected: this prevents another agenda to occupy the same clinic room during the short period of inactivity).
- An agenda is compatible with 56 clinic rooms on average.
- A clinic room is compatible with 227 agendas and 2 waiting rooms on average.
- A waiting room is compatible with 9 clinic rooms on average.

The hospital has four floors, and each floor is decomposed into several blocks where clinic rooms and waiting rooms are located. The distribution of clinic rooms and waiting rooms per floor is relatively even (i.e., there are around 30 clinic rooms and 8 waiting rooms in each floor). The location of each waiting room and its capacity was provided by the hospital. The capacities range from 1 to 16, with an average of 4 spots available.

Out of the 53 tested instances, 52 could be solved to optimality within the time limit. The average CPU time required was 660s (we used 3600s as the time required for the unsolved instance). The solution value found by the model and the computing

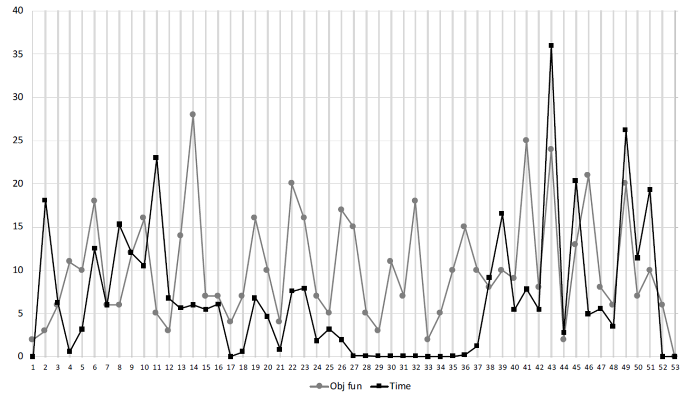


Figure 3.1: Solution value and computation time required by the ILP model for each of the 53 tested weeks.

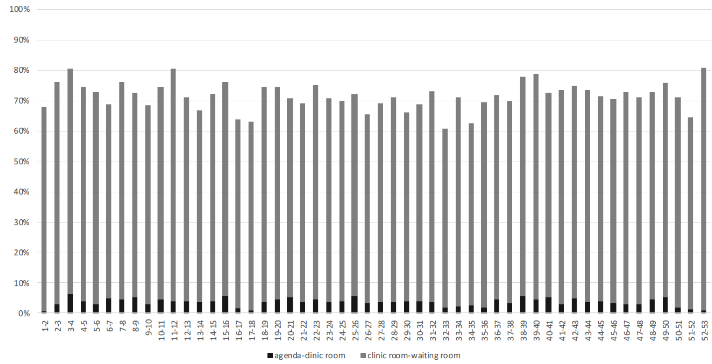


Figure 3.2: Proportion of agenda-clinic room and clinic room-waiting room allocations kept stable from one week to the other.

time for each week are summarized in Figure 3.1. We observe that the computational time is highly correlated with the period of the year: for holiday weeks (e.g., Christmas, Easter, and during the summer), the time to solve an instance is very low (a few dozens of seconds): this can be explained by the fact that there are less patients during these weeks, which results in smaller size models as there are less non-zero values in constraints (3.2) and (3.6). In the opposite, during the most active weeks, (i.e., where the hospital is full of appointments such as in October), the time required to solve an instance often goes above 1000s. For the unsolved instance (week 43), the solution provided by the solver has a noticeable relative gap ($UB - LB = 19$), but the solution found by the solver ($UB = 24$) is acceptable when compared to other weeks.

On average, we obtained solutions that were from 70% to 90% below those observed by the hospital with its manual allocation. This indicates that there is an interest to use optimisation techniques to allocate agendas to clinic rooms and waiting rooms. These good results should be put in perspective however, as in its current allocation, the hospital tends to avoid any major shift in the configurations from one week to the other, which is not the case in our model. Indeed, we observed from Figure 3.2. that, on average, the proportion of assignments kept stable from one week to the other is only 4% for the agenda-clinic room allocations and 68% fort the clinic room-waiting room allocations.

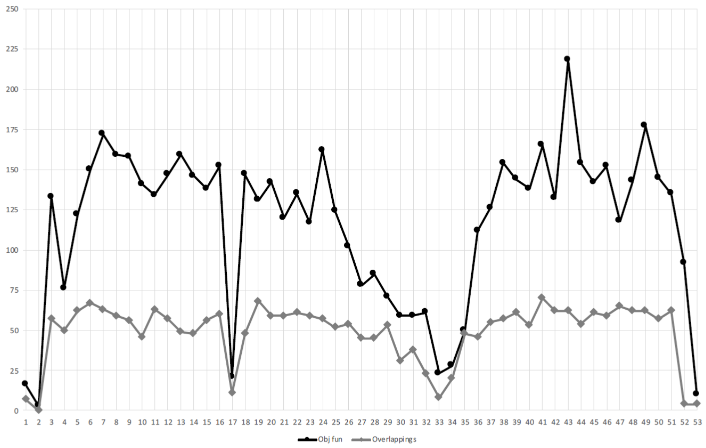


Figure 3.3: Solution value and number of overlapping for each of the 53 tested weeks if the allocation found for week 2 is used in other weeks.

Even if the hospital is not against having a new agenda/clinic room and clinic room/waiting room allocation every week, this would require a very reactive management system. In addition, it is not realistic to impose such a drastic change to the doctors who have mostly used the same clinic room for a given agenda every week. We took this aspect into account in our second set of experiments in which we evaluated how good the allocation found in a given week would be if it was maintained in the other weeks. We decided to pick the configuration from week 2 as the solution from week 1 fall in Christmas holidays. We used two indicators to evaluate the fitness of this allocation: the overcrowding in the waiting room as in our first set of experiments, but also the average number of overlapping (or conflicts). When two agendas who are active during the same time slot are assigned to the same clinic room, this results in an overlapping. The solution value found by the model and the number of overlapping for each week are summarized in Figure 3.3.

The average overcrowding for the whole year when using the configuration of week 2 is 116 on average, which is closer to the level of overcrowding observed by the hospital in practice. In addition, we identified 50 conflicts on average, indicating that besides having significantly more overcrowding, the best configuration found in a given week cannot directly be used in the other weeks. We also underline that more than 50 conflicts and an overcrowding of above 100 were observed in week 3, indicating that even two successive weeks have significant agenda differences.

We tried to take this aspect into consideration in our third set of experiments by considering a longer time horizon. Our set D now contains 14 days (i.e., 2 weeks) while it was 7 days in the previous experiments. Model (3.1)-(3.16) now provides a unique configuration that is valid for both weeks that minimizes the overcrowding happening during the 14 days of the time horizon. The model is significantly bigger as it now contains twice as many Δ_{kdt} variables and twice as many incompatibility constraints (3.6). As for the previous experiments, we set a time limit of 3600s. The results are summarized in Figure 3.4.

When we consider two weeks simultaneously, the computation time required to solve an instance increased from 660s on average to 2470s (we still used 3600s as the time required for the unsolved instances). Only 9 instances out of 26 could be solved to

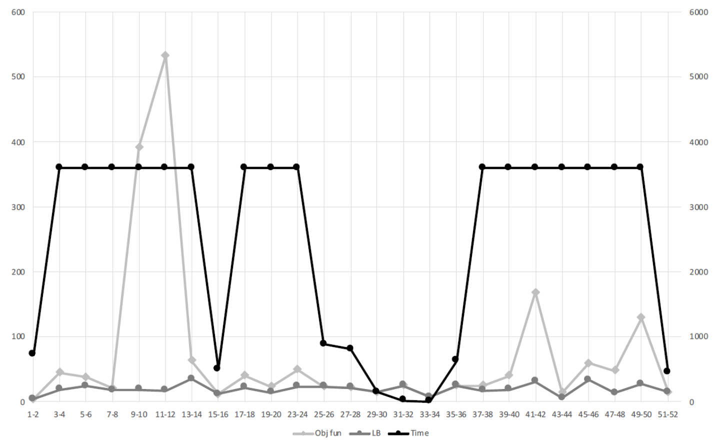


Figure 3.4: Solution value, lower bound, and computation time required by the ILP model for each of the 26 pairs of weeks.

optimality within the time limit. We observed that the average overcrowding during these 9 pairs of weeks was 12 while it was 8 for the corresponding 18 weeks in our first set of experiments. Thus, keeping the configuration fixed has a significant impact on the minimum overcrowding. For the instances that could not be solved to optimality but that were feasible, the average absolute gap between the upper and lower bound was 84 (56 if we exclude the outsider pair of weeks 11 and 12) and the average relative gap was 56,45% (53,76% if we exclude the outsider). We identified several possible causes for the increase in computing time: (i) the increase in the model size due to the additional Δ_{kdt} variables and incompatibility constraints (3.6), and (ii) the difficulty for the solver to find good quality solutions. We also point out that one instance was infeasible (the one corresponding to weeks 21 and 22). This indicates that the ideal goal of finding a unique fixed configuration for the whole year is not reachable.

We also noticed some similarities with our first set of experiments, namely that the instances that could be solved to optimality correspond to holiday weeks, which are periods in which the hospital has less appointments. Finally, we also mention that we tried similar experiments with a 3-week time horizon, but the results were significantly worse (both in terms of feasibility and average computing time) indicating that other techniques should be developed if we want to have a relatively steady configuration for longer time horizons.

3.5 Conclusions and future research directions

In this work, we studied the layout design of an Italian hospital with the aim of reducing overcrowding, an objective that is particularly important during the COVID-19 pandemic. We proposed an integer programming model to assign agendas to clinic rooms and waiting rooms, and we showed through extensive computational experiments based on a real-world data set that if the layout can be changed every week, a significant decrease in the overcrowding can be reached (between 70% and 90%). We also showed that the optimal layout for one week of the year might not be necessarily good or even feasible for another week of the year, even if we consider two successive weeks. We concluded our experiments by solving the model with time horizons of two weeks and found out that solving the model required more computing time, that a feasible solution was not guaranteed to exist, and that the overcrowding increased significantly, meaning that there is a clear trade-off between keeping the layout identical over the weeks and minimizing overcrowding.

As a follow-up of this work, we would like to enhance the performance of our model so that it becomes able to solve instances with larger time horizon (a trimester, a semester, or a full year). We would also like to modify its constraints so that it allows a given number of changes from one week to the other. We will also investigate decomposition techniques where the allocation of agendas to waiting rooms is done at a first stage while the allocation of agendas to clinic rooms is done at a later stage. Finally, we will also study the quality of matheuristic approaches where one does the allocation floor by floor or even block by block.

References

Ahmadi-Javid, A., Jalali, Z., & Klassen, K. J. (2017). Outpatient appointment systems in healthcare: A review of optimization studies. *European Journal of Operational Research*, *258*(1), 3–34.

Brandao, F., & Pedroso, J. P. (2016). Bin packing and related problems: General arc-flow formulation with graph compression. *Computers & Operations Research*, *69*, 56–67.

Cardoen, B., Demeulemeester, E., & Beliën, J. (2010). Operating room planning and scheduling: A literature review. *European Journal of Operational Research*, *201*(3), 921–932.

Caselli, G., Delorme, M., & Iori, M. (2022). Integer linear programming for the tutor allocation problem: A practical case in a british university. *Expert Systems with Applications*, *187*, 115967.

Denton, B. T., Miller, A. J., Balasubramanian, H. J., & Huschka, T. R. (2010). Optimal allocation of surgery blocks to operating rooms under uncertainty. *Operations Research*, *58*(4-part-1), 802–816.

Green, L. V., Savin, S., & Wang, B. (2006). Managing patient service in a diagnostic medical facility. *Operations Research*, *54*(1), 11–25.

Rais, A., & Viana, A. (2011). Operations research in healthcare: A survey. *International Transactions in Operational Research*, *18*(1), 1–31.

Santibáñez, P., Begen, M., & Atkins, D. (2007). Surgical block scheduling in a system of hospitals: An application to resource and wait list management in a british columbia health authority. *Health Care Management Science*, *10*(3), 269–282.

Wang, W.-Y., & Gupta, D. (2011). Adaptive appointment systems with patient preferences. *Manufacturing & Service Operations Management*, *13*(3), 373–389.

Chapter 4

Exact Algorithms for a Parallel Machine Scheduling Problem with Workforce and Contiguity Constraints[1]

Abstract

We consider a real-world scheduling problem where the objective is to allocate a set of tasks to a set of machines and to a set of workers in such a way that the total weighted tardiness is minimized. Our case study takes into account four types of constraints: precedence, resource, eligibility, and contiguity. While the first three are common in the scheduling literature, contiguity constraints, which can be defined as a form of precedence constraints that requires both a predecessor and its successor to be processed on the same machine with no intermediate jobs in-between (but idle time is allowed), have never been studied in the literature. We present four exact methods to solve the problem: two methods use integer linear programming, one uses constraint programming, and one uses a combinatorial Benders' decomposition. We introduce method-specific strategies to model the contiguity constraints for each of the proposed methods. We empirically evaluate, through an extensive set of computational experiments, the performance of the four methods on a heterogeneous dataset composed of real, realistic, and random instances, and outline that every method offers a competitive advantage on a targeted subset of instances. We also show that our algorithms can be generalized to solve related scheduling problems with contiguity constraints.

4.1 Introduction

Owing to the numerous practical applications and theoretical properties, the field of Parallel Machines Scheduling Problems (PMSP) is one of the most studied in combinatorial optimization. In a PMSP, one wants to allocate a set of jobs to a set of machines, and determine, for each machine, the order in which the jobs should be processed (see, e.g., Baker and Trietsch, 2019). Some PMSPs can be solved in

[1]The results of this chapter appear in: Caselli, G., Delorme, M., Iori, M., Magni, C. A. (2024). Exact algorithms for a Parallel Machine Scheduling Problem with workforce and contiguity constraints. Computers & Operations Research, 163, 106484.

polynomial time, but most of them are $\mathcal{NP}$-hard (see Pinedo, 2016). Over the years, many additional features have been introduced and studied in PMSPs, by considering the characteristics of the machines (e.g., machine speed), of the jobs (e.g., precedence constraints), and of the objective function (e.g., sum of completion time minimization). A particularly important feature in real-world applications is resource consumption. In a Resource-Constrained Parallel Machine Scheduling Problem (RCPMSP), one also considers a set of resources that must be used in order to process a job on a machine.

Even though RCPMSPs have been extensively studied in the literature (see Edis et al., 2013 for a complete survey; see also Section 4.2 below), there are practical applications that have not yet been modeled in the literature because they either involve a novel combination of features or because they require a type of constraints that was never studied before. Our work deals with the latter case and originates from a collaboration with the engineering test laboratory of Dana Inc., a company working in the hydraulic automation industry, where complex and customized components of motion systems (such as pumps, motors, and valves) require several tests (e.g., performance or endurance tests) before being delivered to the customers.

The decision problem faced by the company can be modeled as an RCPMSP where the tests (called jobs in the remainder of the paper to stick with the most common scheduling notation) are assigned to machines and workers by meeting a number of operational constraints. The job occupies the machine entirely for a fixed number of days, and for each day the worker is required to process the job for a few hours. A limited number of workers is available every day; each of them can process one or more jobs, without exceeding their daily maximum number of hours.

The problem also requires to consider three sets of eligibility constraints: (i) a job can only be performed by a subset of workers (for example, a test on a valve must be done by a worker who is an expert in valves); (ii) a job can only be performed by a subset of machines (for example, a performance test on a valve must be done on a machine dedicated to valves performance tests); (iii) a worker can only manipulate a subset of machines (in our case, a worker can only use a machine for which they have a skill certification). In addition, every job has a release date, a duration, a desired due date, and a weight (expressing the priority of the job). The objective is to minimize the total weighted tardiness.

The main novelty of our application comes from the relations existing between pairs of jobs. We consider (i) standard precedence constraints that require a predecessor to be finished before its successor starts and (ii) contiguity constraints, which are precedence constraints that additionally require both a predecessor and its successor to be processed on the same machine with no intermediate jobs in-between (but idle time is allowed). As defined by the company, precedence relations occur between two types of jobs that must be performed on the same product (such as a short performance test followed by a long endurance test on the same hydraulic component), whereas contiguity relations take place on a chain of similar jobs that must be performed on the same product and on the same machine (such as consecutive performance tests on the same hydraulic component with different system conditions, like the oil pressure level for example), but not necessarily by the same worker. To the best of our knowledge, contiguity constraints were never studied in the RCPMSP literature. Indeed, as reported by Khatami et al. (2020), there is a research gap in the area of RCPMSPs with non-ordinary precedence constraints (such as "coupled tasks" in Khatami et al.'s paper or contiguous tasks in our case).

In this work, our first objective is to provide a set of exact algorithms able to solve real instances provided by the company. Inspired by the recent trends in the scheduling literature (see Fang et al., 2021; Koné et al., 2011; Lovato et al., 2022), we propose (i) a descriptive Mixed Integer Linear Programming (MILP) formulation, (ii) an enhanced MILP model, (iii) a new Constraint Programming (CP) formulation, and (iv) a new combinatorial Benders' decomposition that first determines the starting date of every job and then finds a suitable machine and a suitable worker to process each job at the scheduled time. For each method, we also introduce an innovative strategy to model contiguity constraints. Another objective is to determine the effectiveness of our techniques on large size realistic instances (i.e., randomly generated instances aimed at mimicking the dataset provided by the company) and on artificial instances studying the impact of key features such as the presence of precedence constraints, the presence of eligibility constraints, and various worker/machine/job ratios. These experiments show that the effectiveness of the proposed approaches depends on the instance features and, therefore, that no method consistently outperforms the others. Our contribution is not only limited to the problem considered in this work: we show that our algorithms can easily be generalized to model other scheduling problems with contiguity constraints. This can be useful if, for example, Dana Inc. prefers minimizing the total weighted completion time or the total weighted flow time in the future.

The rest of the paper is organized as follows. In Section 4.2, we provide a review of the RCPMSP literature related to our problem. In Section 4.3, we give a detailed description of our RCPMSP and introduce the descriptive MILP formulation. The enhanced MILP formulation, the CP model, and the decomposition approach are described in Section 4.4. Section 4.5 gives a summary of the results obtained by an extensive set of computational experiments. Finally, concluding remarks are provided in Section 4.6. A preliminary version of this work in which an MILP formulation was used to solve a single instance with unrelated parallel machines was presented as Caselli et al. (2022).

4.2 Literature review

Scheduling problems on parallel machines have been extensively studied since the early fifties. We refer to Mokotoff (2001) for a survey on PMSPs, to Pinedo (2016) for a comprehensive review of the scheduling theory, and to Fuchigami and Rangel (2018) for a discussion on analysis and perspectives on scheduling case studies. Following the three-field notation by Graham et al. (1979), our problem can be denoted as $P|res1,\mathcal{M}_j, r_j,d_j,prec,cont|\sum w_jT_j$. Specifically, we aim to minimize the total weighted tardiness ("$\sum w_jT_j$") on identical parallel machines ("P"), with an additional resource ("$res1$"), eligibility constraints ("$\mathcal{M}_j$"), release dates ("r_j"), due dates ("d_j"), and precedence ("*prec*") and contiguity ("*cont*") constraints. To the extent of our knowledge, characteristic "*cont*" has never been formally defined in the three-field notation. In the following, we briefly review closely related scheduling problems and their applications.

Additional resources ($res1$) As stressed by Blazewicz et al. (1983), most RCPMSPs are $\mathcal{NP}$-hard, even though some of them with unit-length jobs are polynomially solvable (see also Lawler et al., 1993 and Ventura and Kim, 2000). Various kinds of additional resources were considered in the literature, including human workforce (Seifi et al., 2021, Edis and Ozkarahan, 2011), tools (Ventura and Kim, 2000 and Hall et al., 2000), and automated guided vehicles (Ulusoy et al., 1997 and Reddy et al., 2022). For a complete survey on PMSPs with additional resources, we refer the reader to Edis et al. (2013). We only mention the work of Edis and Ozkarahan (2012), who used solution methods similar to the ones we propose in this paper. The authors solved a real-world RCPMSP with identical machines by using a two-phase algorithm that first assigns the jobs to the machines and then finds the optimal sequence of jobs on every machine. The first phase was solved with a MILP formulation while both CP and MILP were tested to solve the second phase. The two resulting algorithms were run on realistic instances with up to 100 jobs, 36 machines, and 12 workers, and consistently outperformed the original MILP model.

We also point out that in some RCPMSPs, known as parallel machine flexible-resource scheduling problems, resources can be used to speed-up the processing time of a job (see Daniels et al., 1997 and Chen, 2004). This is not the case in our problem because assigning two workers to perform a job does not shorten the job processing time.

Eligibility constraints ($\mathcal{M}_j$) Eligibility constraints are used in many practical applications where jobs must be processed on a specific subset of machines. For a complete survey on PMSPs with eligibility constraints, we refer the reader to Leung and Li (2016). Problems closely related to ours were studied by Afzalirad and Rezaeian (2016) and Edis and Ozkarahan (2011). The former introduced metaheuristics to solve a PMSP with unrelated machines, several additional resources, eligibility constraints, and precedence constraints. The latter used a combination of CP and MILP to solve a PMSP with identical machines, one additional resource, and eligibility constraints. To the best of our knowledge, our work is the first in the RCPMSP literature dealing with multiple eligibility constraints simultaneously: worker/machine, worker/job, and job/machine. We point out, however, that such multiple eligibility constraints could be modeled within the framework of the multi-mode resource-constrained project scheduling problem, with one resource per machine and per worker, and, for each job, one mode per compatible (i.e., eligible) machine/worker pair. For an extensive literature review on multi-mode resource-constrained project scheduling problems, we refer the reader to Section 2.4 of the recent survey by Hartmann and Briskorn (2022) focusing on resource-constrained project scheduling problem extensions, and to Section 5 of the survey by Węglarz et al. (2011) focusing on multi-mode project scheduling problems.

Due dates and total weighted tardiness minimization ($d_j|\sum w_jT_j$) Total weighted tardiness is one of the most common objective functions in the scheduling literature, because it models many practical problems in industrial engineering and production management (see, e.g., Cheng and Sin, 1990). We refer the reader to Koulamas (1994) for a survey focused on total tardiness minimization in scheduling problems and to Janiak et al. (2015) for a survey on scheduling problems with due windows. We also mention the relevant work of Su et al. (2017), who studied a PMSP with eligibility constraints where the objective was to minimize the total weighted tardiness, and the recent work of A. Kramer et al. (2019), who proposed MILP formulations for a PMSP where the objective is to minimize the total weighted completion time (equivalent to the total weighted tardiness if all due dates were set to 0).

Release dates (r_j) The addition of release dates for jobs tends to make PMSPs harder to solve. Graham et al. (1979) showed that $1|r_j, d_j| \sum w_j T_j$ is $\mathcal{NP}$-hard while $1|d_j| \sum w_j T_j$ can be solved in polynomial time. We refer the reader to Lee (2004) for a survey on machine scheduling problems with availability constraints. Several recent works include release dates in their PMSPs (see, e.g., Afzalirad and Rezaeian, 2016 and Li et al., 2022).

Precedence relations between jobs ($prec$) The complexity of scheduling problems with precedence constraints has been studied since the seventies (see Lenstra and Rinnooy Kan, 1978). Chudak and Shmoys (1999) introduced approximation algorithms for the PMSP with precedence constraints where each machine has its own speed. Hu et al. (2010) proposed a heuristic for a realistic PMSP with precedence and eligibility constraints. An extended literature on precedence constraints is also available in the area of assembly line balancing problems (Becker and Scholl, 2006), bin packing (R. Kramer et al., 2017), and (resource-constrained) project scheduling problems (Hartmann and Briskorn, 2022).

Contiguity constraints ($cont$) Contiguity constraints can be seen as a special form of precedence constraints where the successor of a job must be processed on the same machine as its predecessor and without any intermediate jobs in-between (but idle time is allowed). The roots of contiguity constraints can be found in batch scheduling problems (see Potts and Kovalyov, 2000 for an extensive survey) and scheduling problems with set-up times and changeover costs (see Bruno and Downey, 1978), where the concept of families of jobs that must be performed together was introduced. In recent years, concepts that are related to contiguity constraints have been introduced. We mention the recent work of Mischek and Musliu (2021), who introduced the test laboratory scheduling problem, an extension of the resource-constrained project scheduling problem inspired by a real-world problem similar to ours. They introduced the notion of "link" between two jobs to indicate that they need to be performed by the same employee. In contrast, we originally consider the interactions between contiguity constraints and workforce. Khatami et al. (2020) surveyed coupled task scheduling problems, where it is required to have an exact time interval between the two jobs of a pair and this time can be used to process other jobs (see also Khatami and Salehipour, 2021a and Khatami and Salehipour, 2021b for other recent works). While the definition of contiguity in our problem has a few similarities with the aforementioned constraints, it cannot exactly be considered just a combination of existing concepts, because we formalize the notion of contiguity constraints between pairs of jobs, which enriches the RCPMSP literature. We mention that a different definition of contiguity in which one requires a pair of jobs to be scheduled in consecutive machines exists in the cutting and packing literature. To the best of our knowledge, that definition was only used to introduce $P|cont|C_{max}$, a relaxation of the well-known strip packing problem (see Côté et al., 2014).

We categorize in Table 4.1 the relevant PMSP literature discussed in this work and identify, for each paper, the types of constraints considered (among those studied in our problem) and the type of approaches used: heuristics and metaheuristics (HM), MILP models, CP models, or decomposition approaches (DA).

Table 4.1: Categorization of the recent related PMSP literature.

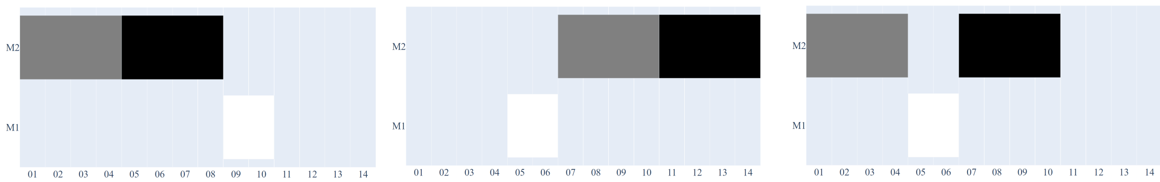
Reference	Constraints						Approach			
	*res*1	$\mathcal{M}_j$	r_j	d_j	*prec*	*cont*	HM	MILP	CP	DA
Afzalirad and Rezaeian (2016)	✓	✓	✓		✓		✓	✓		
Chen (2004)	✓			✓						✓
Chudak and Shmoys (1999)					✓		✓			
Daniels et al. (1997)	✓						✓	✓		
Edis and Ozkarahan (2011)	✓	✓						✓	✓	
Edis and Ozkarahan (2012)	✓	✓						✓	✓	✓
Fang et al. (2021)	✓						✓	✓		✓
Fleszar and Hindi (2018)	✓							✓	✓	✓
Gökgür et al. (2018)	✓								✓	
Hooker (2006)*		✓	✓	✓	✓			✓	✓	✓
Hu et al. (2010)		✓			✓		✓			
A. Kramer et al. (2019)				✓				✓		
Li et al. (2022)			✓				✓	✓		
Mischek and Musliu (2021)	✓	✓	✓	✓	✓		✓		✓	
Su et al. (2017)		✓		✓			✓			
Ventura and Kim (2000)	✓			✓				✓		
Caselli et al. (2022)	✓	✓	✓	✓	✓	✓		✓		
This paper	✓	✓	✓	✓	✓	✓		✓	✓	✓

4.3 Problem statement and descriptive model

In our RCPMSP, we are given a set J of jobs that must be scheduled on a set M of machines by a set K of workers in the time horizon $\mathcal{T} = 0, \dots, t_{max}$ expressed in days. Every job $j \in J$ has a release date r_j, a due date d_j, a daily resource consumption q_j (number of working hours), a processing time p_j, and a weight w_j. The weighted tardiness of each job is computed by multiplying its weight by its delay with respect to the due date, if any. The aim of our RCPMSP is to minimize the total weighted tardiness (z). To this aim, each job j must be processed on a machine i that belongs to the subset $M_j \subseteq M$ of machines compatible with job j, and by a worker k that belongs to the subset $K_{ij} \subseteq K$ of workers compatible with both job j and machine i. We also define $K_j \subseteq K$ as the set of workers compatible with job j. We assume that every machine can be used every day of the time horizon, while every worker k is available for u_{kt} hours in day t ($k \in K, t \in \mathcal{T}$).

We are given a set P of precedence relations and a set Q of contiguity relations. A precedence relation $(j, l) \in P$ between job j and job l implies that the starting time of l must be greater than or equal to the completion time of j. A contiguity relation $(j, l) \in Q$ between job j and job l states that the starting time of l must be greater than or equal to the completion time of j and that the two jobs must be processed on the same machine without any other jobs in-between (but idle time is allowed).

Interestingly, the notion of contiguity is only relevant in our problem because of the workers. One could think that the tasks involved in a contiguity constraint could simply be merged so that they are forced to be processed one after the other on the same machine. However, doing so would remove the possibility to slightly postpone the second task to free its associated worker so that they become available to perform another (more urgent) job. A numerical example illustrating this situation is given in Figure 4.1. We consider three jobs $J = \{1, 2, 3\}$ (in white, gray, and black in the figure) with release dates $r_j = \{4, 0, 4\}$, due dates $d_j = \{8, 5, 9\}$, processing times $p_j = \{2, 4, 4\}$, and weights $w_j = \{1, 1, 1\}$. There is a contiguity relation between jobs 2 and 3 ($Q = \{(2, 3)\}$). We consider two machines $M = \{M1, M2\}$ with $M1$ compatible with job 1 and $M2$ compatible with jobs 2 and 3, and a single worker compatible with every job and every machine who can perform one job at a time. We show in Figures 4.1a and 4.1b the two possible outcomes if contiguous jobs 2 and 3 are merged. In that case, the total weighted tardiness (z) is either equal to 2 if the merged jobs are scheduled before job 1 or it is equal to 10 if the merged jobs are scheduled after job 1. If a contiguity relation is considered between the two jobs instead, then a better solution with objective value 1 can be found as displayed in Figure 4.1c. We point out



(a) $z = 2$ (b) $z = 10$ (c) $z = 1$

Figure 4.1: Numerical example on contiguity relation.

that if there is a contiguity relation (j, l) between two jobs j and l, then the sets of compatible machines M_j and M_l should be identical. If this is not the case, then a simple preprocessing step sets M_j and M_l to $M_j \cap M_l$ for every $(j, l) \in Q$.

For each job j and day t, we also define the set of starting times S_{jt} for which a machine would still be occupied at day t if job j were to start at any day τ that belongs to S_{jt}. Set S_{jt} is equal to $\{\emptyset\}$ if $r_j > t$, to $\{r_j, \dots, t\}$ if $t \geq r_j \geq t - p_j + 1$, and to $\{t - p_j + 1, \dots, t\}$ otherwise. The mathematical notation is summarized in Table 4.2.

Our RCPMSP can be modeled using a so-called discrete-time formulation (see Pritsker et al., 1969) where binary decision variable x_{ijkt} takes the value 1 if job j is processed on machine i by worker k on day t ($j \in J, i \in M_j, k \in K_{ij}, t \in \mathcal{T}$) and where continuous decision variables C_j and T_j indicate the completion time of job j and its tardiness (i.e., the maximum between 0 and $C_j - d_j$), respectively. We obtain the following MILP model:

$$\min \sum_{j \in J} w_j T_j \tag{4.1}$$

$$\text{s.t.} \sum_{i \in M_j} \sum_{k \in K_{ij}} \sum_{t \in \mathcal{T}} x_{ijkt} = 1 \qquad j \in J \tag{4.2}$$

$$\sum_{j \in J} \sum_{k \in K_{ij}} \sum_{\tau \in S_{jt}} x_{ijk\tau} \leq 1 \qquad i \in M, t \in \mathcal{T} \tag{4.3}$$

Table 4.2: Mathematical notation.

Notation	Definition
J	Set of jobs
M	Set of available machines
M_j	Set of machines compatible with job j
K	Set of available workers
K_j	Set of workers compatible with job j
K_{ij}	Set of workers compatible with job j and machine i
$\mathcal{T}$	Set of days in the time horizon ($\mathcal{T} = 0, \dots, t_{\max}$)
r_j	Release date of job j
d_j	Due date of job j
q_j	Daily resource consumption of job j (hours per day)
p_j	Processing time of job j (days)
w_j	Weight of job j
u_{kt}	Working availability of worker k in day t (hours)
P	Set of precedence relations
Q	Set of contiguity relations
S_{jt}	Set of starting times for which job j would not be finished by day t

$$\sum_{j \in J} \sum_{i \in M_j : k \in K_{ij}} \sum_{\tau \in S_{jt}} q_j x_{ijk\tau} \leq u_{kt} \qquad k \in K, t \in \mathcal{T} \quad (4.4)$$

$$C_j = \sum_{i \in M_j} \sum_{k \in K_{ij}} \sum_{t \in \mathcal{T}} x_{ijkt}(t + p_j) \qquad j \in J \quad (4.5)$$

$$C_l - p_l \geq C_j \qquad (j, l) \in P \cup Q \quad (4.6)$$

$$\sum_{k \in K_{ij}} \sum_{t \in \mathcal{T}} x_{ijkt} = \sum_{k \in K_{il}} \sum_{t \in \mathcal{T}} x_{ilkt} \qquad (j, l) \in Q, i \in M_j \quad (4.7)$$

$$\sum_{j' \in J \setminus \{j,l\}} \sum_{k \in K_{ij'}} x_{ij'kt} \leq 1 - \sum_{k \in K_{ij}} \sum_{\tau = r_j}^{t} x_{ijk\tau} + \sum_{k \in K_{il}} \sum_{\tau = r_l}^{t} x_{ilk\tau} \qquad (j, l) \in Q, i \in M_j, t \in \mathcal{T} \quad (4.8)$$

$$T_j \geq C_j - d_j \qquad j \in J \quad (4.9)$$

$$x_{ijkt} = 0 \qquad j \in J, i \in M_j, k \in K_{ij}, t \in \mathcal{T} \setminus \{r_j, \dots, t_{max} - p_j\} \quad (4.10)$$

$$x_{ijkt} \in \{0, 1\} \qquad j \in J, i \in M_j, k \in K_{ij}, t \in \mathcal{T} \quad (4.11)$$

$$C_j, T_j \geq 0 \qquad j \in J \quad (4.12)$$

The objective function (4.1) minimizes the total weighted tardiness. Constraints (4.2) guarantee that each job is processed exactly once and is assigned to exactly one compatible machine and one compatible worker. Constraints (4.3) ensure that each machine processes at most one job at the same time (i.e., one job per day). Constraints (4.4) guarantee that no operator works more than the number of hours they are supposed to work in each day of the time horizon. Constraints (4.5) define the completion time of each job. Constraints (4.6) guarantee that the order between jobs defined by precedence and contiguity relations is respected. Constraints (4.7) make sure that two contiguous jobs are processed on the same machine. Constraints (4.8) forbid any job j' to be processed on a machine i at day t (i.e., $\sum_{j' \in J \setminus \{j,l\}} \sum_{k \in K_{ij'}} x_{ij'kt} = 1$) if that machine has processed a job j, the first element of a contiguous pair, before day t (i.e., $\sum_{k \in K_{ij}} \sum_{\tau=r_j}^{t} x_{ijk\tau} = 1$) but has not yet processed job l, the second element of that same contiguous pair, by day t (i.e., $\sum_{k \in K_{il}} \sum_{\tau=r_l}^{t} x_{ilk\tau} = 0$). Constraints (4.9) define the tardiness of the jobs, constraints (4.10) prevent any job from starting before its release date or finishing after the time horizon (note that in the model implementation the variables set to zero are not created), and constraints (4.11) and (4.12) limit the variable domains. We point out that the innovative aspect of this formulation comes from constraints (4.7) and (4.8), which are used to model contiguity. The other constraints are fairly common in the scheduling literature (see, e.g., Koné et al., 2011). Model (4.1)-(4.12) can be solved with an MILP solver to produce an optimal solution, but computational experiments have shown that the model could quickly become too large to be solved in a reasonable time (it involves $O(|J| \cdot |M| \cdot |K| \cdot t_{max})$ variables).

4.4 Advanced algorithms

In this section, we introduce an enhanced MILP model, a CP model, and a combinatorial Benders' decomposition for our RCPMSP.

4.4.1 Enhanced MILP model

There are often more than one MILP formulation to model a given combinatorial optimization problem. The literature has shown that, for a given instance of a problem and a given MILP solver, some MILP formulations could be faster to solve than others, all else being equal. This difference can be explained by major differences in terms of number of variables, number of constraints, number of non-zero elements, and continuous relaxation value (among others). For example, we refer the reader to Koné et al. (2011) for a comparison of MILP formulations for the resource-constrained project scheduling problem. A question frequently arising in MILP modeling is whether or not one should use a set of α-index variables or two sets of (α-1)-index variables together with a set of linking constraints. While the former is usually shown to have a better continuous relaxation value, the latter tends to involve much fewer variables. As preliminary results showed that the model size was an issue with model (4.1)-(4.12), we tried a version of the model using two sets of 3-index variables instead of one set of 4-index variables: we now define x^m_{ijt} that takes the value 1 if job j is processed on machine i on day t ($j \in J, i \in M_j, t \in \mathcal{T}$) and x^w_{jkt} that takes the value 1 if job j is processed by worker k on day t ($j \in J, k \in K_j, t \in \mathcal{T}$). The resulting MILP model, which uses $O(|J| \cdot |M + K| \cdot t_{max})$ variables, is defined as follows:

$$\min \quad \sum_{j \in J} w_j T_j \tag{4.13}$$

$$\text{s.t.} \quad \sum_{i \in M_j} \sum_{t \in \mathcal{T}} x^m_{ijt} = 1 \qquad j \in J \tag{4.14}$$

$$\sum_{k \in K_j} \sum_{t \in \mathcal{T}} x^w_{jkt} = 1 \qquad j \in J \tag{4.15}$$

$$\sum_{j \in J} \sum_{\tau \in S_{jt}} x^m_{ij\tau} \leq 1 \qquad i \in M, t \in \mathcal{T} \tag{4.16}$$

$$\sum_{j\in J}\sum_{\tau\in S_{jt}} q_j x^w_{jk\tau} \le u_{kt} \qquad k\in K, t\in\mathcal{T} \tag{4.17}$$

$$C_j = \sum_{i\in M_j}\sum_{t\in\mathcal{T}} x^m_{ijt}(t+p_j) \qquad j\in J \tag{4.18}$$

$$C_l - p_l \ge C_j \qquad (j,l)\in P\cup Q \tag{4.19}$$

$$\sum_{t\in\mathcal{T}} x^m_{ijt} = \sum_{t\in\mathcal{T}} x^m_{ilt} \qquad i\in M, (j,l)\in Q \tag{4.20}$$

$$\sum_{j'\in J\setminus\{j,l\}} x^m_{ij't} \le 1 - \sum_{\tau=r_j}^{t} x^m_{ij\tau} + \sum_{\tau=r_l}^{t} x^m_{il\tau} \quad i\in M, t\in\mathcal{T}, (j,l)\in Q \tag{4.21}$$

$$T_j \ge C_j - d_j \qquad j\in J \tag{4.22}$$

$$\sum_{t\in\mathcal{T}} x^m_{ijt} \le \sum_{t\in\mathcal{T}}\sum_{k\in K_{ij}} x^w_{jkt} \qquad j\in J, i\in M_j, \tag{4.23}$$

$$\sum_{i\in M_j} x^m_{ijt} = \sum_{k\in K_j} x^w_{jkt} \qquad j\in J, t\in\mathcal{T} \tag{4.24}$$

$$x^m_{ijt} = 0 \qquad j\in J, i\in M_j, t\in\mathcal{T}\setminus\{r_j,\dots,t_{max}-p_j\} \tag{4.25}$$

$$x^w_{jkt} = 0 \qquad j\in J, k\in K_j, t\in\mathcal{T}\setminus\{r_j,\dots,t_{max}-p_j\} \tag{4.26}$$

$$x^m_{ijt}\in\{0,1\} \qquad j\in J, i\in M_j, t\in\mathcal{T} \tag{4.27}$$

$$x^w_{jkt}\in\{0,1\} \qquad j\in J, k\in K_j, t\in\mathcal{T} \tag{4.28}$$

$$C_j, T_j \ge 0 \qquad j\in J \tag{4.29}$$

Model (4.13)-(4.29) is an adaptation of model (4.1)-(4.12) where the additional constraints (4.23) allow a job j to be assigned to a compatible machine i only if j is assigned to a worker k that is compatible with both j and i, and the additional constraints (4.24) link the two sets of 3-index variables.

4.4.2 Constraint Programming model

In the last decade, the literature has shown that CP is particularly effective to solve highly constrained combinatorial optimization problems (see, e.g., Jain and Grossmann, 2001 and Gedik et al., 2016). While CP is sometimes used as a stand-alone approach, it can also be integrated into more sophisticated algorithms, either to quickly find a feasible solution to serve as a warm start for an MILP model (see, e.g., Delorme and Santini, 2022) or to solve the subproblem of a decomposition approach (see, e.g., Delorme et al., 2017). We refer the reader to Bockmayr and Hooker (2005) for an outline of CP basic concepts and its integration with MILP.

Several works using CP models to solve scheduling problems have been proposed in the PMSP literature (see Hooker, 2006, Gedik et al., 2016, Fleszar and Hindi, 2018, Gökgür et al., 2018, and Lovato et al., 2022). In the following, we describe our CP formulation together with a detailed explanation of every necessary function. To take the worker availability into account, we consider that every worker is initially available at full capacity every day and we use a set of worker-dependent dummy jobs with pre-defined resource consumption and starting/ending dates to represent the possible reduction of the full capacity. To model the eligibility constraints, we use the multi-mode resource-constrained project scheduling problem transformation mentioned in Section 4.2, which is to consider $|M| + |K|$ resources and create, for each job j, one mode for every single compatible machine/worker pair (i, k) $(i \in M_j, k \in K_{ij})$ that occupies machine i and consumes resource k for p_j days. To represent the contiguity relations $(j, l) \in Q$, we use a dummy job with one mode per compatible machine i $(i \in M_j)$ that occupies machine i and no worker resource for an unrestricted duration. This dummy job must start exactly when job j finishes and must stop exactly when job l starts.

We make use of the following sets of interval variables to model the problem:

- ivl_j, interval variable that represents the execution of job j;
- ivl_m_{ijk}, interval variable that represents the execution of job j using the mode that occupies machine i and worker k;
- $ivld_{jl}$, interval variable that represents the execution of the dummy job associated with the contiguity relation between jobs j and l;
- $ivld\_m_{ijl}$, interval variable that represents the execution of the dummy job associated with the contiguity between jobs j and l using the mode that consumes machine i.

Every interval variable must be processed in exactly one of its modes, and if a constraint applies to a specific interval variable, then this constraint is propagated to each of the variable modes. For example, if a constraint states that ivl_j cannot start before day r_j for a given job j, then none of the ivl_m_{ijk} involving job j can start before day r_j.

In our CP formulation, we minimize the total weighted tardiness:

$$\min \sum_{j \in J} w_j * \texttt{max}(0, \texttt{endof}(ivl_j) - d_j) \tag{4.30}$$

where $\texttt{endof}(ivl_j)$ indicates the completion time of job j. Constraints (4.31)-(4.32) limit the domain of the variables so that the jobs are processed between their release date and t_{max}:

$$ivl_j.\texttt{setStartMin}(r_j) \quad j \in J \tag{4.31}$$

$$ivl_j.\texttt{setStartMax}(t_{max} - p_j) \quad j \in J \tag{4.32}$$

The following constraints fix the duration of each job j to be equal to p_j:

$$ivl_j.\texttt{setSizeMin}(p_j) \quad j \in J \tag{4.33}$$

$$ivl_j.\texttt{setSizeMax}(p_j) \quad j \in J \tag{4.34}$$

We use the constraints `alternative` to ensure that exactly one mode ivl_m_{ijk} $(i \in M_j, k \in K_{ij})$ per job j $(j \in J)$ is present in the solution:

$$\texttt{alternative}(ivl_j, ivl\_m_{ijk} : \forall i \in M_j, k \in K_{ij}) \quad j \in J \tag{4.35}$$

To guarantee that each machine processes at most one job per day (whether the job is a real one requiring a worker or a dummy one representing an idle time on the machine), we impose

$$\sum_{j \in J} \sum_{k \in K_{ij}} \texttt{pulse}(ivl\_m_{ijk}, 1) + \sum_{(j,l) \in Q : i \in M_j} \texttt{pulse}(ivld\_m_{ijl}, 1) \leq 1 \quad i \in M \tag{4.36}$$

where $\texttt{pulse}(ivl\_m_{ijk}, 1)$ counts one unit of occupation for machine i from the starting time of ivl_m_{ijk} to its completion time, provided that mode ivl_m_{ijk} is used. Similarly, to guarantee that every worker is only active for the number of hours they are supposed to work, we add

$$\sum_{j \in J} \sum_{i \in M_j : k \in K_{ij}} \texttt{pulse}(ivl\_m_{ijk}, q_j) + \sum_{h \in H_k} \texttt{pulse}(ivl_h, q_h) \leq \max_{t \in \mathcal{T}} \{u_{kt}\} \quad k \in K \quad (4.37)$$

where $\max_{t \in \mathcal{T}} \{u_{kt}\}$ is the number of hours per day worker k is available when they work at full capacity, and where H_k contains the dummy jobs with fixed starting/ending dates and the resource consumption associated with worker k that is used to model a decrease in the worker availability (e.g., for holidays). Precedence relations (including those coming from contiguity) are modeled as

$$\texttt{endBeforeStart}(ivl_j, ivl_l) \quad (j, l) \in P \cup Q \quad (4.38)$$

where $\texttt{endBeforeStart}(ivl_j, ivl_l)$ indicates that ivl_j must end before ivl_l can start. The supplementary constraints related to contiguity relations are enforced by

$$\sum_{k \in K_{ij}} \texttt{presenceOf}(ivl\_m_{ijk}) = \sum_{k \in K_{il}} \texttt{presenceOf}(ivl\_m_{ilk}) \; (j, l) \in Q_j, i \in M_j \quad (4.39)$$

$$\sum_{k \in K_{ij}} \texttt{presenceOf}(ivl\_m_{ijk}) = \texttt{presenceOf}(ivld\_m_{ijl}) \quad (j, l) \in Q_j, i \in M_j \quad (4.40)$$

$$\texttt{StartAtEnd}(ivld_{jl}, ivl_j) \quad (j, l) \in Q \quad (4.41)$$

$$\texttt{EndAtStart}(ivld_{jl}, ivl_l) \quad (j, l) \in Q \quad (4.42)$$

$$\texttt{alternative}(ivld_{jl}, ivld\_m_{ijl} : \forall i \in M_j) \quad (j, l) \in Q \quad (4.43)$$

Constraints (4.39) and (4.40) use job modes to make sure that two contiguous jobs j and l and their associated dummy job $ivld_{jl}$ are processed on the same machine. To do so, constraint $\texttt{presenceOf}(ivl\_m_{ijk})$ returns true if ivl_m_{ijk} is present in the solution. Constraints (4.41) and (4.42) force the dummy job $ivld_{jl}$ to start at the end of job j and finish at the start of job l. To do so, constraints $\texttt{StartAtEnd}(ivld_{jl}, ivl_j)$ and $\texttt{EndAtStart}(ivld_{jl}, ivl_l)$ impose an exact starting and ending time for $ivld_{jl}$. Finally, constraints (4.43) ensure that exactly one mode among those associated with contiguity relation (j, l) is selected.

4.4.3 Combinatorial Benders' decomposition

Benders' decomposition, which was introduced more than sixty years ago by Benders (1962), splits a large MILP problem (called *original problem* afterwards) into two problems, called *master problem* (MP) and *subproblem* (SP). The main idea behind the decomposition is to solve an MP of reasonable size (typically, a relaxation of the original problem where a set of variables/constraints is aggregated or removed) to obtain a solution ζ and check in the SP if ζ is also feasible for the original problem. If that is the case, then ζ is also optimal for the original problem, otherwise a cut is added to the MP, so that ζ cannot be generated once more, and the MP is solved again.

Benders' decomposition has significantly evolved during the last decades. In the seminal work of Benders (1962), the MP was an MILP and the SP was an LP. A few years later, Geoffrion (1972) generalized the decomposition to the case where the SP too was an MILP. Later on, Hooker and Ottosson (2003) proposed the logic-based Benders' decomposition, where the SP was modeled with CP. A major advance in the area can be attributed to Codato and Fischetti (2006), who introduced the concept of combinatorial Benders' cuts by searching for the smallest subset of variables in the MP solution that causes infeasibility. Even though such cuts can be costly to find, they were shown to be more effective in practice because they strongly reduce the number of MP/SP iterations since they cut larger portions of the solution space. Another advance was proposed by Côté et al. (2014) who used a lifting procedure to increase even further the solution space removed by the cuts. Recent successful applications can be found in Dell'Amico et al. (2019), Fang et al. (2021), Karlsson and Rönnberg (2022), and Seo et al. (2022), among others.

In our decomposition, the MP determines the starting day of every job while taking into account the precedence constraints and an aggregated form of resource constraints (derived from the machines and the workers). The SP tries to assign every job to a suitable machine and a suitable worker while also ensuring the contiguity relations among jobs. In such a decomposition framework, only feasibility cuts are added to the MP. Indeed, optimality cuts are never needed because the objective value of a given MP solution is always equal to the objective value of the corresponding solution for the original problem (if such a corresponding solution was shown to exist by the SP). Preliminary experiments showed that, most of the time, only a few MP/SP iterations were needed to reach an optimal solution by using this strategy. An alternative idea would be to take the decisions in reverse order and assign workers and machines to jobs in the MP and then determine the best schedule in the SP. Such a decomposition framework would require feasibility cuts as the objective value of a given MP solution for the original problem would only be known after solving the SP. Considering that many instances have a non-zero optimal value and that it is difficult to get any non-trivial bound on the total weighted tardiness in the MP without knowing the starting dates of the jobs, it is expected that using such a strategy would result in significantly more MP/SP iterations. We thus opted not to investigate this alternative idea. In the following, we describe each component of the decomposition method in more detail.

Master Problem In the MP, we only consider two resources: one is an aggregation of the $|M|$ machines and the other one is an aggregation of the $|K|$ workers. Therefore, the MP does not take into account the eligibility constraints and only considers a relaxed version of the contiguity ones. The MP uses a new set of binary variables y defined as

$$y_{jt} = \sum_{i \in M_j} \sum_{k \in K_{ij}} x_{ijkt}$$

where y_{jt} takes the value 1 if job j starts at day t, and 0 otherwise ($j \in J, t \in \mathcal{T}$). We re-use the two sets of continuous variables C_j and T_j to represent the completion time and tardiness of job j, respectively. The MP is defined as follows:

$$\min \quad \sum_{j \in J} w_j T_j \tag{4.44}$$

$$\text{s.t.} \quad \sum_{t \in \mathcal{T}} y_{jt} = 1 \qquad j \in J \tag{4.45}$$

$$\sum_{j' \in J} \sum_{\tau \in S_{j't}} y_{j'\tau} \leq |M| - \sum_{(j,l) \in Q} \sum_{\tau = r_j}^{t-p_j} y_{j\tau} + \sum_{(j,l) \in Q} \sum_{\tau = r_l}^{t} y_{l\tau} \qquad t \in \mathcal{T} \qquad (4.46)$$

$$\sum_{j \in J} \sum_{\tau \in S_{jt}} q_j y_{j\tau} \leq \sum_{k \in K} u_{kt} \qquad t \in \mathcal{T} \qquad (4.47)$$

$$C_j = \sum_{t \in \mathcal{T}} y_{jt}(t + p_j) \qquad j \in J \qquad (4.48)$$

$$C_l - p_l \geq C_j \qquad (j,l) \in P \cup Q \qquad (4.49)$$

$$T_j \geq C_j - d_j \qquad j \in J \qquad (4.50)$$

$$\sum_{(j,t) \in \zeta} y_{jt} \leq |\zeta| - 1 \qquad \zeta \in Z \qquad (4.51)$$

$$y_{jt} = 0 \qquad j \in J, t \in \mathcal{T} \setminus \{r_j, \ldots, t_{max} - p_j\} \qquad (4.52)$$

$$y_{jt} \in \{0,1\} \qquad j \in J, t \in \mathcal{T} \qquad (4.53)$$

$$C_j, T_j \geq 0 \qquad j \in J \qquad (4.54)$$

which is an adaptation of model (4.1)-(4.12) using the two-index variables y_{jt}. The only differences are for constraints (4.51), which forbid the solutions that were previously shown to be infeasible by the SP, and for constraints (4.46), which merge constraints (4.3) and (4.8). More specifically, constraints (4.46) limit the number of jobs running in parallel to the total number $|M|$ of machines minus the number of machines that cannot be used because of the contiguity relations. For every day t, the latter quantity can be computed as the number of contiguity pairs for which the first member has been completed by day t (i.e., $\sum_{(j,l) \in Q} \sum_{\tau = r_j}^{t-p_j} y_{j\tau}$) minus the number of contiguity pairs for which the second member has not started by day t (i.e., $\sum_{(j,l) \in Q} \sum_{\tau = r_l}^{t} y_{l\tau}$).

Subproblem Given an MP solution $\{\widetilde{y_{jt}}, \widetilde{C_j}\}$ indicating the starting time of every job, the SP determines whether or not there is a feasible job/machine/worker allocation respecting both the eligibility and contiguity constraints. The SP uses a set of binary decision variables ξ_{ijk} that take the value 1 if job j is processed on machine i by worker k, and 0 otherwise ($j \in J, i \in M_j, k \in K_{ij}$). Let $J^Q_{jl} = \{j' \in J \setminus \{j,l\} : \widetilde{C_j} - p_j < \widetilde{C_{j'}},\ \widetilde{C_{j'}} - p_{j'} < \widetilde{C_l}\}$. The SP becomes

$$\min \quad 0 \qquad (4.55)$$

$$\text{s.t.} \quad \sum_{i \in M_j} \sum_{k \in K_{ij}} \xi_{ijk} = 1 \qquad j \in J \qquad (4.56)$$

$$\sum_{\substack{j \in J \\ \widetilde{C_j} - p_j \in S_{jt}}} \sum_{k \in K_{ij}} \xi_{ijk} \leq 1 \qquad i \in M, t \in \mathcal{T} \qquad (4.57)$$

$$\sum_{\substack{j \in J \\ \widetilde{C_j} - p_j \in S_{jt}}} \sum_{i \in M_j : k \in K_{ij}} q_j \xi_{ijk} \leq u_{kt} \qquad k \in K, t \in \mathcal{T} \qquad (4.58)$$

$$\sum_{k \in K_{ij}} \xi_{ijk} = \sum_{k \in K_{il}} \xi_{ilk} \qquad (j,l) \in Q, i \in M_j \qquad (4.59)$$

$$\sum_{j' \in J^Q_{jl}} \sum_{k \in K_{ij'}} \xi_{ij'k} + \sum_{k \in K_{ij}} \xi_{ijk} \leq 1 \qquad (j,l) \in Q, i \in M_j \tag{4.60}$$

$$\xi_{ijk} \in \{0,1\} \qquad j \in J, i \in M_j, k \in K_{ij} \tag{4.61}$$

Model (4.55)-(4.61) is an adaptation of model (4.1)-(4.12) where the starting date of each job is fixed. We point out the necessary adaptations in resource constraints (4.57) and (4.58), where, for each day t, the ξ_{ijk} sum is taken only over the jobs j that are active at day t (i.e., those whose starting time $\widetilde{C_j} - p_j$ is in set S_{jt}). As far as constraints (4.60) are concerned, they simply indicate that a job j' cannot be processed on the same machine as a pair of contiguous jobs $(j,l) \in Q$ with which it is in conflict. A job j' is in conflict with a pair of contiguous jobs (j,l) if the completion time of job j' is greater than the starting time of job j (i.e., $\widetilde{C_{j'}} > \widetilde{C_j} - p_j$) and if the starting time of job j' is smaller than the completion time of job l (i.e., $\widetilde{C_{j'}} - p_{j'} < \widetilde{C_l}$). We also tried a version of the SP that uses two sets of two-index binary variables instead of one set of three-index variables, but we did not observe significant improvements in terms of average computation time.

Cut generation and overall framework Given an MP solution $\{\widetilde{y_{jt}}, \widetilde{C_j}\}$ that was proven to be infeasible by the SP, a valid cut to prevent $\{\widetilde{y_{jt}}, \widetilde{C_j}\}$ to be generated again is

$$\sum_{(j,t) \in \zeta} y_{jt} \leq |J| - 1 \tag{4.62}$$

where ζ contains all the variables y_{jt} ($j \in J, t \in \mathcal{T}$) for which $\widetilde{y_{jt}}$ is equal to 1 in the solution. Such a constraint is known as a "no-good cut" in the literature (Hooker, 2011) and does not forbid any integer solution besides $\{\widetilde{y_{jt}}, \widetilde{C_j}\}$. In other words, it only forbids the integer solution in which every job j starts exactly at day $\widetilde{C_j} - p_j$.

Stronger cuts can be obtained by using the combinatorial Benders' cuts of Codato and Fischetti (2006). To do so, one needs to find a reduced subset of jobs J' for which the SP remains infeasible. In other words, one now aims at forbidding all the integer solutions in which every job j in J' starts exactly at day $\widetilde{C_j} - p_j$, regardless of the starting day of the jobs in $J \setminus J'$. This enhanced cut therefore forbids $O\Big(\prod_{j \in J \setminus J'} (t_{max} - r_j)\Big)$ solutions. As more MP solutions are removed when $|J'|$ decreases, we are interested in finding the smallest subset of jobs J' causing infeasibility for the SP, also known as the Minimum Infeasible Subset (MIS) in the literature. Since finding a MIS is usually $\mathcal{NP}$-hard, a common strategy is to determine it heuristically by removing the set of variables that is the least likely to cause infeasibility, solve the SP again, and iterate until the SP becomes feasible. In our case, this difficulty is confirmed by the $\mathcal{NP}$-completeness of the SP. We thus adopt the same strategy.

In detail, the variables that are less likely to cause infeasibility in model (4.55)-(4.61) are the ξ_{ijk}'s associated with the jobs that (i) are short, (ii) are involved in few precedence/contiguity constraints, (iii) are compatible with most machines and workers, and (iv) have their completion interval $[\widetilde{C_j} - p_j; \widetilde{C_j}]$ intersecting with few other jobs. Preliminary experiments showed that criterion (i) was the most relevant in our instances. Therefore, we determine the minimum subset of jobs J' that cause infeasibility in the

SP by initializing J' to J and removing the job with the smallest duration from J' until the SP becomes feasible. The last subset J' for which the SP is infeasible is then used to generate the combinatorial Benders' cut:

$$\sum_{(j,t)\in\zeta'} y_{jt} \leq |\zeta'| - 1. \tag{4.63}$$

This is the modified version of (4.51) where ζ' contains all the variables y_{jt} ($j \in J', t \in \mathcal{T}$) for which $\widetilde{y_{jt}}$ is equal to 1 in the MP solution. A flowchart summarizing the main idea behind the combinatorial Benders' decomposition is shown in Figure 4.2.

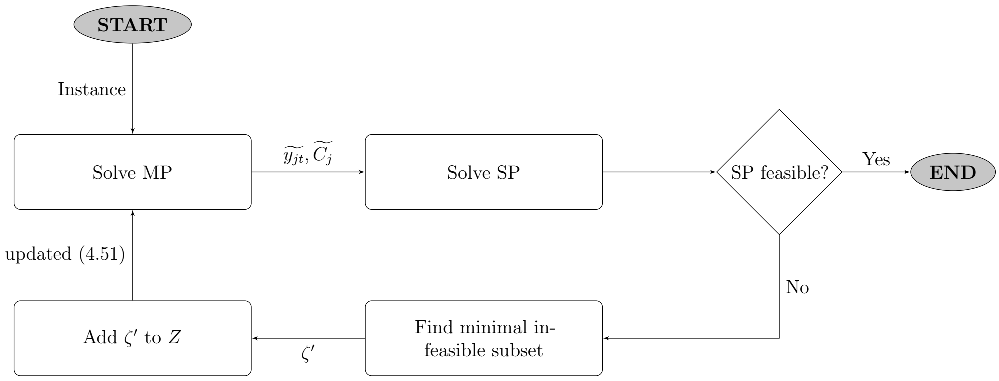


Figure 4.2: Flowchart of the combinatorial Benders' decomposition approach.

In our implementation, we embedded the SP solution in a callback function provided by the MILP solver we adopted. In this way, we only need to solve MP once. We also tried the lifting procedure proposed by Côté et al. (2014). The idea behind such a cut lifting is to forbid integer solutions in which every job $j \in J$ starts within a job-specific range. The resulting cut is a generalization of the combinatorial Benders' cut that forbids integer solutions in which every job $j \in J'$ starts in the (job-specific) range $[\widetilde{C_j} - p_j, \widetilde{C_j} - p_j]$ and every job $j \in J \setminus J'$ starts in the (job-specific) range $[r_j, t_{max} - p_j]$. We adapted the lifting procedure described in Côté et al. (2014) to our problem but did not observe significant improvements in terms of average computation time.

4.5 Computational experiments

In this section, we study the empirical performance of each of the proposed algorithms on three sets of instances. All algorithms were coded in `C++`. The MILP models were solved with `Gurobi 9.5.1` while the CP model was solved with `IBM ILOG CPLEX CP Optimizer 22.1.0` (as Gurobi does not offer a CP solver). The tests were executed on a single thread of a virtual machine Intel(R) Xeon(R) Gold with 2.30 GHz and 20 GB of RAM memory, running under Windows 10 Pro N. We fed each approach with a warm start obtained by running the CP model until a feasible solution was found.

We first investigate the impact of key instance features such as the presence of precedence/contiguity constraints, the worker/machine/job ratios, and the probability for worker/ machine/job to be compatible. We then describe a case study originating from Dana Inc. and outline the potential gain that could be reached using optimization models instead of handmade solutions. We conclude by studying the scaling potential of our algorithms using large size instances designed to mimic the case study. All instances can be downloaded from the online repository https://github.com/regor-unimore/Parallel-Machine-Scheduling-with-Contiguity.

4.5.1 Experiment on random instances

To test the impact of various instance features, we generated a set of random instances where some parameters are randomly created while others are taken from the case study. The features that were investigated are:

1. number of jobs, machines, and workers: we tested 9 combinations $(|J|, |M|, |K|) \in \{(50, 2, 2), (50, 5, 3), (50, 5, 5), (100, 4, 4), (100, 10, 5), (100, 10, 10), (200, 8, 8), (200, 20, 10), (200, 20, 20)\}$;

2. number of precedence and contiguity relations: we tested two configurations, one without any precedence/contiguity constraints at all, and another one with a number of precedence/contiguity constraints randomly distributed in the range $[0, 0.05|J| + 1]$ for the precedence constraints and $[0, 0.35|J| + 1]$ for the contiguity constraints. Once the numbers of constraints were determined, we randomly selected pairs of successive jobs $(j, j + 1)$ and added them to either P or Q. The resulting chain structure for the precedence/contiguity relations imitates the project structure of the case study. Instances without precedence and contiguity relations were tested in order to serve as a "control group" and determine whether an approach is better than another because it is more effective at handling precedence and contiguity constraints or because it is more effective overall;

3. job/machine, job/worker, and machine/worker eligibility ratios: we tested five ratios $r \in \{1, 0.9, 0.8, 0.7, 0.6\}$ and randomly selected $r \times |M|$ compatible machines and $r \times |K|$ compatible workers for every job. We also randomly selected $r \times |K|$ compatible workers per machine. A post-processing step changed the eligibilities of the jobs involved in the same chain of contiguities so that every job in the chain has the same job/machine and job/worker eligibilities as the first job of the chain.

One instance was generated for each of the $9 \times 2 \times 5 = 90$ combinations. The other features are:

1. the time horizon was fixed at $t_{max} = 400$;

2. every worker is available 8 hours per day during the week and is not available during the weekend;

3. the release date r_j of a job j was randomly selected in the range $[0, 230]$. A post-processing step changed the release dates of the jobs involved in the same chain of precedence/contiguity relations so that every job in the chain has the same release date, which is set to the earliest release date r_{min} among all the jobs in the chain;

4. the daily resource consumption q_j of a job j is either equal to 1, with probability 0.2, or 8 with probability 0.8. In other words, a job needs a worker either punctually (we call such a job "passive") or for its whole duration (we call such a job "active");

5. the processing time p_j of every job j is a value randomly selected in the range $[1, 10]$ for active jobs and $[10, 50]$ for passive jobs;

6. the due date of every job d_j was set to $r_j + \xi p_j$, where ξ is an integer value randomly selected in the range $[1, 3]$. A post-processing step changed the due dates of the jobs involved in the same chain of precedence/contiguity relations so that every job in the chain has the same due date. This was set to $r_{min} + \xi \sum p_j$, where ξ is an integer value randomly selected in the range $[1, 2]$. Note that due dates above the time horizon were set to t_{max};

7. the weight of every job w_j is equal to 4 with probability 0.4, to 3 with probability 0.3, to 2 with probability 0.2, and to 1 with probability 0.1.

We report in Tables 4.3-4.6 a summary of the results obtained by our algorithms on the random instances. In each table, the first set of columns indicates the instance features, while the following columns report, for each algorithm, the number of optimal solutions found within the time limit ("# opt"), the average CPU time in seconds ("T(s)") including the instances that were not solved to optimality and the instances that were stopped because of the memory limit (for such instances, 3600 seconds of running time were considered), and the average CPU time in seconds computed only on the instances solved to optimality ("T_{opt}(s)"). In the tables and thereafter, "MILP" refers to the MILP model (4.1)-(4.12), "MILP+" refers to the enhanced MILP model (4.13)-(4.29), "CP" refers to model (4.30)-(4.43), and "Decomp" refers to the combinatorial Benders' decomposition algorithm in Section 4.4.3. We also report in columns "OBA" ("Only Best Algorithm") the results obtained by the best approach among MILP, MILP+, CP, and Decomp for every instance. In other words, OBA simulates the performance of a hyper-algorithm able to predict with 100% accuracy the algorithm that is the fastest to solve a given instance; it is thus of interest as a comparison term with respect to the performance of the previous algorithms. To identify the methods that contribute the most to the performance of OBA, we also include in the "# opt" columns (within round brackets) the number of instances that are solved to optimality by a given approach while being unsolved by the other three algorithms. If no number is included within brackets, then each of the instances in the group was solved to proven optimality by two or more algorithms.

We report in Table 4.3 the summary of results on the nine $(|J|, |M|, |K|)$ combinations. Overall, we observe that algorithms MILP+ and Decomp are both able to find an optimal solution for 71 out of 90 instances, compared with 66 for CP and 57 for MILP. The results obtained by the hypothetic OBA hyper-algorithm show that 82 instances could be solved to optimality (if we knew a priori which is the best algorithm for each instance). This indicates that the best algorithm is not always the same for all the instances: for example, five instances are solved by Decomp but not solved by MILP, MILP+, and CP; similarly, one instance is solved by MILP+, but not solved by MILP, CP, and Decomp. As far as instance combinations are concerned, MILP+

Table 4.3: Summary of results: impact of $(|J|, |M|, |K|)$ combinations.

Parameters				Results														
				MILP			MILP+			CP			Decomp			OBA		
$\|J\|$	$\|M\|$	$\|K\|$	#	# opt	T(s)	T_{opt}(s)	# opt	T(s)	T_{opt}(s)	# opt	T(s)	T_{opt}(s)	# opt	T(s)	T_{opt}(s)	# opt	T(s)	T_{opt}(s)
50	2	2	10	8	966	307	9	762	446	5	2233	867	6	1456	27	9	546	207
50	5	3	10	10	129	129	10	124	124	10	2	2	7	1089	12	10	2	2
50	5	5	10	10	102	102	10	57	57	9	360	0	10	3	3	10	1	1
100	4	4	10	4	2372	528	5(1)	2342	1083	1	3259	188	4(3)	2314	385	8	1173	566
100	10	5	10	10	774	774	9	570	234	10	2	2	9	385	28	10	2	2
100	10	10	10	8	889	211	10	145	145	10	1	1	10	4	4	10	1	1
200	8	8	10	2	3289	2044	2	3159	1395	1	3249	94	5(2)	2078	557	5	2078	557
200	20	10	10	3	2718	659	7	1239	227	10	8	8	10	25	25	10	7	7
200	20	20	10	2	2936	279	9	824	516	10	8	8	10	9	9	10	6	6
Tot			90	57	1575	402	71(1)	1025	336	66	1014	73	71(5)	818	74	82	424	114

solves the most instances for combinations (50,2,2) and (100,4,4), Decomp solves the most instances for combination (200,8,8), and CP solves the most instances, together with MILP, for combination (100,10,5). We also point out that CP and Decomp results are very heterogeneous in the sense that the algorithms either solve an instance to optimality very fast or not at all, as witnessed by the low T_{opt}(s) values for the two algorithms compared with the values of MILP and MILP+. The most difficult instances seem to be the ones with 200 jobs, 8 machines, and 8 workers.

Table 4.4: Summary of results: comparison between MILP and MILP+.

Parameters				Results													
				MILP							MILP+						
$\|J\|$	$\|M\|$	$\|K\|$	#	# opt	T(s)	T_{opt}(s)	LP	# var.	# cons.	# nzs.	# opt	T(s)	T_{opt}(s)	LP	# var.	# cons.	# nzs.
50	2	2	10	8	966	307	692.20	39,106	5444	2,157,317	9	762	446	692.20	44,528	25,594	1,380,991
50	5	3	10	10	129	129	0.00	115,790	14,985	7,844,411	10	124	1234	0.00	89,742	35,285	4,019,729
50	5	5	10	10	102	102	0.00	192,341	13,780	11,707,342	10	57	57	0.00	112,252	34,080	3,960,135
100	4	4	10	4	2372	528	78.90	265,873	16,984	15,787,486	5	2342	1083	77.50	178,565	57,484	5,862,264
100	10	5	10	10	774	774	0.00	780,006	60,450	64,969,996	9	570	234	0.00	342,722	101,550	20,064,260
100	10	10	10	8	889	211	0.00	1,346,082	22,840	43,757,243	10	145	145	0.00	423,762	63,940	10,382,416
200	8	8	10	2	3289	2044	6.98	1,750,779	9675	37,842,773	2	3159	1395	6.98	673,733	91,475	11,047,170
200	20	10	10	3	2718	659	0.00	4,648,955	12,600	97,902,775	7	1239	227	0.00	1,193,206	96,800	21,548,621
200	20	20	10	2	2936	278	0.00	6,265,306	16,600	125,042,161	9	825	516	0.00	1,455,916	100,800	26,859,121
Tot			90	57	1575	402	86.45	1,711,582	19,262	45,223,501	71	1025	336	86.31	501,603	67,445	11,680,523

Our next set of experiments, which is reported in Table 4.4, aims at comparing the performance of MILP and MILP+. In the table, column "LP" indicates the average continuous relaxation value of the model and columns "# var.", "# cons.", and "# nzs." indicate the average number of variables, constraints, and non-zero elements in the coefficient matrix of the model, respectively. To provide meaningful comparisons, the averages were computed by only using the instances for which the LP relaxation value could be obtained by both models (i.e., the instances for which neither of the two algorithms ran out of memory). We observe that, overall, MILP+ is faster on average (336s to solve an instance on average compared with 402s for MILP) and solves more instances (71 instances solved in total compared with 57 for MILP). These results can mostly be explained by the reduced size of the model (501,603 variables on average for MILP+ compared with 1,711,582 for MILP) at the expense of a negligible decrease in the quality of the continuous linear relaxation.

We report in Table 4.5 the impact of the job/machine/worker eligibility ratios ("r") and the impact of precedence/contiguity relations ("P/Q") on the performance of our methods. Following the aforementioned description, an instance in which the eligibility ratio "r" is 0.9 has (i) each of its jobs compatible with 90% of the machines, (ii) each of

Table 4.5: Summary of results: impact of eligibility percentage and precedence/contiguity relations.

Parameters			Results															
			MILP			MILP+			CP			Decomp			OBA			
r	P/Q	#	# opt	T(s)	T_{opt}(s)	# opt	T(s)	T_{opt}(s)	# opt	T(s)	T_{opt}(s)	# opt	T(s)	T_{opt}(s)	# opt	T(s)	T_{opt}(s)	
1	No	9	6	1339	208	9	150	150	7	803	4	9	5	5	9	5	5	
	Yes	9	3	2431	93	6	1408	312	6	1205	8	8(1)	577	200	8	575	197	
0.9	No	9	6	1264	96	7	815	20	7	891	117	9(2)	9	9	9	8	8	
	Yes	9	4	2326	733	7	1196	509	6	1204	6	9(2)	306	306	9	292	292	
0.8	No	9	6	1414	321	9(1)	480	480	8	689	325	8	407	7	9	219	219	
	Yes	9	6	1534	501	7	1084	365	6	1224	35	6	1210	15	8	429	32	
0.7	No	9	9	492	492	8	560	180	7	864	82	7	824	31	9	89	89	
	Yes	9	5	2038	789	6	1828	942	6	1203	5	5	1608	14	7	991	245	
0.6	No	9	8	563	183	8	567	187	7	851	66	6	1218	27	8	430	33	
	Yes	9	4	2347	781	4	2159	358	6	1203	4	4	2019	43	6	1203	4	
Tot		90	57	1575	402	71(1)	1025	336	66	1014	73	71(5)	818	74	82	424	114	

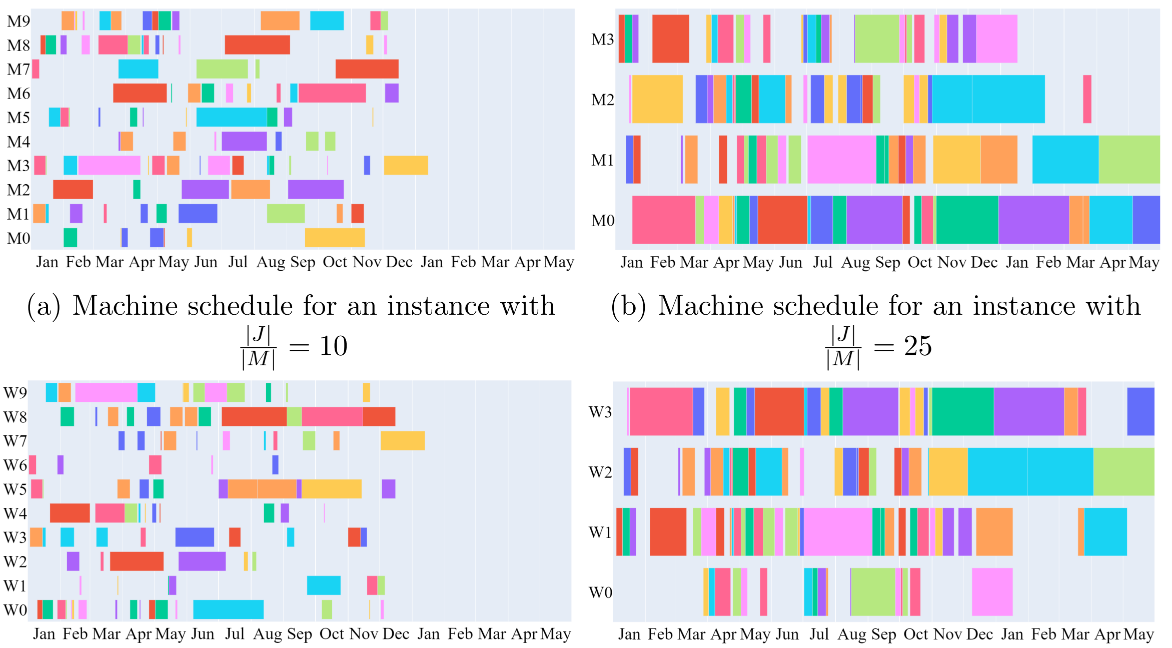
its jobs compatible with 90% of the workers, and (iii) each of its machines compatible with 90% of the workers. Overall, it appears that instances with lower eligibility ratios are harder to solve, and so are instances with precedence/contiguity constraints. A detailed analysis shows that Decomp obtains the best results for instances with eligibility ratio above 0.9, CP shines on instances with precedence/contiguity constraints and eligibility ratio below 0.7, and MILP+ is working well in the remaining cases.

Table 4.6: Summary of results: impact of size and worker/machine/job ratios.

Parameters			Results														
			MILP			MILP+			CP			Decomp			OBA		
Par.	Value	#	# opt	T(s)	T_{opt}(s)	# opt	T(s)	T_{opt}(s)	# opt	T(s)	T_{opt}(s)	# opt	T(s)	T_{opt}(s)	# opt	T(s)	T_{opt}(s)
$\lvert J\rvert$	50	30	28	399	170	29	314	201	24	865	182	23	849	12	29	183	65
	100	30	22	1345	525	24(1)	1019	374	21	1087	10	23(3)	901	80	28	392	163
	200	30	7	2981	946	18	1741	501	21	1088	12	25(2)	704	125	25	697	116
$\frac{\lvert J\rvert}{\lvert M\rvert}$	25	30	14	2209	618	16(1)	2087	764	7	2914	659	15(5)	1950	299	22	1266	417
	10	60	43	1258	332	55	493	211	59	63	4	56	253	13	60	3	3
$\frac{\lvert J\rvert}{\lvert K\rvert}$	25	30	14	2209	618	16(1)	2087	764	7	2914	659	15(5)	1950	299	22	1265	417
	20	30	23	1207	479	26	644	190	30	4	4	26	450	23	30	4	4
	10	30	20	1309	163	29	342	230	29	123	3	30	5	5	30	3	3
$\frac{\lvert M\rvert}{\lvert K\rvert}$	2	30	23	1207	479	26	644	190	30	4	4	26	500	23	30	4	4
	1	60	34	1759	351	45(1)	1215	420	36	1518	131	45(5)	978	103	52	634	178

We report in Table 4.6 the impact of other instance features (in column "Par") on the performance of our methods such as the number of jobs ($|J|$) and the ratios job/machine, job/worker, and machine/worker ($\frac{|J|}{|M|}$, $\frac{|J|}{|K|}$, and $\frac{|M|}{|K|}$, respectively). As expected, instances with 200 jobs tend to be harder to solve than instances with 50 jobs, especially for the two MILP models.

We also notice that instances are easier to solve when the ratios $\frac{|J|}{|M|}$ and $\frac{|J|}{|K|}$ decrease. This can be explained by the fact that the optimal schedule for instances with high $\frac{|J|}{|M|}$ and $\frac{|J|}{|K|}$ ratios tend to be very busy compared with the optimal schedule of instances with lower ratios. This can be observed in Figure 4.3, which reports the best machine schedule (Figures 4.3a-4.3b) and the best worker schedule (Figures 4.3c-4.3d) for two instances, one with $\frac{|J|}{|M|} = \frac{|J|}{|K|} = 10$ (on the left) solved in 0.23s by CP (214.45s for MILP+), and the other with $\frac{|J|}{|M|} = \frac{|J|}{|K|} = 25$ unsolved after 3600s by CP and MILP+. In the two top figures, a rectangle represents the machine occupation for a job. In the two bottom figures, a rectangle represents the worker occupation for at least a job (since a worker may perform either one active job or up to eight passive jobs per day). As far



(a) Machine schedule for an instance with $\frac{|J|}{|M|} = 10$

(b) Machine schedule for an instance with $\frac{|J|}{|M|} = 25$

(c) Worker schedule for an instance with $\frac{|J|}{|K|} = 10$

(d) Worker schedule for an instance with $\frac{|J|}{|K|} = 25$

Figure 4.3: Graphical representation of the optimal schedule for 2 instances with different $\frac{|J|}{|M|}$ and $\frac{|J|}{|K|}$ ratios.

as the ratio $\frac{|M|}{|K|}$ is concerned, we observe that instances with twice as many machines as workers are easier to solve than instances with the same number of machines and workers. This can be explained by our instance generation procedure, since each of the 30 instances in which the ratio $\frac{|M|}{|K|}$ is equal to 2 has a counterpart (i.e., an instance with the same numbers of jobs and machines) in which the ratio is equal to 1 (i.e., with twice as many workers). In other words, supplementary workers increase the instance difficulty (all other parameters being equal). If we focus on the individual performance of each algorithm, we notice that (i) MILP+ is the most efficient approach in practice for instances with 50 and 100 jobs while instances with 200 jobs are solved best with Decomp; (ii) CP is particularly effective on instances where the ratio $\frac{|J|}{|M|}$ is equal to 10, while MILP+ obtains the best performance on instances where that ratio is equal to 25; (iii) Decomp obtains the best results on instances where the ratio $\frac{|J|}{|K|}$ is equal to 10, CP is the best when that ratio is equal to 20, and MILP+ is the best when that ratio is equal to 25; and (iv) CP performs best on instances where the ratio $\frac{|M|}{|K|}$ is equal to 2, while Decomp obtains the best results for instances where this ratio is equal to 1. Once more, an analysis of $T_{opt}(s)$ shows that, when CP and Decomp solve an instance to optimality, they tend to do so relatively fast, whereas MILP and MILP+ require a few hundred seconds on average. Overall, we observe that each algorithm is effective on a subset of instances.

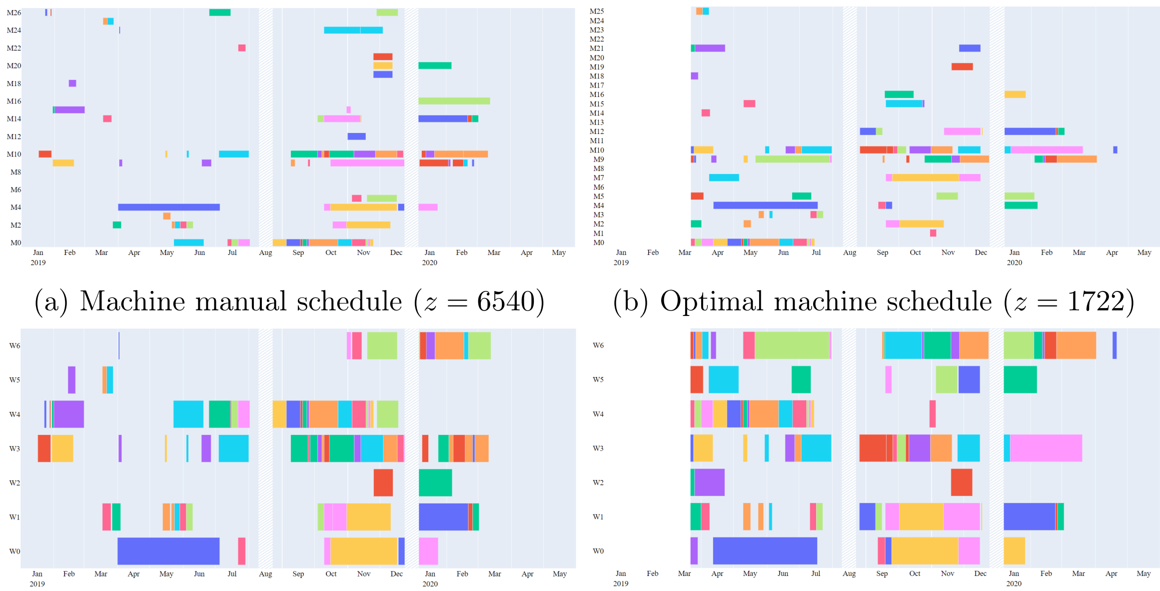
4.5.2 Case study

We now test our approaches on the case study instance provided by Dana Inc. A *project* is defined as a set of one or more *tests* (i.e., jobs) to be executed on the same product. Each project has a release date, a desired due date, and a *priority* (i.e., a weight). Therefore, all jobs belonging to a given project have the same release date, the same due date, and the same weight, and are linked by either a precedence or a contiguity relation.

To transform the original data into a valid instance, we took into account all the jobs taking place in calendar year 2019. We selected the jobs whose release date was between day 01 (January 01) and day 240 (early December – note that Saturdays and Sundays are not counted) and whose due date was before day 400 (mid-July 2020), obtaining an instance with 49 projects, 86 jobs (with an average duration of 10 days), 6 precedence relations, 27 contiguity relations, and a time horizon of 400 days. In the real-world application, there were also some tasks that were released in year 2018 and scheduled for early 2019. Based on the discussions with the company and the limited data we could access, we estimated that an additional 20% of jobs were in that situation. As we had no accurate information regarding the release date or due date of such tasks in the available dataset, we decided to take them into account by forbidding the jobs to be scheduled during the first 2 months and a half of the year.

In the resulting instance, machines are considered available all the time after 2.5-months. Workers are active at most eight hours per working day and are all available eight hours per day except during the weekends, the bank holidays, and the four annual weeks of holidays in which the company is closed. Tasks can be interrupted during the weekend (i.e., a three-day job may start on a Friday morning and finish on the following Tuesday evening) but not during the annual holidays (i.e., every job that was started needs to be finished before the first day of holidays). As explained in the previous section, a job is either active (i.e., it necessitates the constant supervision of a worker) and requires eight worker hours per day or it is passive (i.e., it only involves a punctual supervision) and requires one worker hour per day. Eligibility relations were given by the company and are based on machine features and worker skills. On average, a job is compatible with 14% of machines and 38% of workers, while a machine is compatible with 29% of workers.

MILP+ was able to find an optimal solution in 572 seconds. Figure 4.4 shows the manual schedule used by the company (on the left) and the optimal schedule found by MILP+ (on the right). We obtained an overall decrease of 73% in the total weighted tardiness, with 27 late jobs against 44 jobs in delay in the manual solution. This outstanding result can be explained by two reasons. First, manual solutions are often sub-optimal, even when they are provided by experienced workers (see, e.g., Seifi et al., 2021 and Baykasoğlu and Özbel, 2021). Second, there are some real-world aspects that could not be captured by our model, such as machine breakdown/maintenance, worker-specific unavailability (e.g., due to illness or extra holidays), stochasticity of the job duration due to unexpected delay of product delivery, among others. In summary, it is very unlikely that the best schedule for Dana Inc. can be determined solely with our optimization tools, but the solutions found can serve as a basis to support the company experts in producing the final schedules.



(a) Machine manual schedule ($z = 6540$) (b) Optimal machine schedule ($z = 1722$)

(c) Workers' manual schedule ($z = 6540$) (d) Optimal workers' schedule ($z = 1722$)

Figure 4.4: Graphical representation of solutions for the Dana case study (summer and winter holidays are identified by two white rectangles).

4.5.3 Experiment on realistic instances

To test the scaling potential of our algorithms, we generated a set of realistic instances that include the real-world features of the case study. In particular,

1. the number of projects is set to 40, 80, 120, 160, or 200. For each project, the number of jobs is equal to 1 with probability 0.7, 2 with probability 0.2 (linked by a precedence constraint with probability 0.4 or a contiguity constraint with probability 0.6), or is a random number between 3 and 15 with probability 0.1 (always linked by contiguity constraints);

2. every job is considered active (requiring eight worker-hours per day) except the second job of the projects that contains two tasks linked by a precedence constraint;

3. the job/machine, job/worker, and machine/worker eligibility ratios are set to the realistic values (14%, 38%, and 29%, respectively). We enforced consistency in the eligibility matrices by imposing that every worker that is compatible with job i must also be compatible with at least one machine that is compatible with i;

4. 20% supplementary late projects are included to model all jobs from the previous year. All tasks in these projects have their release date and due date set to the first time period.

We report in Table 4.7 a summary of the results obtained by our algorithms on the realistic instances. Table 4.7 has the same structure as the other tables except for the first column which now indicates the number of projects contained in the instance.

As expected, larger instances (with 120/160/200 projects) are harder to solve, even though we observe that a few instances with 200 projects (4 times as large as the

Table 4.7: Summary of results: impact of size on realistic instances.

Parameters		Results														
		MILP			MILP+			CP			Decomp			OBA		
Size	#	# opt	T(s)	T_{opt}(s)	# opt	T(s)	T_{opt}(s)	# opt	T(s)	T_{opt}(s)	# opt	T(s)	T_{opt}(s)	# opt	T(s)	T_{opt}(s)
40	10	10	526	526	10	332	332	10	61	61	10	16	16	10	10	10
80	10	5	2944	2288	9	1681	1467	3	2521	2	10	108	108	10	42	42
120	10	0	3600	-	2	3309	2143	4	2163	6	10(4)	799	799	10	284	284
160	10	0	3600	-	0	3600	-	4(2)	2172	31	8(6)	1887	1459	10	784	784
200	10	0	3600	-	0	3600	-	1(1)	3248	82	3(3)	3297	2591	4	2946	1964
Tot	50	15	2854	1113	21	2504	991	22(3)	2033	38	41(13)	1222	699	44	813	433

case study) can be solved to optimality. MILP and MILP+ appear to be the weakest approaches on this dataset as they are only able to solve 15 and 21 instances in total, respectively. CP performs slightly better than MILP+ as it is able to solve 22 instances in less than one minute on average (compared with 17 minutes on average for MILP+, if we exclude unsolved instances). Interestingly, we observe that CP is often able to find good-quality (or even optimal) solutions, even for the largest instances, but it struggles in finding good-quality lower bounds. Decomp is the most effective algorithm in terms of the number of instances solved to proven optimality (41 out of 50, among which 13 are unsolved by the other three methods, compared with 22 for CP, among which 3 are unsolved by the other three methods). However, an analysis of T_{opt}(s) shows that Decomp may take more than a thousand seconds to solve an instance whereas CP solves an instance to optimality in less than one hundred seconds. We also emphasize that if one could pre-select the best algorithm for each instance, then the total number of instances solved to optimality would have increased by 3 compared with Decomp, while simultaneously reducing the average computation time of the solved instances by 266 seconds.

We also tested a version of MILP and MILP+ in which `Cplex 22.1.0.0` was used instead of `Gurobi 9.5.1` as solver. The results, reported in Table 4.8, show that both models solved more instances to optimality with Gurobi than they did with Cplex. While one should be cautious before claiming an empirical advantage of one solver over another (see, e.g., Mittelmann, 2020), such a comparison indicates that the poor performance of the two MILP models on this dataset is not caused by the choice of the solver.

Table 4.8: Summary of results: solvers comparison.

Parameters		Gurobi						Cplex					
		MILP			MILP+			MILP			MILP+		
Size	#	# opt	T(s)	T_{opt}(s)	# opt	T(s)	T_{opt}(s)	# opt	T(s)	T_{opt}(s)	# opt	T(s)	T_{opt}(s)
40	10	10	526	526	10	332	332	2	2905	123	3	2970	1501
80	10	5	2944	2288	9	1681	1467	0	3600	-	0	3600	-
120	10	0	3600	-	2	3309	2143	0	3600	-	0	3600	-
160	10	0	3600	-	0	3600	-	0	3600	-	0	3600	-
200	10	0	3600	-	0	3600	-	0	3600	-	0	3600	-
Tot	50	15	2854	1113	21	2504	991	2	3461	123	3	3474	1501

Our next set of experiments, which is reported in Table 4.9, aims at comparing the performance of MILP and MILP+ on the realistic instances. As MILP ran out of memory when solving the instances with 120 projects (which means that the LP relaxation value could not be computed or was not returned), we only compare the two models on instances with 40 and 80 projects. Once more, we observe that, overall, MILP+ is faster on average (1467s to solve instances with 80 projects on average compared with 2288s for MILP, if we exclude unsolved instances) and solves more instances (19 instances with 80 projects or less solved compared with 15 for MILP). Again, these results can mostly be explained by the reduced size of the model (683,197 variables on average for MILP+ on instances with 80 projects compared with 869,438 for MILP). This time, MILP and MILP+ had the same continuous linear relaxation value for all instances.

Table 4.9: Summary of results: comparison between MILP and MILP+.

Parameters		Results													
Size	#	MILP							MILP+						
		# opt	T(s)	T_{opt}(s)	LP	# var.	# cons.	# nzs.	# opt	T(s)	T_{opt}(s)	LP	# var.	# cons.	# nzs.
40	10	10	526	526	993.28	230,910	559,444	53,061,913	10	332	332	993.28	250,044	600,028	32,912,832
80	10	5	2944	2288	874.00	869,438	1,336,717	173,201,182	9	1681	1467	874.00	683,197	1,398,401	81,498,170

Our two final sets of experiments, which are reported in Tables 4.10 and 4.11, aim at evaluating whether our algorithms remain effective when solving related RCPMSP with contiguity constraints. We first tested our three best approaches MILP+, CP, and Decomp on the version of our problem where one wants to minimize the total weighted flow time, which is the same as minimizing the total weighted tardiness when the due date of every task is equal to its release date. Note that the total weighted flow time (a generalization of the total flow time, which is a common measure in system performance) is often used in the PMSP literature (see, e.g., Shabtay and Kaspi, 2004 and Becchetti et al., 2006). As shown in Table 4.10, changing the objective function from weighted tardiness minimization to weighted flow time minimization barely impacts the performance of Decomp and MILP+, whereas it has a negative effect on CP. This could be explained by the fact that the optimal weighted flow time is always strictly positive, meaning that a good quality lower bound (which can be difficult to obtain with CP) is necessary to provide a certificate of optimality.

Table 4.10: Summary of results: impact of size on realistic instances with weighted flow time minimization.

Parameters		Results											
Size	#	MILP+			CP			Decomp			OBA		
		# opt	T(s)	T_{opt}(s)	# opt	T(s)	T_{opt}(s)	# opt	T(s)	T_{opt}(s)	# opt	T(s)	T_{opt}(s)
40	10	10	200	200	10	327	327	10	28	28	10	9	9
80	10	9	1029	744	2	2880	2	10	85	85	10	39	39
120	10	2	3063	1181	4	2162	6	10(4)	578	578	10	310	310
160	10	0	3600	-	2	2887	37	8(6)	1705	1231	8	1232	640
200	10	0	3600	-	1(1)	3248	81	4(4)	2691	1327	5	2339	1078
Tot	50	21	2283	526	19(1)	2301	182	42(14)	1018	526	43	786	328

We then tested the same three approaches on the version of our problem where one wants to minimize the maximum tardiness. As shown in Table 4.11, changing the objective function from weighted tardiness minimization to maximum tardiness minimization has a slight positive impact on the performance of Decomp and MILP+ and an outstanding positive effect on CP. This could be explained by the fact that the optimal maximum tardiness is always a small number (sometimes even zero), meaning that being able to find a good quality upper bound (which is where CP shines) is more important to obtain a certificate of optimality.

Table 4.11: Summary of results: impact of size on realistic instances with maximum tardiness minimization.

Parameters		Results											
Size	#	MILP+			CP			Decomp			OBA		
		# opt	T(s)	T_{opt}(s)	# opt	T(s)	T_{opt}(s)	# opt	T(s)	T_{opt}(s)	# opt	T(s)	T_{opt}(s)
40	10	10	338	338	10	0	0	10	16	16	10	0	0
80	10	8	1646	1157	10	3	3	10	111	111	10	3	3
120	10	1	3346	1060	10	11	11	10	651	651	10	11	11
160	10	0	3600	-	10(1)	28	28	9	2047	1875	10	28	28
200	10	0	3600	-	10(6)	49	49	4	3283	2808	10	49	49
Tot	50	19	2506	721	50(7)	18	18	43	1222	835	50	18	18

4.6 Conclusions

We studied a new parallel machine scheduling problem motivated by the real-world case study of an hydraulic motion components engineering test laboratory. The problem contains release dates, due dates, weights, workforce resource with daily availability, three sets of eligibility constraints, precedence constraints, and contiguity constraints. The introduction of contiguity constraints makes our problem original in the machine scheduling literature. We proposed two original MILP models, a CP formulation, and a combinatorial Benders' decomposition. We also introduced method-specific strategies to model the contiguity constraints for each of the proposed approaches. We tested the effectiveness of the four algorithms with an extensive set of computational experiments, which included random, realistic, and real instances. Our best algorithm, the decomposition approach, was able to solve 71 out of 90 random instances and 41 out of 50 realistic instances to optimality in one hour of computing time. However, we empirically showed that this algorithm does not clearly dominate the others because (i) both the CP formulation and the MILP formulations were able to solve instances that the decomposition approach could not solve and (ii) the CP formulation was able to solve some large realistic instances much faster than the decomposition approach. We also showed that our approaches could easily be extended to take other objective functions into account. We also identified some features that make an instance more difficult to solve (e.g., the presence of precedence/contiguity constraints, and high job/machine and job/worker ratios). We were also able to solve a real instance to optimality with a significant improvement on the total weighted tardiness with respect to the manual solution produced by the company. However, there are a few aspects that we could not take into account in our problem modelization such as machine unavailability caused by a breakdown or by a maintenance intervention, worker unavailability due to illness or supplementary holidays, and the job release date and duration stochasticity caused by unexpected events such as a delay in the product delivery or a machine breakdown happening while the job is being processed.

Future research directions include (i) the development of a hyper-algorithm able to predict and select the most effective method to solve a given instance with supervised learning techniques (i.e., an algorithm able to replicate the results of OBA) and (ii) the study of the dynamic version of the problem to include a rolling horizon aspect where the problem is solved at regular time intervals and where, in each run, a set of job/machine and job/worker assignments is fixed (because it was determined in the previous run), a set of job/machine and job/worker assignments is planned (because it was determined in the previous run but can still be modified if necessary), and a set of job/machine and job/worker assignments is free (because the projects were released after the last run took place or were delayed and need to be rescheduled); the objective function would then be hybridized to also include a minimization component for disruption (number of planned assignments that are modified). While Dana Inc. is interested in the one-year-ahead solutions obtained by our approaches for scheduling and prevision purposes, the company also needs to have the tools to regularly update those schedules and predictions later on when new data are available.

References

Afzalirad, M., & Rezaeian, J. (2016). Resource-constrained unrelated parallel machine scheduling problem with sequence dependent setup times, precedence constraints and machine eligibility restrictions. *Computers & Industrial Engineering*, *98*, 40–52.

Baker, K. R., & Trietsch, D. (2019). *Principles of sequencing and scheduling*. John Wiley & Sons.

Baykasoğlu, A., & Özbel, B. K. (2021). Modeling and solving a real-world cutting stock problem in the marble industry via mathematical programming and stochastic diffusion search approaches. *Computers & Operations Research*, *128*, 105173.

Becchetti, L., Leonardi, S., Marchetti-Spaccamela, A., & Pruhs, K. (2006). Online weighted flow time and deadline scheduling. *Journal of Discrete Algorithms*, *4*(3), 339–352.

Becker, C., & Scholl, A. (2006). A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research*, *168*(3), 694–715.

Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, *4*(1), 238–252.

Blazewicz, J., Lenstra, J. K., & Rinnooy Kan, A. H. G. (1983). Scheduling subject to resource constraints: Classification and complexity. *Discrete Applied Mathematics*, *5*(1), 11–24.

Bockmayr, A., & Hooker, J. N. (2005). Constraint programming. *Handbooks in Operations Research and Management Science*, *12*, 559–600.

Bruno, J., & Downey, P. (1978). Complexity of task sequencing with deadlines, set-up times and changeover costs. *SIAM Journal on Computing*, *7*(4), 393–404.

Caselli, G., Delorme, M., Iori, M., & Magni, C. A. (2022). Mixed integer linear programming for a real-world parallel machine scheduling problem with workforce and precedence constraints. In L. Amorosi, P. Dell'Olmo, & I. Lari (Eds.), *Optimization in artificial intelligence and data sciences* (pp. 61–71). Springer International Publishing.

Chen, Z.-L. (2004). Simultaneous job scheduling and resource allocation on parallel machines. *Annals of Operations Research*, *129*(1), 135–153.

Cheng, T. C. E., & Sin, C. C. S. (1990). A state-of-the-art review of parallel-machine scheduling research. *European Journal of Operational Research*, *47*(3), 271–292.

Chudak, F. A., & Shmoys, D. B. (1999). Approximation algorithms for precedence-constrained scheduling problems on parallel machines that run at different speeds. *Journal of Algorithms*, *30*(2), 323–343.

Codato, G., & Fischetti, M. (2006). Combinatorial benders' cuts for mixed-integer linear programming. *Operations Research*, *54*(4), 756–766.

Côté, J.-F., Dell'Amico, M., & Iori, M. (2014). Combinatorial benders' cuts for the strip packing problem. *Operations Research*, *62*(3), 643–661.

Daniels, R. L., Hoopes, B. J., & Mazzola, J. B. (1997). An analysis of heuristics for the parallel-machine flexible-resource scheduling problem. *Annals of Operations Research*, *70*, 439–472.

Dell'Amico, M., Delorme, M., Iori, M., & Martello, S. (2019). Mathematical models and decomposition methods for the multiple knapsack problem. *European Journal of Operational Research*, *274*(3), 886–899.

Delorme, M., Iori, M., & Martello, S. (2017). Logic based benders' decomposition for orthogonal stock cutting problems. *Computers & Operations Research*, *78*, 290–298.

Delorme, M., & Santini, A. (2022). Energy-efficient automated vertical farms. *Omega*, *109*, 102611.

Edis, E. B., Oguz, C., & Ozkarahan, I. (2013). Parallel machine scheduling with additional resources: Notation, classification, models and solution methods. *European Journal of Operational Research*, *230*(3), 449–463.

Edis, E. B., & Ozkarahan, I. (2011). A combined integer/constraint programming approach to a resource-constrained parallel machine scheduling problem with machine eligibility restrictions. *Engineering Optimization*, *43*(2), 135–157.

Edis, E. B., & Ozkarahan, I. (2012). Solution approaches for a real-life resource-constrained parallel machine scheduling problem. *The International Journal of Advanced Manufacturing Technology*, *58*(9), 1141–1153.

Fang, K., Wang, S., Pinedo, M. L., Chen, L., & Chu, F. (2021). A combinatorial benders decomposition algorithm for parallel machine scheduling with working-time restrictions. *European Journal of Operational Research*, *291*(1), 128–146.

Fleszar, K., & Hindi, K. S. (2018). Algorithms for the unrelated parallel machine scheduling problem with a resource constraint. *European Journal of Operational Research*, *271*(3), 839–848.

Fuchigami, H. Y., & Rangel, S. (2018). A survey of case studies in production scheduling: Analysis and perspectives. *Journal of Computational Science*, *25*, 425–436.

Gedik, R., Rainwater, C., Nachtmann, H., & Pohl, E. A. (2016). Analysis of a parallel machine scheduling problem with sequence dependent setup times and job availability intervals. *European Journal of Operational Research*, *251*(2), 640–650.

Geoffrion, A. M. (1972). Generalized benders decomposition. *Journal of Optimization Theory and Applications*, *10*(4), 237–260.

Gökgür, B., Hnich, B., & Özpeynirci, S. (2018). Parallel machine scheduling with tool loading: A constraint programming approach. *International Journal of Production Research*, *56*(16), 5541–5557.

Graham, R. L., Lawler, E. L., Lenstra, J. K., & Rinnooy Kan, A. H. G. (1979). Optimization and approximation in deterministic sequencing and scheduling: A survey. In *Annals of discrete mathematics* (pp. 287–326, Vol. 5). Elsevier.

Hall, N. G., Potts, C. N., & Sriskandarajah, C. (2000). Parallel machine scheduling with a common server. *Discrete Applied Mathematics*, *102*(3), 223–243.

Hartmann, S., & Briskorn, D. (2022). An updated survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, *297*(1), 1–14.

Hooker, J. N. (2006). An integrated method for planning and scheduling to minimize tardiness. *Constraints*, *11*(2), 139–157.

Hooker, J. N. (2011). *Logic-based methods for optimization: Combining optimization and constraint satisfaction*. John Wiley & Sons.

Hooker, J. N., & Ottosson, G. (2003). Logic-based benders decomposition. *Mathematical Programming*, *96*(1), 33–60.

Hu, X., Bao, J.-S., & Jin, Y. (2010). Minimising makespan on parallel machines with precedence constraints and machine eligibility restrictions. *International Journal of Production Research*, *48*(6), 1639–1651.

Jain, V., & Grossmann, I. E. (2001). Algorithms for hybrid MILP/CP models for a class of optimization problems. *INFORMS Journal on Computing*, *13*(4), 258–276.

Janiak, A., Janiak, W. A., Krysiak, T., & Kwiatkowski, T. (2015). A survey on scheduling problems with due windows. *European Journal of Operational Research*, *242*(2), 347–357.

Karlsson, E., & Rönnberg, E. (2022). Logic-based benders decomposition with a partial assignment acceleration technique for avionics scheduling. *Computers & Operations Research*, *146*, 105916.

Khatami, M., & Salehipour, A. (2021a). A binary search algorithm for the general coupled task scheduling problem. *4OR*, *19*(4), 593–611.

Khatami, M., & Salehipour, A. (2021b). Coupled task scheduling with time-dependent processing times. *Journal of Scheduling*, *24*(2), 223–236.

Khatami, M., Salehipour, A., & Cheng, T. C. E. (2020). Coupled task scheduling with exact delays: Literature review and models. *European Journal of Operational Research*, *282*(1), 19–39.

Koné, O., Artigues, C., Lopez, P., & Mongeau, M. (2011). Event-based MILP models for resource-constrained project scheduling problems. *Computers & Operations Research*, *38*(1), 3–13.

Koulamas, C. (1994). The total tardiness problem: Review and extensions. *Operations Research*, *42*(6), 1025–1041.

Kramer, A., Dell'Amico, M., & Iori, M. (2019). Enhanced arc-flow formulations to minimize weighted completion time on identical parallel machines. *European Journal of Operational Research*, *275*(1), 67–79.

Kramer, R., Dell'Amico, M., & Iori, M. (2017). A batching-move iterated local search algorithm for the bin packing problem with generalized precedence constraints. *International Journal of Production Research*, *55*(21), 6288–6304.

Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G., & Shmoys, D. B. (1993). Sequencing and scheduling: Algorithms and complexity. In S. Graves, A. Rinnooy Kan, & P. Zipkin (Eds.), *Logistics of production and inventory* (pp. 445–522). Elsevier.

Lee, C. Y. (2004). Machine scheduling with availability constraints. In J. Y. T. Leung (Ed.), *Handbook of scheduling: Algorithms, models, and performance analysis* (pp. 22-1–22-14). CRC Press.

Lenstra, J. K., & Rinnooy Kan, A. H. G. (1978). Complexity of scheduling under precedence constraints. *Operations Research*, *26*(1), 22–35.

Leung, J. Y.-T., & Li, C.-L. (2016). Scheduling with processing set restrictions: A literature update. *International Journal of Production Economics*, *175*, 1–11.

Li, Y., Côté, J.-F., Coelho, L. C., & Wu, P. (2022). Novel efficient formulation and matheuristic for large-sized unrelated parallel machine scheduling with release dates. *International Journal of Production Research*, *60*(20), 6104–6123.

Lovato, D., Guillaume, R., Thierry, C., & Battaia, O. (2022). Managing disruptions in aircraft assembly lines with staircase criteria. *International Journal of Production Research*, *(forthcoming)*.

Mischek, F., & Musliu, N. (2021). A local search framework for industrial test laboratory scheduling. *Annals of Operations Research*, *302*(2), 533–562.

Mittelmann, H. D. (2020). Benchmarking optimization software-a (hi) story. *SN operations research forum*, *1*(1), 2.

Mokotoff, E. (2001). Parallel machine scheduling problems: A survey. *Asia-Pacific Journal of Operational Research*, *18*(2), 193–242.

Pinedo, M. L. (2016). *Scheduling - theory, algorithms, and systems*. Springer International Publishing.

Potts, C. N., & Kovalyov, M. Y. (2000). Scheduling with batching: A review. *European Journal of Operational Research*, *120*(2), 228–249.

Pritsker, A. A. B., Waiters, L. J., & Wolfe, P. M. (1969). Multiproject scheduling with limited resources: A zero-one programming approach. *Management Science*, *16*(1), 93–108.

Reddy, N. S., Ramamurthy, D. V., Padma Lalitha, M., & Prahlada Rao, K. (2022). Integrated simultaneous scheduling of machines, automated guided vehicles and tools in multi machine flexible manufacturing system using symbiotic organisms search algorithm. *Journal of Industrial and Production Engineering*, *39*(4), 317–339.

Seifi, C., Schulze, M., & Zimmermann, J. (2021). A new mathematical formulation for a potash-mine shift scheduling problem with a simultaneous assignment of machines and workers. *European Journal of Operational Research*, *292*(1), 27–42.

Seo, K., Joung, S., Lee, C., & Park, S. (2022). A closest benders cut selection scheme for accelerating the benders decomposition algorithm. *INFORMS Journal on Computing*, *34*(5), 2804–2827.

Shabtay, D., & Kaspi, M. (2004). Minimizing the total weighted flow time in a single machine with controllable processing times. *Computers & Operations Research*, *31*(13), 2279–2289.

Su, H., Pinedo, M. L., & Wan, G. (2017). Parallel machine scheduling with eligibility constraints: A composite dispatching rule to minimize total weighted tardiness. *Naval Research Logistics*, *64*(3), 249–267.

Ulusoy, G., Sivrikaya-Şerifoğlu, F., & Bilge, Ü. (1997). A genetic algorithm approach to the simultaneous scheduling of machines and automated guided vehicles. *Computers & Operations Research*, *24*(4), 335–351.

Ventura, J. A., & Kim, D. (2000). Parallel machine scheduling about an unrestricted due date and additional resource constraints. *IIE Transactions*, *32*(2), 147–153.

Węglarz, J., Józefowska, J., Mika, M., & Waligóra, G. (2011). Project scheduling with finite or infinite number of activity processing modes–a survey. *European Journal of operational research*, *208*(3), 177–205.

Chapter 5

A Real-World Application of Demand Forecasting in the Perishable Food Industry[1]

Abstract

Demand forecasting is a critical area of study with widespread applications in decision-making, inventory management, and production planning across various industries. Accurate forecasts are particularly needed in the perishable food industry, characterized by products with highly variable demand trends and short storage, leading to severe operational challenges along the supply chain. This work provides an extensive analysis of statistical and machine learning methods for predicting daily demand of perishable food products on the real dataset of a large company operating in the processed meat industry. By exploring different training procedures and evaluating their forecasting error, we demonstrate that machine learning methods are able to outperform classical approaches of the forecasting literature. We also perform a simulation on the same dataset, showing that by the use of forecasting models significant improvements can be obtained on the economic performance of the company. The research thus helps bridging the gap between theoretical advancements and practical implementation in the demand forecasting literature.

5.1 Introduction

The global market of consumer goods is today faced with the growing need to expand both the number and the variety of its products. To satisfy the large and personalized market demand and, on the other hand, to limit their economic and environmental impact, industrial businesses require efficient solutions for process optimization. Among many optimization practices, such as inventory management and workforce planning, demand forecasting is acquiring more and more attention from production managers and business leaders (Spiliotis et al., 2022). To achieve this goal, companies collect

[1]Mucciarini, M., Caselli, G., Iori, M., Lippi, M. (2024). A Real-World Application of Demand Forecasting in the Perishable Food Industry (under review).

historical product demand data through their management systems and organize it into databases. This data is then analyzed using statistical or Machine Learning (ML) tools to predict future demand values. Accurate forecasts are especially needed in the food industry, where perishable products require reactive markets with efficient stock management and flexible production to manage variable demand trends and high stock turnover rates. Optimization is needed in all the activities across the food supply chain from production to distribution, including, for instance, inventory management and vehicle routing (Violi et al., 2020). Therefore, forecasting tools represent a competitive advantage not only for individual companies operating in the food industry, but also for the whole food supply chain (Dellino et al., 2018; Lemma et al., 2014).

The literature on the theory and practice of demand forecasting is vast (see, e.g., the work by Petropoulos et al., 2022 and references therein). A notorious area of application is demand forecasting for perishable goods in the retail industry (Fildes et al., 2022) and, more specifically, in the food industry. With short production times and shelf-life, fresh food requires accurate forecasts at the level of both production and sales. In this work, we focus our attention on the former. The majority of works in this area deal with the application of statistical forecasting methods such as simple exponential smoothing models and autoregressive integrated moving average (ARIMA) models (Mor et al., 2019; Silva et al., 2019). Although such methods have the advantage of low computational cost, ML methods have been shown to achieve a better forecasting performance within a limited or even negligible increase in the computational effort (Barrow and Kourentzes, 2018; Huber and Stuckenschmidt, 2020). Our attempt is indeed to provide a successful application of ML methods to demand forecasting in the perishable food industry field.

Our work deals with a real-world demand forecasting problem originated from a long-term research collaboration with Inalca SpA, which, with an annual revenue above 3 billion euro, is the leading company in the Italian market of processed meat. We study the problem of forecasting the daily demand of different processed meat products required by supermarkets and food retailers to a production facility located in Calabria and serving one of the main market segments of Inalca in Italy. The typical process of Inalca is to start producing every day (Monday-Saturday) in the early morning to fulfill the orders for the same day, before receiving them. Then, orders are collected during the day and production is adjusted accordingly. If demand was underestimated, more packages are produced by the end of the day (at the cost of additional setups). On the other hand, if demand was overestimated, the additional packages already completed but not shipped are stored in a refrigerated warehouse. Packages can be stored for at most 24 hours (and used to meet the demand of the following day), after which they must be discarded. Production is operated on a six-day working week, from Monday to Saturday. The daily demand of products is affected by a general intermittent trend, in which a day with high volume is typically followed by a day with low volume. The demand time series of each item strongly depends on its specific characteristics, including seasonality (e.g., some items are ordered only in some seasons or during a specific period, such as boiled beef meat in winter and lamb products during Easter), novelty (e.g., some items have been in the market for years, while others have just entered it) and market promotions. Demand time series are also affected by special events (e.g., some larger-size items are introduced for special promotion weeks) and holidays. The company's goal is to improve production efficiency and minimize waste by accurately predicting market trends.

In this work, our objective is to apply different ML methods to the Inalca demand forecasting problem and to compare their performance, also with respect to classical approaches from the forecasting literature. Comparisons are made by measuring the forecasting error in terms of the number of ordered packages, using a real dataset provided by the company on a 12-month period. We applied support vector regression, random forests, and artificial neural networks. We compared the performance of such methods with classical baseline models, among which we included the one currently used by the company. Since demand forecasting has a practical importance in inventory management, as recognized by the forecasting literature (Barrow and Kourentzes, 2018; Teunter and Duncan, 2009; Yuna et al., 2023), we further investigated the performance of our forecasting models in terms of inventory-related aspects. In particular, we simulated the implementation of our most effective model on the Inalca test set, assuming that the company sets their daily production quantities of each item equal to the forecast values minus the quantity of packages stored in the refrigerated warehouse the day before.

In this work, we provide the following main contributions:

- We propose various ML methods to address a real-world demand forecasting problem in the perishable food industry. This area of application remains unexplored in demand forecasting but holds great importance for the economic and environmental sustainability of food supply chains;

- We introduce a novel forecasting approach that combines different ML models, by selecting in the training phase the best model for each category of items. Categorization is based on the theory by Syntetos et al. (2005) and allows to distinguish between intermittent time series, which are notoriously difficult to forecast, and more regular ones. Extensive computational comparisons on a real data set, which we have made publicly available on the web, show the superior performance of the combined forecasting approach with respect to the expert system currently in use at the company and several other statistical and ML methods;

- We develop a simulation algorithm and apply it to the inventory management of the considered case study. With this algorithm, we show the positive impact of the selected forecasting model on selected key performance indicators, namely the number of dynamic adjustments to the scheduled production, the size of the warehouse, and the quantity of discarded products.

The remainder of this paper is organized as follows. In Section 5.2, we provide a summary of related works on demand forecasting problems, with a focus on real-world applications. Section 5.3 describes the demand forecasting problem considered in this study, the data provided by the company, and the related data cleaning process. The ML methods used in our study are presented in Section 5.4, whereas the computational results are described in Section 5.5 and the inventory simulation algorithm is presented in Section 5.6. Section 5.7 concludes the paper with final remarks and future research directions.

5.2 Related works

A recent survey on forecasting (Petropoulos et al., 2022) presents a comprehensive overview of theoretical methods as well as practical applications of this challenging task. As mentioned in this work, forecasting for demand management is a crucial step for all supply chain optimization practices, including inventory management, production planning, and workforce scheduling. Among the approaches proposed throughout the years for the problem of supply chain demand forecasting, we mention classical time-series modeling approaches, clustering methods, and ML techniques (Seyedan and Mafakheri, 2020).

In the last decades, demand forecasting methods have been applied to several real-world industrial contexts, such as fashion industry (Ren et al., 2020), retail industry (Spiliotis et al., 2022) and tourism (Cankurt and Subasi, 2022), just to cite some. The perishable food supply chain has always attracted special attention in the literature, due to its specific characteristics: the short life of products, the high demand fluctuation, the dependence on weather conditions, the constantly increasing food demand, and the public concern for food safety (Lemma et al., 2014). Dellino et al. (2018) proposed a decision support system integrating sales forecasting methods and order planning tools for an Italian supply chain of packaged, fresh, and highly perishable products. The system first selects the best forecasting approach among classical statistical methods including auto-regressive models (ARIMA and ARIMAX) and transfer functions, and then uses the forecast values to feed a multi-objective model for order planning. Silva et al. (2019) presented a case study in the delicatessen industry, for which they developed a demand forecasting model combining moving average, exponential smoothing, and ARIMA models to support production planning and inventory management. Results showed a reduction of Mean Absolute Deviation, Mean Absolute Percentage Error, and Mean Squared Error performance measures between 20% and 30% with respect to the model used by the company. Mor et al. (2019) compared statistical models for the forecasting of dairy products demand with an application in the milk processing industry, concluding that multiple regression and Holt-Winters were the best forecasting models. The authors also stated that their methodology may be applied to different products in the processed food industry.

Our field of application is the processed meat industry, which is still an unexplored area of application for demand forecasting. As reported by Mena et al. (2014), who studied the UK food supply network, inaccurate forecasting is one of the main causes of waste for meat products in the retail industry, in addition to weather-changing conditions, market promotions, and others. The application is thus of great interest.

ML is quite famously applied to demand forecasting. Carbonneau et al. (2008) compared the performance of neural networks and support vector machine models with naïve, moving average, and linear regression models, showing significant improvements in real-world data from the supply chain of Canadian foundries. Ali et al. (2009) studied the stock keeping unit (SKU) demand forecasting problem for a medium-sized European grocery retailer. On their dataset, consisting of 76 weeks of SKU time series, regression trees outperformed statistical methods by up to 65% in the presence of promotions, while the performance of statistical and ML methods was comparable in the cases of no promotions. Aburto and Weber (2007) proposed a hybrid approach for demand forecasting of a Chilean supermarket that combines ARIMA models and neural networks to reduce inventory levels and sales failures. More recently, ML methods for time series forecasting were successfully employed to predict the number of future cases of COVID-19, as reported by Ribeiro et al. (2020).

ML methods have been used also for demand forecasting under special events. A successful application of ML methods to forecast demand in outlier data (e.g., holidays, special events, promotions) is given by Barrow and Kourentzes (2018) in the context of the call arrival forecasting problem for a call center operated by a European entertainment company. A similar work was presented by Huber and Stuckenschmidt (2020) on the case of a bakery chain. In addition to forecasting on special days, ML methods have also been successfully applied in big data applications. For instance, Cerqueira et al. (2019) showed how ML methods usually outperform statistical ones when larger training datasets are available. Despite all these results, as reported by many works (see, e.g., Barrow and Kourentzes, 2018 and references therein), the forecasting literature is still dominated by statistical methods. A few exceptions are the works by Huber and Stuckenschmidt (2021) and Miguéis et al. (2022), reporting the effective use of different artificial neural networks and decision trees. Moreover, integrated and hybrid methods have recently been developed to improve the performance of single methods in forecasting, for instance, financial indicators and stock prices (Hajirahimi and Khashei, 2021; Shahvaroughi Farahani and Razavi Hajiagha, 2021). Our work moves in this direction, as it is an attempt to show the potential of ML methods to effectively forecast demand in the perishable food industry.

5.3 Problem formulation and data description

We hereby formulate our forecasting problem and describe the data used in the experimental evaluation.

5.3.1 Forecasting problem

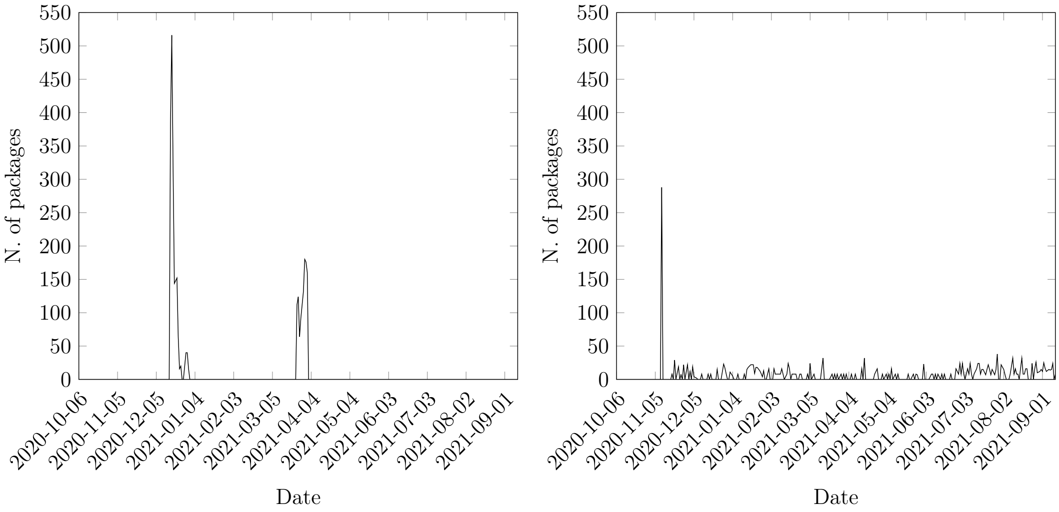
We are given a set of n items, each representing a different product. For each item k we are given a time series $v^k = \{v_1^k, \ldots, v_m^k\}$ that represents the number of ordered packages of the item in a series of m consecutive time units. The granularity of the time series defines the nature of the problem: in this work, we consider daily data and predict the number of ordered packages of the k-th item at day t, which is v_t^k, given the time series up to day $t-1$, which is $\{v_1^k, \ldots, v_{t-1}^k\}$.

To address the problem from an ML point of view, it is convenient to formulate it as a supervised regression task. To this aim, the time series of the k-th item has to be transformed into a dataset of N_k supervised pairs $\mathcal{D}^k = \{(x_i^k, y_i^k)\}_{i=1}^{N_k}$ where the goal is to predict the target value y_i^k given the input (observed) variables encoded in vector x_i^k. In our case, the target variable is an element in the time series, $y_i^k = v_t^k$, whereas the input variable is a time window of W past elements $x_i^k = \{v_{t-W}^k, \ldots, v_{t-1}^k\}$. The width of the time window W that is used to predict the target variable is a problem-dependent hyperparameter, whose value is to be chosen experimentally. In general, the dataset $\mathcal{D}^k$ is split into a training set $\mathcal{D}_{train}^k$ (used to fit the model) and a test set $\mathcal{D}_{test}^k$ (used for evaluation). Any ML approach for regression can be then applied to such a dataset.

The description of the supervised regression task above concerns a single item. To forecast the number of ordered packages for all the items, we consider three alternatives: (i) training a different regressor for each item, independently; (ii) merging the training sets of all the items into a single one and then training a single forecasting model for the whole set of items; and (iii) grouping similar items into categories and then training a different model for each category. In the second scenario, the overall supervised regression task is given by $\mathcal{D} = \bigcup_{k=1}^{n} \mathcal{D}^k$. In the third scenario, we have a different dataset for each category $\mathcal{C}$, namely $\mathcal{D}^{\mathcal{C}} = \bigcup_{k \in \mathcal{C}} \mathcal{D}^k$. This third approach is original in similar forecasting applications and is thus explained in detail in Section 5.4.3.

5.3.2 Data

The original dataset provided by the company includes 390 real time series of various items (i.e., processed meat products), including burgers, steaks, and sausages, referring to a 12-month period from October 06, 2020 up to September 11, 2021 (293 days excluding Sundays). Each time series represents the daily demand of items, spanning from Monday to Saturday, ordered by customers, which are mainly supermarkets and grocery stores. Each time series of the dataset includes 293 observations of daily demand. From an exploratory analysis of such data, we observed that each article presents several days with no demand (i.e., with a zero value in the time series). Around 30% of the articles have at least 150 non-zero observations, while the number of non-zero observations is smaller for (i) *spot items*, which have at least 20 non-zero observations (a value empirically determined with the company's decision makers) in the training set (see Figure 5.1a) and (ii) *new items*, which present an initial sequence of zero values, followed by a sudden, large peak corresponding to the launch of the products on the market, before stabilizing in their regular pattern of orders (see Figure 5.1b). Spot items are those typically introduced in the market for a specific customer and a specific promotional event. The company has accurate prior information for these events. New items are those introduced in the market after October 2020. Providing an accurate forecast for these two types of items would be critical as well as unnecessary in some situations. Therefore, we cleaned the dataset by removing all the spot items and the new items introduced after the training period. We kept only new items whose initial peak is entirely included in the training set and having at least 20 training non-zero observations (i.e., they are not spot items). For such new items, we cut the first peak of their time series and, thus, considered their first regular observation after the peak as the initial value of their (shorter) training set. After the data cleaning process, we obtained a reduced dataset of 190 time series with an average length equal to 176 non-zero observations. Following the work by Syntetos et al. (2005), we categorized the time series based on the values of the coefficient of variation (CV^2) and the average inter demand interval (ADI) of each time series, which indicate, respectively, to what extent non-zero observations vary in terms of numerical value demand (the greater is CV^2, the more the values are variable), and how frequent are non-zero observations in the series (the greater is ADI, the greater is the average distance between non-zero observations). As reported by the authors, the categorization of time series patterns allows to detect the best forecasting method for different time series. Using the thresholds proposed in their work, we defined smooth ($\mathrm{CV}^2 < 0.49$, $\mathrm{ADI} < 1.32$), intermittent ($\mathrm{CV}^2 < 0.49$, $\mathrm{ADI} \geq 1.32$), lumpy ($\mathrm{CV}^2 \geq 0.49$, $\mathrm{ADI} \geq 1.32$), and erratic ($\mathrm{CV}^2 \geq 0.49$, $\mathrm{ADI} < 1.32$) time series.



(a) Spot item (b) New item

Figure 5.1: Example of the demand time series for a spot (left) and a new (right) item.

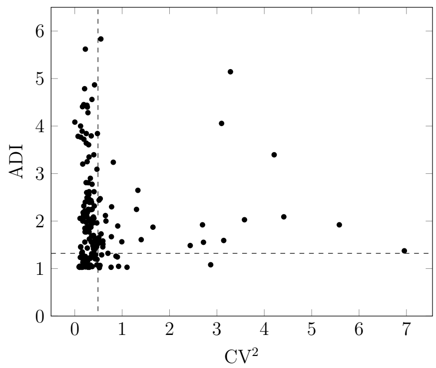
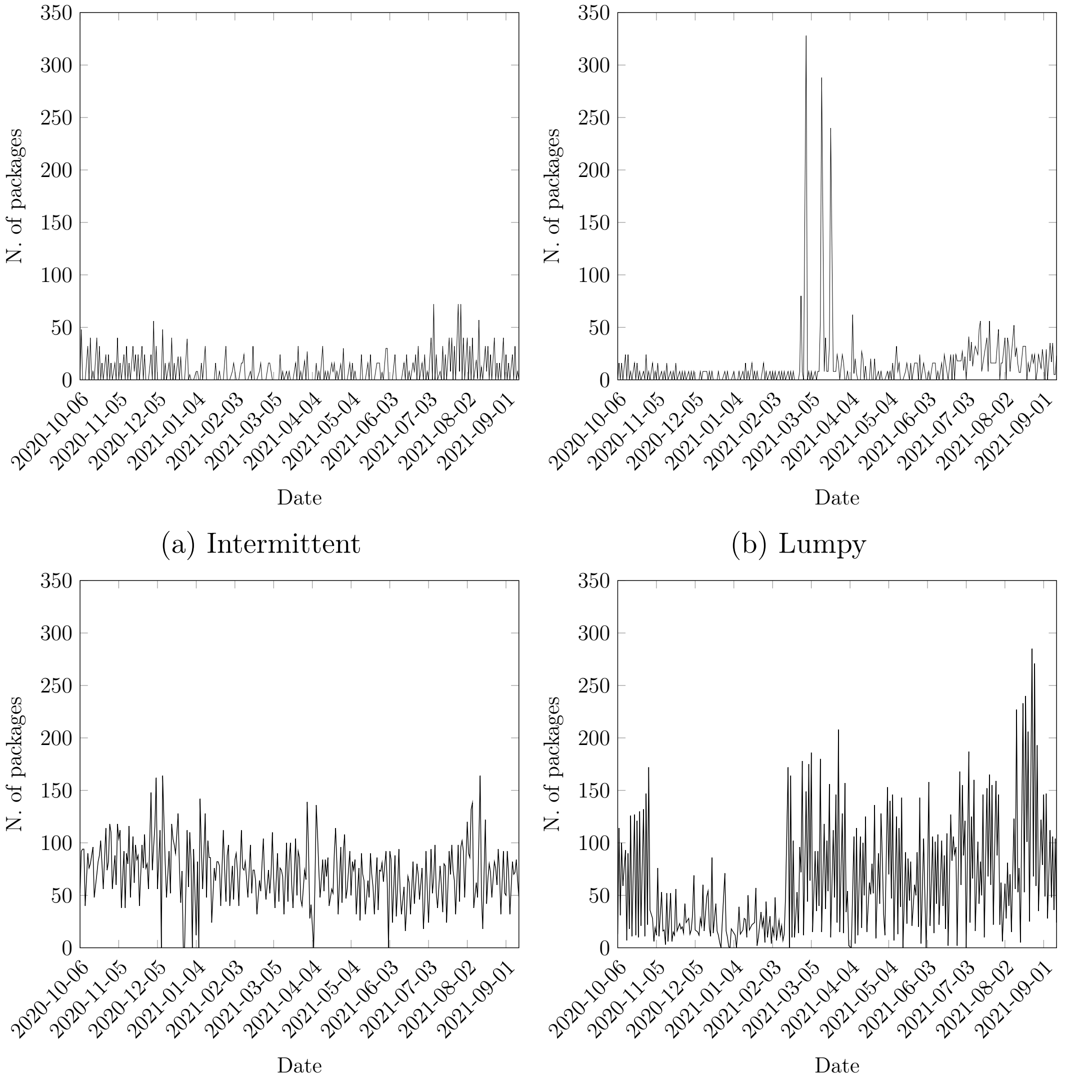


Figure 5.2: Categorization of time series based on CV^2 and ADI.

In Figure 5.2, we represent the 190 series in a scatter plot (each dot is a time series) and their categorization into 99 intermittent, 48 smooth, 34 lumpy, and 9 erratic items. We obtained that 78% of the series have low erraticness (i.e., low CV^2), mainly due to the regularity of demand which is typical of perishable food products for supermarkets. On the other hand, two main classes of items emerged: items that are ordered every day (low intermittency) and items that are ordered every two or more days (high intermittency). Figure 5.3 shows a representative example of time series for each category in our dataset.

5.4 Methods

In this section, we describe our baseline and ML methods to address our demand forecasting problem. The selected methods are the ones that are more frequently used in the related literature (see, e.g., Sarker, 2021) and that showed the most promising preliminary results on our dataset. We pick representatives from the family of statistical approaches, such as linear regression, ARIMA and SARIMA, and a method specifically



(a) Intermittent (b) Lumpy

(c) Smooth (d) Erratic

Figure 5.3: Example time series of the four categories based on CV^2 and ADI.

designed for demand forecasting (ADIDA). As for ML approaches, we choose a bagging approach (random forest), a statistical learning one (support vector regression) and artificial neural networks (multi-layer perceptron). More sophisticated methods such as Long Short-Term Memory networks showed slightly worse results in our preliminary analysis. Then, we present our original forecasting approach that trains and tests a different forecasting model (i.e., the best among all baseline and ML methods in terms of forecasting error) for each category of items. Categorization is performed as described in Section 5.3.2.

5.4.1 Baseline

We selected five baseline methods: a seasonal naïve forecast, the Aggregate-Disaggregate Intermittent Demand Approach (ADIDA) method, the classical ARIMA method, its seasonal variant (SARIMA), and linear regression (LR).

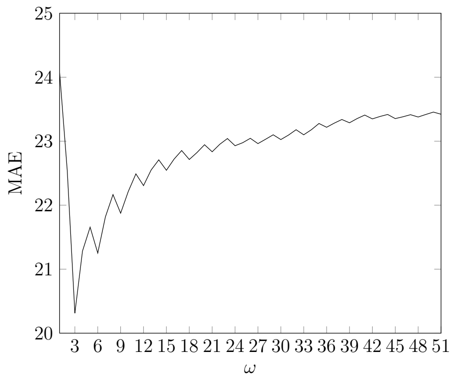


Figure 5.4: MAE of our ADIDA implementation as a function of window size ω.

The naïve forecast predicts the future value of a time series as equal to the last known past value of the series. This simple method (see, e.g., Barrow and Kourentzes, 2018), as well as its seasonal variants (see, e.g., Huber and Stuckenschmidt, 2020 and Spiliotis et al., 2022), have been widely used as valid benchmarks in the forecasting literature. Our naïve baseline model predicts the target value v_t^k as the value of the time series of the k-th item in the same day of the week before:

$$\hat{v}_t^k = v_{t-6}^k \tag{5.1}$$

The ADIDA method proposed by Nikolopoulos et al. (2011) aggregates an intermittent demand time series into lower-frequency "time buckets" of arbitrary length (to be defined) to reduce the presence of zero observations. Then, any forecasting method can be applied to the new aggregated series, before disaggregating the result on the original series with or without a specific weighting procedure. Several works in the literature propose ADIDA as a valid forecasting approach (Li and Lim, 2018; Nikolopoulos, 2021). As more than 50% of the time series in our dataset showed high intermittency with a typical weekly trend of alternate "up" and "down" days, as reported in Section 5.3.2, we combined ADIDA with a seasonal naïve model. We tested different values of the aggregation window by aggregating the daily demand on "up" (Monday-Wednesday-Friday) and "down" (Tuesday-Thursday-Saturday) days and we chose the one that returned the lowest MAE. Given the aggregation window ω, and using uniform weights for disaggregation, the resulting formula for our ADIDA is:

$$\hat{v}_t^k = \frac{1}{\omega} \sum_{j=1}^{\omega} v_{t-2j}^k \tag{5.2}$$

which predicts v_t^k as the average on the last ω observations considering only alternate days (i.e., seasonal naïve with a two-day seasonality). As shown in Figure 5.4, the minimum error was given by a three-day aggregation window.

The ARIMA method proposed by Box and Jenkins (1976) is one of the most classical statistical approaches for time series forecasting. To forecast future values of a given time series, ARIMA combines three distinct models, namely the autoregressive model (AR), the moving average model (MA), and the integrated model (I), which use, respectively, observations, errors, and differences from the past values of the time series. This method is often used as a baseline for evaluating the performance of new and different forecasting models in the literature, but it often turns out to be a very strong competitor (Da Veiga et al., 2014; Ramos et al., 2015). The classical method has been extended in many ways: we hereby consider its variant that incorporates seasonal variations, resulting in the so-called SARIMA method (see, e.g.,Arunraj and Ahrens, 2015). In this study, we evaluate both ARIMA and SARIMA using the `auto-arima` function from the `pmdarima` library. The function determines the optimal hyperparameters of the ARIMA model, which are the orders (i.e., the number of values considered) of the autoregressive and moving average components (in a range from 1 to 10), and the integrated component (by executing a differencing test). To implement SARIMA, we need to specify the seasonal differencing hyperparameter, which is equal to six in our case (since our periodicity is the six-day week from Monday to Saturday) and the `auto-arima` function finds the best values of the seasonal counterpart of the three ARIMA order parameters. Optimal ARIMA parameters are re-estimated by the function at every forecast step (i.e., every day), making such models typically effective in many time series forecasting problems.

LR makes the hypothesis of a linear relationship between the dependent variables x and the independent variable, which is the target y (see, e.g., Lewis-Beck and Lewis-Beck, 2015). LR is typically considered among the baseline approaches for forecasting problems: see, e.g., Carbonneau et al. (2008) for supply chain demand forecasting and Huber and Stuckenschmidt (2020) for daily retail demand forecasting. Such and similar applications show the trade-off between the simplicity of the method and its effectiveness when compared to more advanced methods (e.g., artificial neural networks, support vector regressors). LR represents a strong baseline when the linear approximation is a good fit for the dependency between the target and the features that are to be learned.

5.4.2 Machine learning

The selected ML methods are: random forest, support vector regression, and multi-layer perceptron. For each method, we describe the methodological basics, present the models, and briefly summarize common applications and practices.

Random Forest

A Random Forest (RF) is an ensemble method that averages the prediction of multiple decision trees (in case of classification) or regression trees (in case of regression) to reduce overfitting and achieve better generalization with respect to single trees taken individually (Breiman, 2001; Sarker, 2021).

The structure of a regression tree is learned inductively from a collection of supervised examples, by splitting the training data into smaller and smaller groups (nodes) until each group contains only data points with similar prediction values and no further split is needed. The algorithm used to learn the tree structure can be summarized as follows: at each node of the tree, a feature is selected to perform the best split of training data based on some performance criterion; then, training data are partitioned into subsets (the children nodes) based on the values of the selected feature. This process is recursively applied to children nodes until they are sufficiently homogeneous in terms of target variables. When a new prediction on a test example must be performed, the tree is traversed following the decision rules starting from the root node, until a leaf node is reached; then, the prediction is made by taking the average value among the training examples located in that leaf.

Although they have many interesting properties, especially in terms of interpretability, unfortunately, regression tree models are prone to overfitting. Ensemble methods such as RF are usually preferred, as they combine a set of individual regression trees to reduce overfitting and increase model generalization capability. To make the final prediction, an RF aggregates over the ensemble by averaging the prediction outputs of individual trees. The main hyperparameters to train an RF are the number T of trees, the number F of features tested at each split of each tree, the minimum node size M of each tree (Dudek, 2015), and the maximum depth d of each tree. The two latter hyperparameters are strictly connected: the tree expansion, in fact, is stopped as soon as the maximum depth d (i.e., the number of tree levels) is reached or the leaf size of a node is smaller or equal to M.

Some successful RF forecasting applications were presented by Dudek (2015), where RF showed as good performance as an artificial neural network for the short-term forecasting of electricity load, and Punia et al. (2020), who successfully combined RF with a Long-Short Term Memory (LSTM) network for a retail demand forecasting problem with daily, weekly, and monthly horizons. Such a hybrid method outperformed all the baseline methods (including RF and LSTM taken individually).

Support Vector Regression

Support Vector Regression (SVR) extends the framework of Support Vector Machines to the case of regression. In its linear formulation, SVR aims at finding the hyperplane $f(x) = \ell^T x + b$ that maximizes the number of points that lie at a distance lower than some tolerance ϵ from the hyperplane (Schölkopf et al., 2000). Given a generic training set $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{Q}$ of Q observations, the regression task is to solve the following quadratic optimization problem:

$$\min_{\ell,b,\xi,\xi^*} \quad \frac{1}{2}\|\ell\|^2 + C\sum_{i=1}^{Q}(\xi_i + \xi_i^*) \tag{5.3}$$

$$\text{s. t.} \quad (\ell^T x_i + b) - y_i \leq \epsilon + \xi_i, \qquad i = 1, \ldots, Q \tag{5.4}$$

$$y_i - (\ell^T x_i + b) \leq \epsilon + \xi_i^*, \qquad i = 1, \ldots, Q \tag{5.5}$$

$$\xi_i, \xi_i^* \geq 0, \qquad i = 1, \ldots, Q \tag{5.6}$$

where $\ell \in \mathbb{R}^W$ and $b \in \mathbb{R}$ are the variables to be optimized that define the hyperplane; $C > 0$ and $\epsilon > 0$ are the hyperparameters of the model, which represent, respectively, the regularization coefficient and the tolerance; ξ_i and ξ_i^* are the non-negative slack variables introduced to minimize the error of data point i if such a point falls outside the ϵ threshold. The objective function in (5.3) combines the minimization of the error on training data with a regularization term that privileges smoother functions (i.e., with a smaller norm $\|\ell\|^2$). Constraints in (5.4) and (5.5) define the distance of each point from the hyperplane, and constraints (5.6) state the domain of the slack variables. The optimization problem is solved via its dual formulation (Schölkopf et al., 2000) and the solution is an approximate function $f(x) = \sum_{i=1}^{Q}(-\alpha_i + \alpha_i^*) y_i \langle x, x_i \rangle + b$, where α_i, α_i^* are the (learned) dual variables, and $\langle \cdot, \cdot \rangle$ is the dot product. Training examples with

$(-\alpha_i + \alpha_i^*) \neq 0$ are called support vectors and are the only points concurring to the definition of the hyperplane. This linear formulation can be extended to the non-linear case by replacing the dot product with a positive, semi-definite (and non-linear) kernel function $K(\cdot, \cdot)$ that measures the similarity between training examples. For our SVR we tried different linear and non-linear kernel functions.

Several successful applications of SVR for time-series forecasting problems appear in the literature, especially dealing with data characterized by a strong non-linearity. Among many, we mention the annual electricity load forecasting problem (see, e.g., Wang et al., 2012 and Zhang and Hong, 2021).

Multi Layer Perceptron

Artificial neural networks are nowadays one of the most popular approaches in ML. Inspired by neural connections in the human brain (Bishop and Nasrabadi, 2006), they have been one of the pioneering approaches in artificial intelligence, knowing periods of great success and disillusionment. With the advent of deep learning they have been (and still are, nowadays) successfully applied to a wide variety of real-world problems.

A Multi-Layer Perceptron (MLP) is an artificial neural network composed of a stack of layers, each containing a number of elementary computational units, named neurons. Each neuron combines the signals received from all the neurons in the previous layer and transmits the signal to all the neurons in the next layer by means of an activation function. Such a function is typically non-linear, allowing the MLP to learn complex non-linear relationships between input and output variables (Hornik et al., 1989). Each neuron in the first (input) layer is fed with one of the dependent variables for the task at hand. The output layer is used to predict the target variable, which is a real number in the case of regression. Each connection of the network has a weight, which is learned during the training phase of the network so as to minimize the prediction error on the entire training dataset.

Suppose to address our demand forecasting task with an MLP with a single hidden layer. The output layer produces $\hat{v}_t^k$, which is the predicted number of ordered packages of the k-th item at day t. The input layer has size W and is fed by the set of input variables $v_{t-W}^k, \ldots, v_{t-1}^k$. The weights of the first layer are represented by $w_{t-W}^0, \ldots, w_{t-1}^0$, and the weights of the second layer are represented by $w_1^1, \ldots, w_h^1$, where h is the number of neurons in the (single) hidden layer. Given the non-linear activation function f, the output of our MLP is given by:

$$\hat{v}_t^k = f(\sum_{j=1}^{h} w_j^1 f(\sum_{i=1}^{W} w_{t-i}^0 v_{t-i}^k)) \tag{5.7}$$

Equation (5.7) can be easily generalized to MLPs with multiple hidden layers. The more hidden layers and number of neurons per layer, the more parameters w the network has to tune during the training. However, a trade-off between such parameters and the risk of local minima and overfitting must be considered in the use of MLPs (see, e.g., Heidari et al., 2016).

MLPs are usually trained via the backpropagation algorithm (Bishop and Nasrabadi, 2006), a technique based on gradient descent. To design a specific network architecture and perform the training, several hyperparameters must to be defined. In this work, we focus on the following: (i) the number of hidden layers H and the number of neurons h_i for each hidden layer $i = 1, \dots, H$; (ii) the activation function f that modulates the signal transmitted to the following layer of the network (e.g., simple threshold function or more complex non-linear functions); and (iii) the solver algorithm to find the best weights of the network during backpropagation.

Among many recent applications of MLPs to demand forecasting problems, we mention the work by Huber and Stuckenschmidt (2020), who compared several statistical and ML approaches on a daily retail demand forecasting problem, and Spiliotis et al. (2022), who solved a daily demand problem for a Greek retail company comparing MLP, RF, SVR, and other traditional methods.

5.4.3 Combined approach

Different demand forecasting applications have shown that alternative approaches based on the ensembling, hybridization, aggregation, and combination of different ML models may result in improved forecasting results with respect to those of the classical application of single models (Dudek, 2015; Hajirahimi and Khashei, 2021). Selecting the best model is not a trivial task and the trade-off between the increase in forecasting accuracy and the corresponding increase in computational time and complexity must be taken into account. In this work, we implemented an alternative approach that is original in the selected application. The new approach is based on the result of the initial categorization procedure and the idea that selecting the best forecasting model for each category of items may result in improved overall accuracy.

Our approach, simply called "Combined" hereafter, trains one model for each of the four categories of smooth, intermittent, lumpy, and erratic items (as presented in Section 5.3.2). The name indicates that items of the same category are grouped together and on each group the method with minimum forecasting error is selected to achieve the most effective combination of forecasting methods across all items. Note that this approach is not a hybrid method in the traditional sense, where a single time series is decomposed and trained using different methods. Instead, each set of time series data (one for each item) is trained using distinct models, namely one per category. On the other hand, it is not properly even an ensemble method, because we neither combine nor aggregate the predictions of multiple models, but we just call a different model for each product category. The model for items belonging to category $\mathcal{C}$ is trained on the training set $\mathcal{D}^{\mathcal{C}}_{train}$. This approach is novel in similar applications and provides four smaller models, each one for a different category of items based on the erraticness and intermittency of their time series, potentially having stronger similarities among each other. Once the four models are trained, during the test phase, given an item of a specific category, the corresponding model is selected to produce the desired forecast.

5.5 Experiments

In this section, we present the empirical evaluation of our methodology on the real dataset of Inalca. First, we define our experimental setup. We then summarize the results, comparing the forecasting accuracy of the different baseline and ML methods on the daily demand data. In particular, we investigate which model provides the best daily forecast, we show how our combined approach improves the forecasting performance of classical training procedures, and we also simulate the impact of the different model forecasts on the size and dynamics of the company warehouse. All models were implemented in `Python 3.10.4`, using the open source data science libraries `scikit-learn`, `pmdarima`, `pandas`, and `numpy`. We trained and tested our models on a virtual machine Intel(R) Xeon(R) Gold with 2.30 GHz and 16 GB of RAM memory, running under Windows 10 Pro N. Our dataset can be downloaded from the online repository https://github.com/regor-unimore/Perishable-Food-Demand-Forecasting.

In the online repository we removed the names of the items and assigned an incremental index to each of them to anonymize the available dataset and identify the item position within the columns. The columns contain special values identified by "-1" for days corresponding to holidays. For such days, no prediction is made in case they are included in the test set. However, when they are present in the training set, to avoid null values in the input data computed via windowing, linear interpolation is applied taking the corresponding days of the two precedent weeks. This interpolation method is employed to ensure continuity in the data and enable the accurate training process of each model.

5.5.1 Experimental setup

As previously discussed in Section 5.3.2, our dataset includes 190 time series for the daily demand of different items expressed in the number of packages. The total length of each time series is equal to 293 observations, referring to a 12-month period from October 06, 2020 up to September 11, 2021 (excluding Sundays). We chose such a period length in accordance with the company as a representative time frame of the typical fluctuations of the related market due to several factors, including varying weather conditions, holidays, new customers entering the market, new items, and the typical product life cycle. The company also excluded a significant impact of COVID-19 pandemics on products demand during this period based on a statistical analysis of historical data. We split our dataset into a training set and a test set. The training set includes the first six-month period from October 06, 2020 to April 17, 2021, which counts 167 daily data for 173 items, and fewer data for the remaining 17 new items, for which only the regular part of the demand was considered in the training set (see Section 5.3.2). The test set comprises the following period from April 19, 2021 to September 11, 2021, which counts 126 daily data for each of the 190 items.

Regarding each of the ML methods presented in Section 5.4, we defined a set of possible values for the model hyperparameters and selected the best values by means of a grid search cross validation algorithm on the training set (see Section 5.5.2): the training set was split into five folds and all possible combinations of the model parameters were computed. Each combination was tested five times on the five different folds (using the union of the residual four folds as the training set), and the combination providing the overall minimum forecasting macro-averaged error was used for the evaluation on the test set. After this model selection phase, all the ML methods were trained following three different procedures:

1. the traditional "series-by-series" training procedure (see, e.g., Semenoglou et al., 2021), where a different forecasting model is trained for the k-th item on $\mathcal{D}^k_{train}$ —

Table 5.1: Grid search cross validation on RF, SVR, and MLP models.

Model	Hyperparameter	Values
RF	Number of trees T	200, 500, 700, 1000
	Maximum tree depth d	No limit, 1, 2, 3
	Minimum node size M	2, 3, 4
SVR	Kernel function K	Linear, RBF, Sigmoid
	Regularization parameter C	1.5, 5, 10, 30, 50, 100, 150, 200, 250
	Accuracy ϵ	0.1, 0.5, 1, 5, 9, 10, 13
MLP	Hidden layer(s) size $(n_1, \ldots, n_H)$	(100), (100, 50), (200), (200, 50)
	Activation function f	Tanh, ReLU
	Optimizer	Stochastic Gradient Descent, Adam

in our case, we trained each ML model 190 times, obtaining one model for each item;

2. the "single" training procedure (successful, for example, in the M4 competition, see Makridakis et al., 2020), where each model is trained only once on the entire training set $\mathcal{D}_{train}$, allowing to capture similarities across different time series;

3. the intermediate training solution "per category" embedded in the proposed combined approach, which trains one model for each of the four categories of smooth, intermittent, lumpy, and erratic items (as presented in Section 5.4.3).

In the case of training procedures (1) and (2), we applied a min-max scaling pre-processing technique to the training set, obtaining comparable values in the range $[0, 1]$. The aim of this scaling procedure was to reduce the impact of different scales of time series values in the same dataset (e.g., large and small daily ordered quantities for different items). Note that the scaling procedure was applied before the grid search cross-validation on the entire dataset $\mathcal{D}_{train}$ in the case of the single training procedure, and on set $\mathcal{D}^{\mathcal{C}}_{train}$ in the case of training per category.

5.5.2 Hyperparameter setting

As reported in Section 5.3.1, the hyperparameter W, which indicates the window size of the past time series values to be used as input to the predictors, was set experimentally. We tested three different values, namely $W = 6, 12, 18$ (i.e., 1, 2 or 3 working weeks). For each window size value, we generated a dataset $\mathcal{D}$ on which we performed the scaling process (when needed), the grid search, the training, and the testing processes. Table 5.1 shows the grid search cross-validation hyperparameters and the corresponding values for our RF, SVR, and MLP models. Any additional setting, including the ones of LR, was left to the default value as implemented in the `scikit-learn` library. The grid search algorithm generated all the combinations of hyperparameters and found the best one by evaluating a macro-averaged error over multiple validation folds, as explained in Section 5.5.1.

5.5.3 Evaluation criterion

To evaluate and compare the forecasting performance of the tested approaches, we used a standard metric for regression tasks, namely the Mean Absolute Error (MAE). We call $\mathcal{D}^k_{test} = \{(x^k_i, y^k_i)\}_{i=1}^{N^k_{test}}$ the portion of the test set observations related to the k-th item, with $k = 1, \dots, n$. We remind the reader that y^k_i and $\hat{y}^k_i$ are, respectively, the target and the predicted values of the forecast for observation i of the k-th item on the test set $\mathcal{D}^k_{test}$. The MAE, which is commonly used in the forecasting literature (see, e.g., Petropoulos et al., 2022), was then computed as

$$\text{MAE} = \frac{1}{n} \sum_{k=1}^{n} \left(\frac{1}{N^k_{test}} \sum_{i=1}^{N^k_{test}} |y^k_i - \hat{y}^k_i| \right) \tag{5.8}$$

Thus, we first computed the MAE for every time series (of the k-th item) individually, and then we averaged the results across all the time series of items $k = 1, \dots, n$.

5.5.4 Forecasting results

Tables 5.2-5.4 show the forecasting performance of the ML methods presented in Section 5.4, on the three different training procedures (single dataset, series-by-series, per category), respectively. We highlighted in bold the minimum error value of every column for MAE, which is our reference error measure. Our results suggest that MLP with window size $W = 12$ and RF with window size $W = 12$ are the best models respectively, in the single training setting and in the series-by-series training setting.

Table 5.2: MAE comparison with single training.

Method	$W = 6$	$W = 12$	$W = 18$
RF	20.90	19.90	19.97
SVR	20.77	20.14	**19.68**
MLP	**20.54**	**19.10**	20.16

Table 5.3: MAE comparison with series-by-series training.

Method	$W = 6$	$W = 12$	$W = 18$
RF	**21.78**	**21.47**	**21.91**
SVR	21.85	21.55	22.37
MLP	22.28	22.21	24.07

Table 5.4 instead shows the results of every ML method on each subset of data related to each category of items (smooth, intermittent, lumpy, erratic). We observe that SVR clearly outperforms the other methods on intermittent and erratic categories, with the latter being by far (and not surprisingly) the most complicated one to forecast (i.e., its forecasting error is almost one order of magnitude larger than the one of smooth, intermittent, and lumpy categories). RF and MLP are both good forecasting models for the daily demand of the smooth and lumpy categories of items.

Table 5.4: MAE comparison with training by category.

Category	Method	$W = 6$	$W = 12$	$W = 18$
Smooth	RF	**25.05**	**23.67**	**24.41**
	SVR	25.92	25.20	24.45
	MLP	26.29	24.87	24.83
Intermittent	RF	11.13	10.97	11.03
	SVR	**10.45**	**10.22**	**10.25**
	MLP	10.85	10.99	10.56
Lumpy	RF	15.75	15.68	15.62
	SVR	15.20	15.11	**14.70**
	MLP	**14.48**	**14.42**	16.39
Erratic	RF	**111.80**	116.40	115.70
	SVR	120.80	**107.60**	**110.10**
	MLP	114.30	120.90	127.0

Table 5.5 summarizes the results of our experiments, also showing those obtained with the baseline methods described in Section 5.4.1, and the forecasting model currently used by the company. This model, which we consider as our sixth baseline method and we call "Company" thereafter, is the first result of our long-term research collaboration with Inalca. It is an expert system based on an adjusted naïve algorithm. The adjustments were encoded in the model by collecting and analyzing additional information provided by the company business managers based on their knowledge of the system (e.g., characteristics of the items, customer behavior, and typical trends of the related market including commercial promotion and holidays).

The results show the MAE of the best configuration (i.e., training procedure and window size W) for each method. The best window size of each model is indicated in the first column "Method" after the name of the corresponding method (e.g., "(W18)" stands for $W = 18$). The first two blocks report the minimum MAE of Company and baseline methods, followed by ML methods using single and series-by-series training, respectively. The last block refers to the "Combined" model obtained by combining the best predictors for each category (i.e., RF (W12) on the smooth category, SVR (W12) on intermittent and erratic categories, and MLP (W12) on the lumpy one). We observe that the overall best model in terms of MAE is the combined approach, trained by category.

More generally, we observe that, regarding ML methods, the single training fashion is more effective in our context of application with respect to the series-by-series training. This means that our ML models are able to gain additional information from the overall dataset, which combines many similar time series (related to different items with similar trends) covering a time horizon of only one year (which reflects the short stability of the related market). Moreover, we found that our proposal to train different models for different categories further improved the performance of the system, showing the best overall result in terms of MAE (equal to 18.98) also with respect of all our baseline methods. Combined is the best overall method in our experimental evaluation and a tailored training procedure is particularly effective in such a forecasting context.

Table 5.5: Summary of results on the best training procedure and window size W. "Combined" indicates a combination of RF (W12) on smooth, SVR (W12) on intermittent and erratic, and MLP (W12) on lumpy.

Method	Training	MAE
Company	series-by-series	21.68
Naïve	series-by-series	23.19
ADIDA		20.31
ARIMA		21.13
SARIMA		20.38
LR (W6)		20.65
RF (W12)	single	19.90
SVR (W18)		19.68
MLP (W12)		19.10
RF (W12)	series-by-series	21.47
SVR (W12)		21.55
MLP (W12)		22.21
Combined	per category	**18.98**

5.6 Inventory simulation

We computed three practical measures of inventory efficiency to evaluate the realistic impact of the forecasting methods on the company operations, namely: (i) the daily manual adjustments adj_t^k with respect to the forecasts $\hat{y}_t^k$ required to satisfy the demand y_t^k; (ii) the size of the warehouse h_t^k on the t-th day for the k-th item; and (iii) the daily number of potential discarded packages $disc_t^k$ due to product expiration. All these three quantities are to be minimized. All three indicators have a relevant practical implication for the warehouse management of the company. The first one is related to productivity because frequent dynamic adjustments increase setup times and cause productivity losses. The second one is strictly constrained by the capacity of the warehouse and, moreover, the larger the warehouse, the higher the costs for refrigerating stocks. The third one is the most important one because it is related to direct losses of final products. Note that our simulation represents a cautious behavior for discarded packages with respect to the real-world process. Indeed, if during the day the company receives a demand whose value is lower than the one estimated by the algorithm, the production manager interrupts the production process and thus avoids the excess of production. However, in our simulation we do not know when the actual demand information is released, so we cautiously count every unit in excess as a discarded package.

Algorithm 5.5 shows how we computed the three indicators, adj_t^k, h_t^k, and $disc_t^k$, for each t-th day and k-th item in our simulation process. At day t, the first step is to update the number of days after production for every package of the k-th item in the warehouse and the number of discarded packages (i.e., the packages that reached their expiration date). Then, the daily effective production P_t^k is calculated and compared with the actual demand y_t^k. We evaluate the balance B_t^k of the k-th item in the t-th day as the sum of the current stock h_t^k plus the produced quantity P_t^k minus the actual

Algorithm 1 Inventory simulation algorithm

for $k = 1, \dots, n$ **do**
  $h_0^k \leftarrow 0$, $adj_0^k \leftarrow 0$, $disc_0^k \leftarrow 0$
  for $t = 1, \dots, N_{test}^k$ **do**
    Update h_t^k and $disc_t^k$ based on expiration date
    $P_t^k \leftarrow max(0, \hat{y}_t^k - h_t^k)$, $B_t^k \leftarrow P_t^k + h_t^k - y_t^k$
    if $B_t^k > 0$ **then**
      Case 1: No adjustments, update inventory
      adj_t^k += 0
      if $y_t^k < h_t^k$ **then**
        Remove y_t^k packages from h_t^k (by oldest) and store P_t^k in h_t^k
      else
        Store B_t^k packages (smaller or equal to P_t^k) in h_t^k
      end if
    else if $B_t^k = 0$ **then**
      Case 2: No adjustments, empty inventory
      adj_t^k += 0, $h_t^k \leftarrow 0$
    else
      Case 3: Adjustments, empty inventory
      adj_t^k += $|B_t^k|$, $h_t^k \leftarrow 0$
    end if
  end for
end for

Figure 5.5: Inventory simulation algorithm.

demand y_t^k. When $B_t^k > 0$, the demand is fulfilled first by emptying the stock (starting from older packages), and then by taking (some of) the produced packages. When $B_t^k = 0$, the demand is completely fulfilled by the whole stock plus the produced quantity and no adjustment is necessary. When $B_t^k < 0$, the actual demand is not fulfilled by the stock plus the produced quantity. Therefore, the stock is fully empty and an adjustment adj_t^k (i.e., an additional quantity of units) equal to $|B_t^k|$ is required in production.

To compare the different approaches, we computed the following average daily metrics: (i) Average Number of Dynamic Adjustments (ANDA), (ii) Total Warehouse Size (TWS), and (iii) Average Number of Discarded Packages (ANDP), as follows:

$$\text{ANDA} = \frac{1}{N_{test}} \sum\nolimits_{t=1}^{N_{test}} \left(\frac{1}{n} \sum\nolimits_{k=1}^{n} adj_t^k \right) \tag{5.9}$$

$$\text{TWS} = \frac{1}{N_{test}} \sum\nolimits_{t=1}^{N_{test}} \left(\sum\nolimits_{k=1}^{n} h_t^k \right) \tag{5.10}$$

$$\text{ANDP} = \frac{1}{N_{test}} \sum\nolimits_{t=1}^{N_{test}} \left(\frac{1}{n} \sum\nolimits_{k=1}^{n} disc_t^k \right) \tag{5.11}$$

Among the ML approaches, we chose the Combined model, which achieved the best forecasting performance with respect to MAE (see Section 5.5.4), and we compare it with the selected baseline, namely the Company model. Table 5.6 gives a summary of the simulation results, reporting, for each metric, the absolute values ("Value") and

Table 5.6: Inventory simulation results.

Method	ANDA		TWS		ANDP	
	Value	Gap	Value	Gap	Value	Gap
Company	11.76		1585.28		3.81	
Combined	11.28	**- 4%**	1253.47	**-21%**	2.58	**-32%**

the percentage improvement of Combined with respect to Company values ("Gap"). The results show that Combined improves ANDA by 4%, TWS by 21%, and ANDP by 32% with respect to the method currently employed by the company. In particular, Combined produces a notable reduction in ANDP, which is considered by the company the most important indicator. We thus conclude that the Combined model is effective both in terms of its forecasting accuracy and its practical implications for inventory management.

5.7 Conclusions

Demand forecasting is a crucial as much as a difficult task for businesses working in the perishable food supply chain, which is characterized by extremely short production, storage, and delivery times. Our work was inspired by the real-world demand forecasting problem of an Italian company working in the processed meat industry. The goal was to provide an accurate daily forecast for 190 different products. We were given a real dataset of 12-month-long time series, each referring to one of the products.

We classified such products into four categories based on the demand categorization by Syntetos et al. (2005). Then, we proposed three ML methods, namely an RF, an SVR, and an MLP, and a combined approach where a different forecasting model is trained on each category of items, and the overall model is obtained by combining the best models across the categories. We also analyzed the impact of the number of past observations given as input to each model on the performance in predicting the target value of daily demand. To further assess the performance of our forecasting model, we simulated inventory management based on daily forecasts generated by our approach. The outcomes of our research can be summarized as follows. We evaluated the forecasting accuracy of our models and six baselines on a given test set, measuring the overall MAE. Our ML models outperformed all the baseline methods, including ARIMA and SARIMA, which are typically effective in several time series forecasting applications proposed in the literature. Our combined method of providing different models per category demonstrated superior performance in terms of MAE. Our inventory simulation included relevant indicators of inventory efficiency, namely the number of manual adjustments in daily forecasts, the size of the warehouse, and the quantity of discarded packages. With our combined approach, theese three indicators are reduced by 4%, 21%, and 32%, respectively, compared to the company's existing state-of-the-art algorithm. These results confirmed the practical benefits of using a data-driven technique in such a challenging scenario, a conclusion further supported by the practical outcomes of the algorithm, which is now operational within one of Inalca's production plants.

As future research, we are first of all interested in deploying to the company the new Combined approach. Then, we would like to follow different interesting research directions, namely: (i) the use of more sophisticated ML approaches for time series forecasting, such as recurrent neural networks; (ii) the introduction into our models of additional information about the related market, such as weather conditions; (iii) the evaluation of the impact of more frequent model re-training (e.g., weekly or monthly); (iv) the integration of our demand forecasting output as input of the related production scheduling problem to be studied to further optimize the company operations. We are also interested in the application of combined forecasting methods to similar problems arising in different segments of the food industry, studying the potential savings from both an economic and a sustainable point of view.

References

Aburto, L., & Weber, R. (2007). Improved supply chain management based on hybrid demand forecasts. *Applied Soft Computing*, *7*(1), 136–144.

Ali, Ö. G., Sayın, S., Van Woensel, T., & Fransoo, J. (2009). SKU demand forecasting in the presence of promotions. *Expert Systems with Applications*, *36*(10), 12340–12348.

Arunraj, N. S., & Ahrens, D. (2015). A hybrid seasonal autoregressive integrated moving average and quantile regression for daily food sales forecasting. *International Journal of Production Economics*, *170*, 321–335.

Barrow, D., & Kourentzes, N. (2018). The impact of special days in call arrivals forecasting: A neural network approach to modelling special days. *European Journal of Operational Research*, *264*(3), 967–977.

Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*. Springer.

Box, G. E., & Jenkins, G. M. (1976). Time series analysis. Forecasting and control. *Holden-Day Series in Time Series Analysis*.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Cankurt, S., & Subasi, A. (2022). Tourism demand forecasting using stacking ensemble model with adaptive fuzzy combiner. *Soft Computing*, *26*, 3455–3467.

Carbonneau, R., Laframboise, K., & Vahidov, R. (2008). Application of machine learning techniques for supply chain demand forecasting. *European Journal of Operational Research*, *184*(3), 1140–1154.

Cerqueira, V., Torgo, L., & Soares, C. (2019). Machine learning vs statistical methods for time series forecasting: Size matters. *arXiv preprint arXiv:1909.13316*.

Da Veiga, C. P., Da Veiga, C. R. P., Catapan, A., Tortato, U., & Da Silva, W. V. (2014). Demand forecasting in food retail: A comparison between the Holt-Winters and ARIMA models. *WSEAS Transactions on Business and Economics*, *11*(1), 608–614.

Dellino, G., Laudadio, T., Mari, R., Mastronardi, N., & Meloni, C. (2018). A reliable decision support system for fresh food supply chain management. *International Journal of Production Research*, *56*(4), 1458–1485.

Dudek, G. (2015). Short-term load forecasting using random forests. In D. Filev, J. Jabłkowski, J. Kacprzyk, M. Krawczak, I. Popchev, L. Rutkowski, V. Sgurev,

E. Sotirova, P. Szynkarczyk, & S. Zadrozny (Eds.), *Intelligent Systems' 2014* (pp. 821–828). Springer.

Fildes, R., Ma, S., & Kolassa, S. (2022). Retail forecasting: Research and practice. *International Journal of Forecasting*, *38*(4), 1283–1318.

Hajirahimi, Z., & Khashei, M. (2021). Parallel hybridization of series (phos) models for time series forecasting. *Soft Computing*, *25*, 659–672.

Heidari, E., Sobati, M. A., & Movahedirad, S. (2016). Accurate prediction of nanofluid viscosity using a multilayer perceptron artificial neural network (MLP-ANN). *Chemometrics and Intelligent Laboratory Systems*, *155*, 73–85.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366.

Huber, J., & Stuckenschmidt, H. (2020). Daily retail demand forecasting using machine learning with emphasis on calendric special days. *International Journal of Forecasting*, *36*(4), 1420–1438.

Huber, J., & Stuckenschmidt, H. (2021). Intraday shelf replenishment decision support for perishable goods. *International Journal of Production Economics*, *231*, 107828.

Lemma, Y., Kitaw, D., & Gatew, G. (2014). Loss in perishable food supply chain: An optimization approach literature review. *International Journal of Scientific & Engineering Research*, *5*(5), 302–311.

Lewis-Beck, C., & Lewis-Beck, M. (2015). *Applied regression: An introduction* (Vol. 22). Sage.

Li, C., & Lim, A. (2018). A greedy aggregation–decomposition method for intermittent demand forecasting in fashion retailing. *European Journal of Operational Research*, *269*(3), 860–869.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). The M4 competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, *36*(1), 54–74.

Mena, C., Terry, L. A., Williams, A., & Ellram, L. (2014). Causes of waste across multi-tier supply networks: Cases in the UK food sector. *International Journal of Production Economics*, *152*, 144–158.

Miguéis, V. L., Pereira, A., Pereira, J., & Figueira, G. (2022). Reducing fresh fish waste while ensuring availability: Demand forecast using censored data and machine learning. *Journal of Cleaner Production*, *359*, 131852.

Mor, R. S., Jaiswal, S. K., Singh, S., & Bhardwaj, A. (2019). Demand forecasting of the short-lifecycle dairy products. In H. Chahal, J. Jyoti, & J. Wirtz (Eds.), *Understanding the role of business analytics* (pp. 87–117). Springer.

Nikolopoulos, K. (2021). We need to talk about intermittent demand forecasting. *European Journal of Operational Research*, *291*(2), 549–559.

Nikolopoulos, K., Syntetos, A. A., Boylan, J. E., Petropoulos, F., & Assimakopoulos, V. (2011). An aggregate–disaggregate intermittent demand approach (ADIDA) to forecasting: An empirical proposition and analysis. *Journal of the Operational Research Society*, *62*(3), 544–554.

Petropoulos, F., Apiletti, D., Assimakopoulos, V., Babai, M. Z., Barrow, D. K., Taieb, S. B., Bergmeir, C., Bessa, R. J., Bijak, J., Boylan, J. E., et al. (2022). Forecasting: Theory and practice. *International Journal of Forecasting*, *38*(3), 705–871.

Punia, S., Nikolopoulos, K., Singh, S. P., Madaan, J. K., & Litsiou, K. (2020). Deep learning with long short-term memory networks and random forests for demand forecasting in multi-channel retail. *International Journal of Production Research*, *58*(16), 4964–4979.

Ramos, P., Santos, N., & Rebelo, R. (2015). Performance of state space and ARIMA models for consumer retail sales forecasting. *Robotics and Computer-Integrated Manufacturing*, *34*, 151–163.

Ren, S., Chan, H.-L., & Siqin, T. (2020). Demand forecasting in retail operations for fashionable products: Methods, practices, and real case study. *Annals of Operations Research*, *291*(1), 761–777.

Ribeiro, M. H. D. M., da Silva, R. G., Mariani, V. C., & dos Santos Coelho, L. (2020). Short-term forecasting covid-19 cumulative confirmed cases: Perspectives for brazil. *Chaos, Solitons & Fractals*, *135*, 109853.

Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, *2*(3), 1–21.

Schölkopf, B., Smola, A. J., Williamson, R. C., & Bartlett, P. L. (2000). New support vector algorithms. *Neural Computation*, *12*(5), 1207–1245.

Semenoglou, A.-A., Spiliotis, E., Makridakis, S., & Assimakopoulos, V. (2021). Investigating the accuracy of cross-learning time series forecasting methods. *International Journal of Forecasting*, *37*(3), 1072–1084.

Seyedan, M., & Mafakheri, F. (2020). Predictive big data analytics for supply chain demand forecasting: Methods, applications, and research opportunities. *Journal of Big Data*, *7*(1), 1–22.

Shahvaroughi Farahani, M., & Razavi Hajiagha, S. H. (2021). Forecasting stock price using integrated artificial neural network and metaheuristic algorithms compared to time series models. *Soft computing*, *25*, 8483–8513.

Silva, J. C., Figueiredo, M. C., & Braga, A. C. (2019). Demand forecasting: A case study in the food industry. In S. Misra, O. Gervasi, B. Murgante, E. Stankova, V. Korkhov, C. Torre, A. M. A. Rocha, D. Taniar, B. O. Apduhan, & E. Tarantino (Eds.), *Computational science and its applications – ICCSA 2019* (pp. 50–63). Springer.

Spiliotis, E., Makridakis, S., Semenoglou, A.-A., & Assimakopoulos, V. (2022). Comparison of statistical and machine learning methods for daily SKU demand forecasting. *Operational Research*, *22*, 3037–3061.

Syntetos, A. A., Boylan, J. E., & Croston, J. (2005). On the categorization of demand patterns. *Journal of the Operational Research Society*, *56*(5), 495–503.

Teunter, R. H., & Duncan, L. (2009). Forecasting intermittent demand: A comparative study. *Journal of the Operational Research Society*, *60*(3), 321–329.

Violi, A., Laganá, D., & Paradiso, R. (2020). The inventory routing problem under uncertainty with perishable products: An application in the agri-food supply chain. *Soft computing*, *24*, 13725–13740.

Wang, J., Li, L., Niu, D., & Tan, Z. (2012). An annual load forecasting model based on support vector regression with differential evolution algorithm. *Applied Energy*, *94*, 65–70.

Yuna, F., Erkayman, B., & Yılmaz, M. (2023). Inventory control model for intermittent demand: A comparison of metaheuristics. *Soft Computing*, *27*, 6487–6505.

Zhang, Z., & Hong, W.-C. (2021). Application of variational mode decomposition and chaotic grey wolf optimizer with support vector regression for forecasting electric loads. *Knowledge-Based Systems*, *228*, 107297.

Chapter 6

Minimizing Costs and CO_2 Emissions in a Waste Transfer Facility Location Problem[1]

Abstract

We address a bi-objective real-world facility location problem arising in the Italian waste management industry. The problem is to collect urban waste and deliver it to treatment or disposal facilities. A predicted quantity of different classes of waste must be collected from a set of sources over a predetermined time horizon. The flow must be optimized also by deciding whether and where to open additional intermediate transfer facilities among a set of candidate locations. The goals are to minimize the total costs and the CO_2 emissions. We provide a single-period mixed integer linear programming model and then extend it to a multi-period setting allowing for seasonal waste fluctuations. We apply an approximated ϵ-constraint algorithm to solve our models on two real-world case studies, obtaining approximated-but-well-structured Pareto sets of non-dominated solutions. The efficacy of the models is confirmed by further computational experiments on randomly created instances, which show that the models can be employed for analogous real-world applications.

6.1 Introduction

Waste management is a general term referring to the set of activities related to collection, transport, treatment and disposal of waste, and, in addition, control and prevention actions across the whole process. The increasing amount and complexity of waste generated by modern societies has indeed attracted substantial attention within major industrial sectors and raised major sustainability-related concern around governments, firms, and individuals. As a consequence, waste management has been recently connected to environmental issues, as stated, for example, by Tolaymat et al. (2015), who referred to waste management as the link between all the subjects involved in the waste production

[1]Caselli, G., Columbu, G., Iori, M., Magni, C. A., Oliveira, M. (2024). Minimizing costs and CO_2 emissions in a waste transfer facility location problem (under review).

network and the societal entities taking care of environmental goals. The significant environmental impact of the waste industry is well known and reduction measures have been already introduced in many systems (e.g., ReVelle, 2000) to cut the amount of greenhouse gas emissions along the process (e.g., products' recycling and salvage, collection routing optimization).

Many operational and strategic problems related to each phase of the waste management process have been studied in the literature for decades, confirming a high interest of both researchers and practitioners in this field. Operational problems refer to short-term optimization decisions such as routing and scheduling problems. In this Operations Research (OR) context, the literature on the classical Vehicle Routing Problem (VRP) applied to waste collection contexts is huge (e.g., Golden et al., 2002). Strategic problems refer to medium- and long-term design and management decisions to optimize the waste collection and treatment network, including the location of different facilities (e.g., collection points, intermediate transfer facilities, final treatment facilities, and landfills). The Facility Location Problem (FLP) is addressed in this context (e.g., Van Engeland et al., 2020).

Several works in the literature have dealt with FLPs with applications in the waste industry, focusing on minimization of costs (e.g., Lee et al., 2016). Other recent approaches take into account both costs and CO_2 emissions minimization (e.g., Mohsenizadeh et al., 2020). In this regard, we propose a bi-objective approach to a waste transfer FLP as well, with the aim of providing solutions that are valuable under both the economic and environmental perspectives.

Our work deals with a specific FLP in the context of urban waste collection, that is, the collection of urban waste from multiple sources and its transport to the treatment or disposal plants. The activity is typically managed by municipal services or by public or private corporations. Our work is inspired by the waste transfer activity carried out by Iren Ambiente SpA, an Italian multi-utility private company, from customers (sources of urban waste) to final plants, where waste is collected before treatment.

Iren Ambiente is a division of Gruppo Iren, an industrial holding company operating in the Italian market of multi-utilities. Iren Ambiente manages the operations of waste collection, treatment and disposal, designs waste treatment and disposal systems, and controls renewable energy systems in several areas of Italian regions (mainly Emilia Romagna, Piemonte, Liguria, Toscana, Lombardia, and Sardegna). The two case studies presented in this work refer to the region of Emilia Romagna and, in particular, to the districts of Reggio Emilia and Parma.

The specific problem studied in this work only considers the transfer of waste, excluding downstream and upstream processes of waste production and treatment or disposal (and the related costs and CO_2 emissions), even tough the entire waste management process is managed by the company in its assigned areas. The problem is a particular Capacitated Facility Location Problem (CFLP) (i.e., an FLP where facilities have limited capacity), which we solve by means of bi-objective Mixed Integer Linear Programming (MILP) models. First, the bi-objective single-period model is presented: it is a finite-horizon decision problem, solved under the assumption that costs and emissions generated in a period are constant over time (so that one can focus on minimization in one single period). Then, a multi-period version of the problem is formulated considering the variation of the demands (quantities of waste to be collected), costs, and emissions over the different time periods defined in our time horizon (i.e., we assume a periodic variation of some problem parameters from period to period within the time horizon), which enables us to cope with the seasonality of urban waste production, whereby regular and predictable changes in costs and emissions recur every year. The goal of our bi-objective models is to determine the optimal network in real-world scenarios, evaluating how many locations must be opened and where they must be located to minimize the costs and the CO_2 emissions (i.e., facilities' and vehicles' emissions) involved in the process. In our real-world application, the models' solutions offer a decision support system for the company. Decision support systems not only aim to enhance industrial efficiency but can also have a positive impact on environmental sustainability (Modgil et al., 2020); hence, they are considered as a means to mitigate climate change and achieve the established UN Sustainable Development Goals (SDGs). The methodology developed to solve our bi-objective model is an ϵ-constraint method, applied on random instances created by a tailored random generator as well as real-world instances from the case studies. A preliminary version of this work involving only CO_2 emissions minimization and solving a single instance was presented as Caselli et al. (2022).

In this work, we provide the following main contributions:

- We solve a bi-objective problem in which we minimize costs and CO_2 emissions. Unlike the majority of the bi-objective works including economic and environmental goals in FLPs, which, as stated by Velázquez-Martínez and Fransoo (2017), include only transport emissions in the objective function, we account for emissions from both transport decisions and location decisions (i.e., emissions from opening/operating the facilities), providing a detailed description of the computation of each of the cost and emission components;

- We integrate multi-period and bi-objective settings within an FLP, which, as stated by Nickel and Saldanha da Gama (2019), is rare in the related literature. Our multi-period model, which comprises time-independent location variables and time-dependent flow allocation variables, is in line with the literature on real-world FLP, where location variables are typically constrained by realistic assumptions on opening/closing facilities during the time horizon;

- We solve the models by a tailoring dedicated ϵ-constraint approach, which provides approximated-but-well-structured Pareto sets. In Appendix 6.D, we additionally present MILP-based and greedy versions of a tailored weighted sum method, a simpler method that can be implemented by solution makers who are interested in quickly obtaining reduced sets of solutions;

- We solve two real-world case studies and show that important cost and emissions savings can be obtained with respect to the status quo (i.e., current network with only final facilities);

- We provide further computational experiments that prove the scalability of our models, showing that larger-size instances can be solved by our methods in short computational times.

The rest of the paper is organized as follows. A summary of the related literature is given in Section 6.2. In Section 6.3, we provide a detailed description of the problem. In

Section 6.4, we present our single- and multi-period models and the proposed solution method. Section 6.5 reports the results obtained by our models on the first real-world case study and on random instances. Concluding remarks are provided in Section 6.6. For the sake of conciseness, the details on the process adopted to estimate CO2 emissions, the description of the weighted sum method, and the outcome of the extensive computational tests on the second real-world case study and on the random instances are provided in Appendices 6.A-6.D.

6.2 Literature review

In their recent survey, Van Engeland et al. (2020) reviewed the literature of the so-called waste reverse supply chain, identified as the overlapping subject between waste management and the broader reverse logistics. The latter, commonly referred as the network of activities and processes related to the flow of raw material, inventory, finished goods, and waste from the point of consumption back to the origin point, has been surveyed, among many, by Govindan et al. (2016). Several types of waste can be involved, such as wastewater (see, e.g., Salamirad et al., 2023) and urban waste, which we consider in this work. Collection, transportation, recovery, and disposal of waste are included in the waste reverse supply chain, where several problems are solved with the aim of creating value at three different levels of management decisions: (i) long-term strategic decisions of waste network design; (ii) medium-term decisions on waste quantities and capacity allocation, and (iii) short-term (operational) decisions such as routing and scheduling.

Focusing on the area of long-term strategic network design problems, Van Engeland et al. (2020) surveyed the extensive literature of the period 1995-2020, providing a classification of strategic network design problems and their combinatorial optimization solution methods based on several characteristics: single- or multi-period decisions, single- or multi-product problems, single- or multi-objective optimization, with specific constraints and different objective functions. They found that 60% of collected works deal with single-objective functions, representing cost minimization or profit maximization in around 95% of the cases. In this regard, our work is an attempt to extend the branch of the literature on multi-objective network design problems. In multi-objective models, environmental goals such as the minimization of CO_2 emissions or energy use are often included and balanced with economic ones such as the minimization of costs or maximization of profits. Saldanha da Gama (2022) dedicated a specific section of their recent survey on facility location in logistics and transportation to green logistics and, more specifically, to green facility location problems integrating CO_2 emissions considerations, mainly as additional objective minimizing the amount of carbon emissions, as we do in our work. Another recent work by Waltho et al. (2019) deals with green supply chain network problems where carbon emissions are integrated from a financial point of view based on the so-called carbon policies (carbon cap, carbon offset, cap-and-trade, and carbon tax). Amin Chaabane and Paquet (2011) introduced a cap-and-trade system and formulated a bi-objective cost-emission minimization supply chain network design problem using an MILP model. They applied the classical ϵ-constraint and goal programming methods and introduced tailored constraints for the CO_2 emissions regulation under the real-world cap-and-trade system of a Canadian firm operating in the steel industry. In this scenario, the company could buy and sell carbon credits on the market. A third approach to introduce environmental goals within network design problems involves assigning a monetary value of CO_2 emissions and minimizing the weighted sum of economic and environmental costs. Colicchia et al. (2016) proposed an ILP model with an aggregate objective function, where a price is assigned to CO_2 emissions and a weight is assigned to costs and CO_2 emissions. The authors applied their model to the real-world case study of an Italian distributor of chocolate products in the perishable food industry, and highlighted the importance of addressing more industrial case studies across various sectors to further close the gap between theory and practice. In our research, we employ both the multi-objective and aggregated approaches as our case study does not fall under any specific carbon policies. Sometimes, the social impact is also included in multi-objective problems (see, e.g., Govindan et al., 2016 and Harijani et al., 2017).

In the OR literature, the strategic problems defined above are classified under the broad category of FLPs; they have been of great interest since the 1960s. In their book, Laporte et al. (2019) defined the FLP as the problem of determining the best location for one or multiple facilities or equipment in order to cater to a set of demand points. As reported by Verter (2011), the classical FLP has been extended in a number of ways: (i) increasing the number of products, from single- to multi-commodity FLPs (see, e.g., Liu et al., 2021, who studied a complex multi-commodity CFLP involving sustainability concerns); (ii) increasing the number of facility echelons, that is, the types of facility to locate (e.g., Gendron and Semet, 2009); (iii) increasing the number of time periods included in the model, defining dynamic FLPs where the facility location is determined at each period so as to minimize the total cost over time (e.g., Nickel and Saldanha da Gama, 2019); and (iv) incorporating possible scale economies in the cost function (e.g., Wu et al., 2006) and uncertainties (e.g., Correia and Saldanha da Gama, 2019).

As concerns FLPs in the waste management industry, Adeleke and Olukanni (2020) surveyed models and solution algorithms published between 2006 and 2020 and adapted to cope with several optimization problems. These problems are usually formulated as MILP models, but then solved in practice by means of heuristic algorithms able to find near-optimal solutions in a limited time. In the following, we mention some case studies in the urban waste management contexts. Ghiani et al. (2012) studied a bin allocation problem in Italy where the aim is to minimize the total number of activated waste collection sites. The problem was solved by means of an MILP model and a constructive heuristic. Lee et al. (2016) proposed several mathematical models for the waste management system of Hong Kong, in which they minimize the total cost for the municipal solid waste management system. Gambella et al. (2019) studied a facility location and waste flow allocation problem. They developed a stochastic programming model that was applied to solve a real-world Italian case study.

A relevant part of the current literature on waste transfer FLPs includes environmental concerns in the problem statement and in the model formulation. A summary of the main concepts and models for the so-called Green FLP is included by Velázquez-Martínez and Fransoo (2017). The focus of their work is on the transportation performance of firms in terms of both costs and emissions, which is strongly determined by the design of the network. In the models under review, the main sources of CO_2 emissions associated with the location of facilities derive from both mobile sources (transportation) and stationary sources (production, storage, and handling). Environmental qualitative and quantitative evaluations have been combined with an MILP model and multi-criteria methods by Vaillancourt and Waaub (2002) to solve waste FLPs similar to our problem: valuable results have been obtained on a case study of the city of Montreal, Canada. The massive work in OR on location problems with applications in urban services, and more specifically in solid waste management, has been surveyed by Ghiani et al. (2014) and Farahani et al. (2019), among many.

We now survey the most recent and relevant literature on multi-objective FLPs with environmental goals with applications in the waste industry, which is our main interest. Erkut et al. (2008) solved a case study for a waste FLP in a northern region of Greece, applying the so-called lexicographic minimax approach (from game theory) to an MILP model with five economic and environmental objectives. Coutinho-Rodrigues et al. (2012) proposed a bi-objective MILP model for an urban facility problem minimizing investment costs and dissatisfaction, solved by means of the ϵ-constraint method. Eiselt and Marianov (2014) minimized a bi-objective cost-pollution function in a real-world Chilean landfills' location problem, solved by the weighting sum and ϵ-constraint methods. Yu and Solvang (2017) proposed an MILP model for a three-stage waste management system where three objectives are included in the objective functions: (i) the overall costs, (ii) greenhouse gas emissions, and (iii) the impact of toxic gases on the population living nearby. Rabbani et al. (2018) solved a bi-objective location-routing problem for the minimization of costs and environmental impact (including soil pollution, noise pollution, and total emissions) by means of an evolutionary algorithm. We mention the works by Asefi et al. (2019) and Tirkolaee et al. (2023), which also deal with location-routing problems for sustainable waste collection. Mohsenizadeh et al. (2020) solved a bi-objective cost-pollution transfer station location problem for municipal solid waste management in Ankara. Darmian et al. (2020) proposed an MILP model for the waste districting-location problem minimizing costs along with a measure of destructive environmental consequences and a measure of social dissatisfaction. The model was applied to two case studies from Iran and on randomly generated instances by means of the ϵ-constraint (for small-size instances) and a local search heuristic algorithm (for large-size instances).

As stressed by Velázquez-Martínez and Fransoo (2017), not many companies have implemented facility locations strategies to reduce their environmental impact, especially as primary goal, although a considerable amount of theoretical work is already available in the literature together with a few real-world applications presented above. In this respect, our work is an attempt to provide an example of application of green FLPs in practice and contributes to the relevant literature of multi-objective optimization of urban waste FLPs, where economic and environmental goals are modeled to guide decision makers towards more sustainable decisions in a two-dimensional view.

6.3 Problem statement

In our CFLP, we are given a set I of customers (i.e., waste sources) to be visited over a given time horizon. The total estimated amount of waste produced by each customer must be collected and delivered to final treatment facilities, either directly or passing by one or more intermediate transfer facilities. We call J the set of all (existing and candidate) facility locations, and $\bar{J}$ the subset of all intermediate transfer facility locations. Therefore, $J \setminus \bar{J}$ is the subset of final treatment facility locations. In addition, we call J_e the subset of existing (intermediate and final) facilities. We denote by H the set of waste types, such as paper, plastics, and glass (i.e., we deal with a multi-commodity CFLP).

Let q_{ih} indicate the total estimated quantity of waste type h to be collected from customer i over the considered time horizon (for which there is a corresponding number of trips). Demand aggregation is commonly applied in the literature of FLPs as shown, among many, by Coutinho-Rodrigues et al. (2012) and Eiselt and Marianov (2014), to reduce the high dimensionality and computational time of the problems. The number of trips is predetermined by the municipality (e.g., paper waste is collected once or twice per week). For each facility location j, we define Q_j as the overall waste capacity and Q_{jh} as the capacity for waste type h, where $Q_j \leq \sum_{h \in H} Q_{jh}$. For example, if a location j has $Q_j = 100$ tons and $Q_{j1} = 70$ tons, $Q_{j2} = 60$ tons for $H = \{1, 2\}$, then we can dedicate half capacity to each waste (i.e., 50 tons each) or we can accept an unbalanced solution without exceeding the Q_{jh} values (e.g., 70 tons and 30 tons dedicated to waste type $h = 1$ and $h = 2$ respectively).

For cost minimization, we introduce two components of transport costs and two components of usage costs. Parameters C_{ijh} and C'_{jkh} estimate transport costs for waste type $h \in H$ from customer $i \in I$ to facility $j \in J$ (called "first-level transport") and from intermediate facility $j \in \bar{J}$ to facility $k \in J$ (called "second-level transport"), respectively. Parameters r_{jh} and G_j represent, respectively, variable and fixed cost components of usage costs related to facility j: the former vary in proportion to the amount of waste type h treated by facility j, the latter are independent of the amount of waste treated by the facility.

An analogous set of parameters is introduced for CO_2 emissions minimization. Transport emissions are accounted for by parameters e_{ijh}, for $i \in I, j \in J, h \in H$, and e'_{jkh}, for $j \in \bar{J}, k \in J, h \in H$. The former estimates transport emissions for waste type h from customer i to facility j, while the latter estimates transport emissions from intermediate facility j to (intermediate or final) facility k. The variable and fixed components of emissions generated by operating a new facility $j \in J$ are denoted as p_{jh} and F_j, respectively. The mathematical notation is summarized in Table 6.1.

The goal is to find the optimal waste transfer network (i.e., to select the intermediate transfer locations and final treatment locations to open and the waste flow across the network) that minimizes the total cost and CO_2 emissions involved in the process of collection, transfer, and delivery of waste to final treatment locations over the considered time horizon.

6.4 Mathematical models

In this section, we present the single-period bi-objective model that we have developed to solve our multi-commodity CFLP (Section 6.4.1) and its multi-period extension (Section 6.4.2). Then, we present an approximated ϵ-constraint method (Section 6.4.3).

Table 6.1: Mathematical notation.

Notation	Definition
I	Set of customers
J	Set of facility locations
$\bar{J}$	Set of intermediate facility locations
$J \setminus \bar{J}$	Set of final facility locations
J_e	Set of existing facility locations
H	Set of waste types
q_{ih}	Quantity of waste type h produced by customer i
Q_j	Overall waste capacity of location j
Q_{jh}	Capacity of location j for waste type h
C_{ijh}	Transport costs for waste type h from customer i to location j
C'_{jhk}	Transport costs for waste type h from intermediate location j to location k
r_{jh}	Variable usage costs for facility j and waste type h
G_j	Fixed usage costs for facility j
e_{ijh}	Transport CO_2 emissions for waste type h from customer i to location j
e'_{jhk}	Transport CO_2 emissions for waste type h from intermediate location j to location k
p_{jh}	Variable CO_2 emissions component for operating facility j and waste type h
F_j	Fixed CO_2 emissions component for operating facility j

6.4.1 Single-period model

To formulate our MILP model, we additionally introduce a set of three-index variables x_{ijh} that indicate the fraction of waste of type h collected at customer $i \in I$ and transferred to location $j \in J$ in the overall considered period. The variable is continuous, implying that each customer demand can be fulfilled by one or more facilities. A further set of continuous variables f_{jkh} represents the flow of waste of type h transferred from intermediate location $j \in \bar{J}$ to intermediate/final location $k \in J, j \neq k$. Finally, binary variables y_j take the value 1 if location $j \in J$ is open, and 0 otherwise. Equations (6.1) and (6.2) represent the cost and CO_2 emissions minimization problems, respectively:

$$\begin{aligned} \min\ Z_C = &\sum_{i \in I} \sum_{j \in J} \sum_{h \in H} C_{ijh} x_{ijh} + \sum_{j \in \bar{J}} \sum_{\substack{k \in J \\ j \neq k}} \sum_{h \in H} C'_{jkh} f_{jkh} + \\ &\sum_{k \in J} \sum_{h \in H} r_{kh} (\sum_{i \in I} q_{ih} x_{ikh} + \sum_{\substack{j \in \bar{J} \\ j \neq k}} f_{jkh}) + \sum_{j \in J} G_j y_j \end{aligned} \tag{6.1}$$

$$\begin{aligned} \min\ Z_E = &\sum_{i \in I} \sum_{j \in J} \sum_{h \in H} e_{ijh} x_{ijh} + \sum_{j \in \bar{J}} \sum_{\substack{k \in J \\ j \neq k}} \sum_{h \in H} e'_{jkh} f_{jkh} + \\ &\sum_{k \in J} \sum_{h \in H} p_{kh} (\sum_{i \in I} q_{ih} x_{ikh} + \sum_{\substack{j \in \bar{J} \\ j \neq k}} f_{jkh}) + \sum_{j \in J} F_j y_j \end{aligned} \tag{6.2}$$

The two objective functions (6.1) and (6.2) minimize the total amount of costs and CO_2 emissions involved in the process over the considered time horizon, each of them with the following structure: the first two terms count the transport costs/emissions for customer-facility and facility-facility trips respectively, while the third and fourth terms consider the usage costs/emissions generated by operating new facilities, addressing variable and fixed components of costs/CO_2 emissions respectively. In our MILP model, we include the following sets of constraints:

$$\sum_{j \in J} x_{ijh} = 1 \qquad \forall i \in I, \forall h \in H \tag{6.3}$$

$$\sum_{i \in I} q_{ih} x_{ikh} + \sum_{\substack{j \in \bar{J} \\ j \neq k}} f_{jkh} \leq Q_{kh} y_k \qquad \forall k \in J, \forall h \in H \tag{6.4}$$

$$\sum_{i \in I} \sum_{h \in H} q_{ih} x_{ikh} + \sum_{\substack{j \in \bar{J} \\ j \neq k}} \sum_{h \in H} f_{jkh} \leq Q_k y_k \qquad \forall k \in J \tag{6.5}$$

$$\sum_{i \in I} q_{ih} x_{ikh} + \sum_{\substack{j \in \bar{J} \\ j \neq k}} f_{jkh} - \sum_{\substack{l \in J \\ l \neq k}} f_{klh} = 0 \qquad \forall k \in \bar{J}, \forall h \in H \tag{6.6}$$

$$y_j = 1 \qquad \forall j \in J_e \tag{6.7}$$

$$0 \leq x_{ijh} \leq 1 \qquad \forall i \in I, \forall j \in J, \forall h \in H \tag{6.8}$$

$$f_{jkh} \geq 0 \qquad \forall j \in \bar{J}, \forall k \in J, j \neq k, \forall h \in H \tag{6.9}$$

$$y_j \in \{0, 1\} \qquad \forall j \in J \tag{6.10}$$

Constraint set (6.3) ensures that each waste demand is fulfilled by one or more facilities sharing its entire demand over the considered period. Constraints (6.4) and (6.5) represent the waste-specific and overall capacity constraints for each facility, respectively. Constraints (6.6) guarantee the conservation of waste-specific flow at each intermediate facility (i.e., the total incoming flow from customers and other intermediate facilities is equal to the flow going out to other facilities). Constraint set (6.7) imposes that existing facilities are kept open by the model solution. Constraints (6.8)-(6.10) give the domain of the variables.

6.4.2 Extension to the multi-period model

As reported by Nickel and Saldanha da Gama (2019), "in order to capture predictable variations in the parameters of a facility location problem, we often have to consider a so-called dynamic or time-dependent model." By this quote, the authors introduced their discussion on multi-period facility location models stressing their relevance for realistic applications of this class of problems. By following this hint, we extend our model from a single-period to a multi-period version whereby we divide the time horizon into a predetermined number of shorter periods. Let T be the set of such periods. We divide one year into months (i.e, $T = \{1, 2, ..., 12\}$). In this way, we cope with the seasonality of urban waste production, where costs, emissions, and especially waste collection demands fluctuate monthly. Several factors impact the seasonality of urban waste production in our real-world problem: (i) tourism (e.g., in the non-touristic areas, waste decreases in the summer months, especially in August, and partly also in December and April due to several holidays; on the contrary, in the touristic areas, the quantity of urban waste increases from April to August); (ii) holidays (e.g., the overall quantity of waste increases during Christmas holidays and other local celebration days in some specific municipalities); (iii) calendar (e.g,. February is typically the month with the smallest waste quantities); and (iv) weather and population's habits (e.g., production of plastic and organic waste, as well as plastic bottles, increases during the summer, due to the higher consumption of fruits, vegetables, and beverages).

We define time-dependent (i) demands q_{iht}, (ii) customer-to-facility transport costs C_{ijht}, and (iii) customer-to-facility transport CO_2 emissions e_{ijht}, with $t \in T$. The three-index flow-related sets of variables take one additional index: x_{ijht} indicates the fraction of waste of type h transferred from customer i to location j in period t, while f_{jkht} is the flow of waste of type h transferred from intermediate location j to location k in period t. We keep the set of location decision variables y_j fixed over the time horizon (i.e., time-independent), as they are in model (6.3)-(6.10). In the multi-period model, the set of open locations is defined by considering the entire set of periods (i.e., 12 months in our instances), given the costs, the emissions, and the capacity constraints. Therefore, the model decides whether and which facilities to open at the beginning of the time horizon. Such facilities are kept open until the end of the time horizon. The model can change the flow allocation variables from one period to another, following the seasonal fluctuations, by means of the time-dependent flow-related decision variables x_{ijht} and f_{jkht}.

As reported by Nickel and Saldanha da Gama (2019), in realistic settings, facilities are frequently required to remain operational from the installation until the end of the planning horizon; facilities are rarely allowed to be partially closed or reopened (i.e., to close and reopen a part of the facility) throughout the planning horizon. Such and other situations are typically modeled by additional constraints on time-dependent location decision variables, or, as in our case, by directly keeping time-independent facility location variables. The resulting bi-objective model for the multi-period multi-commodity CFLP has the following objective functions:

$$\begin{aligned} \min\ Z_C^m = & \sum_{i\in I}\sum_{j\in J}\sum_{h\in H}\sum_{t\in T} C_{ijht}x_{ijht} + \sum_{j\in \bar{J}}\sum_{\substack{k\in J\\ j\neq k}}\sum_{h\in H}\sum_{t\in T} C'_{jkh}f_{jkht} + \\ & \sum_{k\in J}\sum_{h\in H} r_{kh}\big(\sum_{i\in I}\sum_{t\in T} q_{iht}x_{ijht} + \sum_{\substack{j\in \bar{J}\\ j\neq k}}\sum_{t\in T} f_{jkht}\big) + \sum_{j\in J} G_j y_j \end{aligned} \tag{6.11}$$

$$\begin{aligned} \min\ Z_E^m = & \sum_{i\in I}\sum_{j\in J}\sum_{h\in H}\sum_{t\in T} e_{ijht}x_{ijht} + \sum_{j\in \bar{J}}\sum_{\substack{k\in J\\ j\neq k}}\sum_{h\in H}\sum_{t\in T} e'_{jkh}f_{jkht} + \\ & \sum_{k\in J}\sum_{h\in H} p_{kh}\big(\sum_{i\in I}\sum_{t\in T} q_{iht}x_{ijht} + \sum_{\substack{j\in \bar{J}\\ j\neq k}}\sum_{t\in T} f_{jkht}\big) + \sum_{j\in J} F_j y_j \end{aligned} \tag{6.12}$$

and is subject to (6.7), (6.10), and to the following sets of constraints:

$$\sum_{j\in J} x_{ijht} = 1 \qquad \forall i \in I, \forall h \in H, \forall t \in T \tag{6.13}$$

$$\sum_{i\in I}\sum_{t\in T} q_{iht}x_{ikht} + \sum_{\substack{j\in \bar{J}\\ j\neq k}}\sum_{t\in T} f_{jkht} \leq Q_{kh}y_k \qquad \forall k \in J, \forall h \in H \tag{6.14}$$

$$\sum_{i\in I}\sum_{h\in H}\sum_{t\in T} q_{iht}x_{ikht} + \sum_{\substack{j\in \bar{J}\\ j\neq k}}\sum_{h\in H}\sum_{t\in T} f_{jkht} \leq Q_k y_k \qquad \forall k \in J \tag{6.15}$$

$$\sum_{i \in I} q_{iht} x_{ikht} + \sum_{\substack{j \in \bar{J} \\ j \neq k}} f_{jkht} - \sum_{\substack{l \in J \\ l \neq k}} f_{klht} = 0 \qquad \forall k \in \bar{J}, \forall h \in H, \forall t \in T \tag{6.16}$$

$$0 \leq x_{ijht} \leq 1 \qquad \forall i \in I, \forall j \in J, \forall h \in H, \forall t \in T \tag{6.17}$$

$$f_{jkht} \geq 0 \qquad \forall j \in \bar{J}, \forall k \in J, j \neq k, \forall h \in H, \forall t \in T \tag{6.18}$$

Equations (6.11)-(6.12) represent the two cost and CO_2 emissions minimization objective functions analogous to (6.1)-(6.2). Constraint set (6.13) ensures that each waste demand is fulfilled by one or more facilities sharing its entire demand over each time period. Constraints (6.14) and (6.15) define, respectively, waste-specific and overall capacity limit for each facility on the overall time horizon. Constraint set (6.16) guarantees the conservation of waste-specific flow for each intermediate facility on each time period. Constraints (6.17)-(6.18) give the domain of the new period-dependent variables.

6.4.3 ϵ-constraint method

To solve our bi-objective models, we use an approximated version of the ϵ-constraint method. Such a method, introduced by Soland (1979) and applied with success to several multi-objective problems in the waste management literature (see, e.g., Coutinho-Rodrigues et al., 2012 and Eiselt and Marianov, 2014), iteratively invokes a single-objective model with an additional constraint limiting the value of the second objective. In our work, because of the large-size of the instances, we are not interested in finding the complete optimal Pareto front, i.e., the complete set of non-dominated solutions, but only in finding a set of well-dispersed solutions along the Pareto front.

We apply the same approximated ϵ-constraint method to both the single- and the multi-period model, as follows. First, the two ideal solutions that optimize the two objectives individually are computed and considered as the "payoff table" (see Coutinho-Rodrigues et al., 2012). For each of the two ideal solutions (i.e., minimum cost and minimum emission solutions), we check if they are dominated (i.e., if it exists a solution with same cost but lower emissions and same emissions but lower cost, respectively). We check for non-dominated solutions in the same way as described in Appendix 6.D for the weighted sum method. If a non-dominated solution is found within a specific time limit (which, in our case, is equal to the time limit on each single point optimization), we update the latter to be the new extreme point of the Pareto front and we obtain a new payoff table accordingly. Otherwise, we keep the original extreme points and proceed with the next step of the algorithm. Then, a single-objective model with only the cost minimization function is solved iteratively, with an additional constraint imposing at each iteration i of the algorithm a limit to the maximum value of CO_2 emissions indicated by E_i and being equal to the value of CO_2 emissions obtained at iteration $i-1$. The resulting single-period model at iteration i is thus min Z_C as in (6.1) subject to (6.3)-(6.10) and:

$$\begin{aligned}\sum_{i \in I} \sum_{j \in J} \sum_{h \in H} e_{ijh} x_{ijh} + \sum_{j \in J_1} \sum_{\substack{k \in J \\ j \neq k}} \sum_{h \in H} e'_{jkh} f_{jkh} + \\ \sum_{k \in J} \sum_{h \in H} p_{kh} (\sum_{i \in I} q_{ih} x_{ih} + \sum_{\substack{j \in J_1 \\ j \neq k}} f_{jkh}) + \sum_{j \in J} F_j y_j \leq E_i - \epsilon\end{aligned} \tag{6.19}$$

where ϵ is a tolerance parameter computed as $\epsilon = (E_1 - E_n)/\delta$. Values E_1 and E_n indicate the total CO_2 emissions obtained by solving the single-objective model minimizing only costs and CO_2 emissions respectively (i.e., $E_1 - E_n$ is our payoff table for CO_2 emissions), while δ is a parameter chosen arbitrarily such that the resulting Pareto set has at most $\delta + 1$ solutions.

Similarly, the multi-period model is derived as min Z_C^m as in (6.11) subject to (6.7), (6.10), (6.13)-(6.18), and:

$$\sum_{i\in I}\sum_{j\in J}\sum_{h\in H}\sum_{t\in T} e_{ijht}x_{ijht} + \sum_{j\in \bar{J}}\sum_{\substack{k\in J\\ j\neq k}}\sum_{h\in H}\sum_{t\in T} e'_{jkh}f_{jkht} + \sum_{k\in J}\sum_{h\in H} p_{kh}\Big(\sum_{i\in I}\sum_{t\in T} q_{iht}x_{ijht} + \sum_{\substack{j\in \bar{J}\\ j\neq k}}\sum_{t\in T} f_{jkht}\Big) + \sum_{j\in J} F_j y_j \leq E_i - \epsilon \tag{6.20}$$

with the same definition for the parameter ϵ. The method stops once an infeasible solution is found (i.e., once the additional constraint on CO_2 emissions becomes too restrictive) or the global time limit is reached. We include a post-processing procedure to eliminate the dominated solutions from the set of generated solutions.

In the complete set of efficient solutions, there might be multiple equivalent solutions, that is, solutions that have the same value of both objectives. In our problem, since we have continuous variables, we could have infinitely many equivalent solutions. In our ϵ-constraint algorithm, we are thus only interested in finding an approximated Pareto front. Therefore, we content ourselves with an approximation of the so-called minimal complete set of solutions, which contains a unique solution for each set of equivalent ones. The search for minimal complete sets is standard in the multi-objective literature, starting from the seminal work of Hansen (1980).

6.5 Computational experiments

In this section, we study the performance of our bi-objective approaches on the real case in the Reggio Emilia (Section 6.5.1) district. Then, we present the outcome of further computational experiments on larger-size randomly-created instances (Section 6.5.2). Our models have been coded in the Python language and solved using the Gurobipy API on a machine with an Intel(R) Xeon(R) Gold 6252N CPU @ 2.30GHz processor, 16 GB of RAM memory, running under a Windows 10 operational system and using four threads of the processor. For every instance, a time limit of 1200 seconds was imposed to the solver to find each individual solution of the Pareto set, and a global time limit of three hours is set for the overall ϵ-constraint method.

Costs and emissions are expressed in million euros and kilotons of CO_2, respectively, and the time horizon of every instance is one year (with 12 monthly periods for the multi-period model). In the computation of CO_2 emissions, we consider fuel consumption and gross weight of various diesel Euro 6 vehicles (ranging between 3 and 15 liters per hour, and 3.5 and 44 tons, respectively). Fuel consumption is converted in CO_2 emissions with the conversion factor (2.63 kg CO_2 per liter of diesel fuel) given by DEFRA (2019), as commonly done in the green facility location literature (see, e.g., Harris et al., 2014). A detailed description of the cost and CO_2 emissions parameters evaluation is provided in Appendix 6.A. Further details on the computational tests concerning the second real-world case study, the random instances, and the weighted sum approach are provided in Appendices 6.B-6.D.

6.5.1 Case study

The case study addresses the district of Reggio Emilia. This district has 37 sources, one existing final treatment facility (located outside of the north-west border of the Reggio Emilia district) where all the waste flow is conveyed from sources, and, currently, there is no intermediate transfer facility. There are 32 candidate intermediate transfer facilities, defined by eight capacity configurations for each of the two locations (Mancasale and Castelnuovo Monti) and each of the two considered waste types (paper and plastic). There is no candidate final treatment facility.

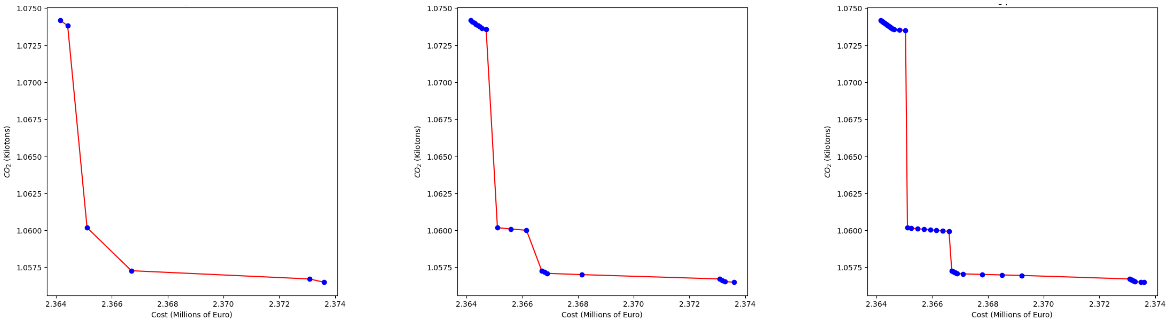
A total quantity of nearly 330 kilotons of urban waste is collected every year to serve approximately 460,000 inhabitants. The waste collection activity of Iren Ambiente consists in moving vehicles from the depots, collecting waste from the sources and transporting them to the final treatment or disposal facility either directly or passing by intermediate transfer facilities, and bringing back the vehicles to the original depots.

The time horizon is one year: the company wants to decide whether and where to open intermediate facilities and determine the network of waste flows associated with minimum costs and minimum emissions. To estimate the annual parameters, we used the real data on paper and plastic waste collected in the province of Reggio Emilia between October 2019 and October 2020. As confirmed by a preliminary analysis, the emergency situation due to Covid-19 pandemics did not affect the urban waste industry significantly, nor was Iren Ambiente's business specifically affected; therefore, those data may be used in our model as reliable estimates for the future annual costs and emissions. In the following, we present the results of our approximated ϵ-constraint method on the case study of Reggio Emilia, providing the details on solution values of objectives and variables for the single-period and multi-period models.

Single-period model

The single-period model (6.1)-(6.10) for Reggio Emilia instance has 4,587 variables, 240 constraints and 25,624 non-zero coefficients. The resulting approximated Pareto sets of solutions obtained by applying the approximated ϵ-constraint method of Section 6.4.3 for three different values $\delta = 50, 200, 500$ are shown in Figure 6.1. All the methods were solved within 10 seconds of CPU time (9.96 seconds for the longest one with $\delta = 500$). With $\delta = 50$ we found six solutions, five of which are extreme points of the front. We obtained 19 and 49 (optimal) solutions for $\delta = 200$ and $\delta = 500$ with six and nine extreme points, respectively. From the fronts in Figure 6.1, we observe that there are four groups of points close to each other (the larger δ, the clearer the distinction of such groups) that represent four groups of similar solutions in terms of selected facility locations with different waste flows. This is in line with the common interpretation of facility location cost/emission Pareto fronts and the so-called trade-off solutions for facility location and flow allocation decisions (see, e.g., Harris et al., 2014).

We now analyze the two extreme solutions: minimum cost and minimum CO_2 emission solutions. The overall minimum cost solution value is equal to 2.36 million



(a) $\delta = 50$ (b) $\delta = 200$ (c) $\delta = 500$

Figure 6.1: Pareto fronts of the single-period model for Reggio Emilia instance.

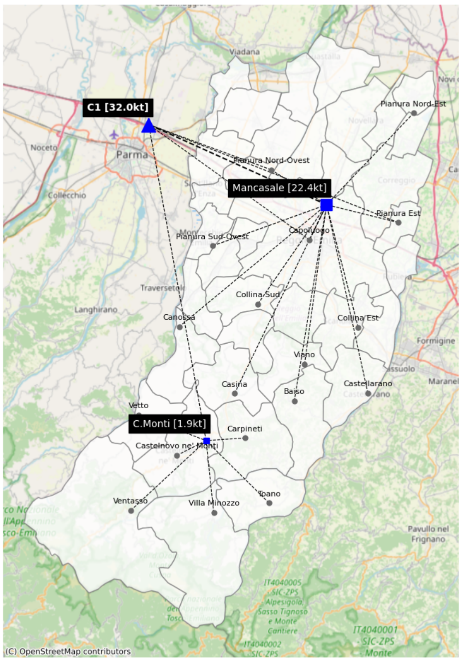
euros of costs with 1.07 kilotons of CO_2 emissions. The overall minimum emission solution value is equal to 1.06 kilotons of CO_2 with 2.37 million euros of costs.

We report the graphical representation of the waste flow of the minimum cost and minimum emission solutions in Figure 6.2, using OpenStreetMaps to get the map of the selected Italian areas. In particular, Figures 6.2a-6.2b refer to paper waste flow, while Figures 6.2c-6.2d refer to plastic waste flow. Final and intermediate facilities are indicated as triangular and squared symbols, respectively, each sized proportionally to the capacity of the facility represented, while dots represent the source points. Dashed lines indicate waste flow between source points and facilities or between two facilities. The line thickness varies in proportion to the quantity of waste flow. Names of facility locations are boxed (with text in bold for final facilities).

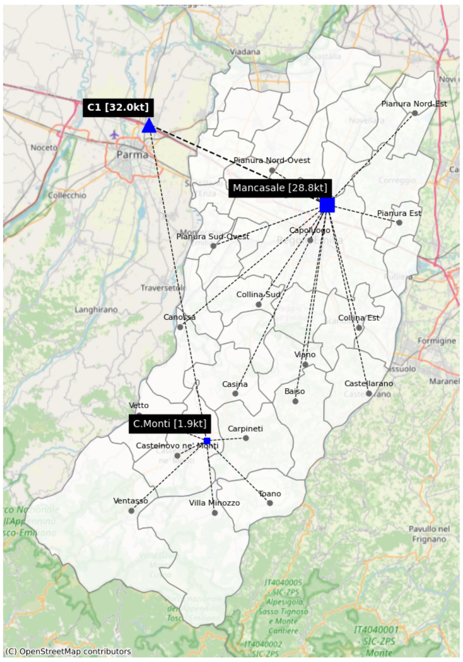
In the status quo of this small-size case study, no intermediate facilities exist and only one final facility exists (indicated as "C1" in Figure 6.2). In the minimum cost and minimum emission model solutions, two intermediate facilities are opened in both cases and for both paper and plastic in Mancasale and C. Monti, with significant reduction of both costs[2] and CO_2 emissions (equal to 24% and 25% in the minimum cost and minimum emission solutions, respectively, considering the aggregation of paper and plastic waste) with respect to the status quo, clearly showing the benefit of intermediate transfer flow of waste.

In the two solutions, the facility in C. Monti has the same size in terms of paper and plastic waste capacity, while a different size is selected for Mancasale: when minimizing costs, more flow is conveyed directly from waste sources to the final facility, while a bigger intermediate facility and more intermediate waste flow appear when minimizing CO_2 emissions both for paper and plastic. This can be explained by the type of waste collection system performed for first-level transports in the Reggio Emilia district: it consists in the street collection with low labor cost (less than 50% of the total costs) but high CO_2 emissions. A single operator controls the automatic collection operation from the cab using cameras, without getting out of the vehicle, while an automatic side loader system empties the waste containers. Heavy vehicles are used in this type of collection. Therefore, the minimum emission solution favors the opening of larger capacity intermediate facilities to optimize a larger number of first-level transports of shorter distance, while the minimum cost solution favors more direct first-level

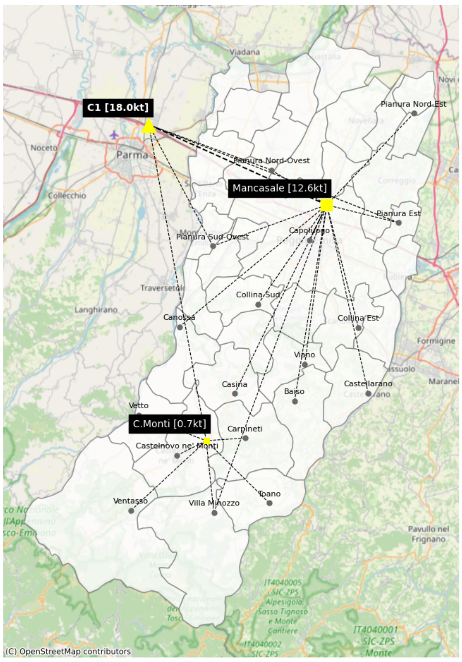
[2]Upon specific request by Iren Ambiente, we may not disclose the data on cost reduction.



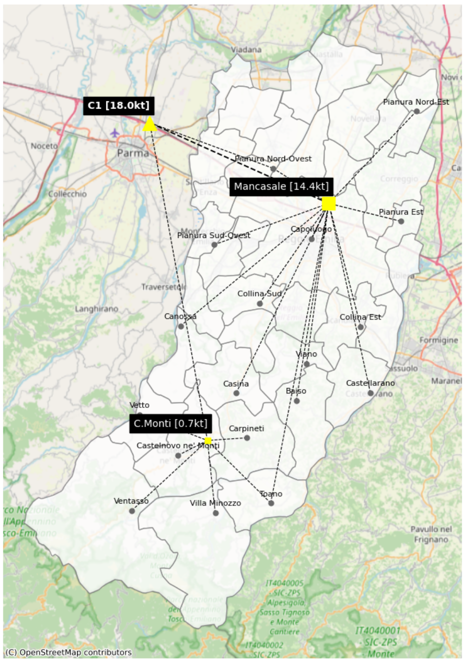
(a) Minimum cost paper flow.



(b) Minimum emission paper flow.



(c) Minimum cost plastic flow.



(d) Minimum emission plastic flow.

Figure 6.2: Flow representation of minimum cost and minimum emission solutions for Reggio Emilia case study (paper and plastic waste): triangles and squares represent final and intermediate facilities with shape size proportional to facility capacity, dots indicate the source points, and dashed lines represent the flow with thickness proportional to the amount of flow.

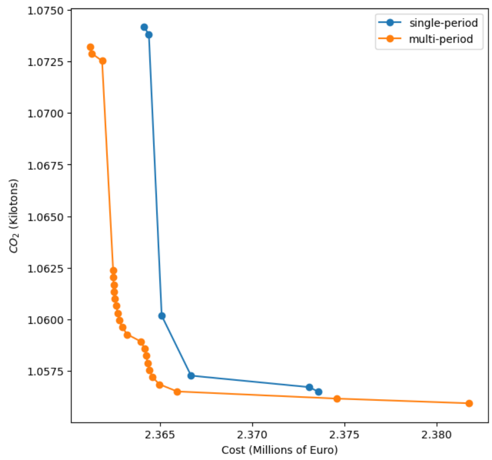


Figure 6.3: Comparison of the generated Pareto fronts with $\delta = 50$ for Reggio Emilia case study: single-period model in blue and multi-period model in orange.

transports to the final facilities (and smaller intermediate facilities) with respect to the high-cost opening of larger intermediate facilities.

In contrast to many studies that use different structures for cost and emission functions as reported by Velázquez-Martínez and Fransoo (2017), in our problem we use similar structures for the two. Therefore, the model tends to minimize both costs and emissions when deciding where to locate facilities. The same location decisions appear in the two extreme solutions since the two objectives are not divergent. However, the model gives rise to different flow allocation decisions (i.e., size of facilities and waste flow allocations), showing different solution behaviors and different total values of costs and emissions.

Multi-period model

In the real context of the considered problem, waste is collected day by day but aggregation is required to have an optimized solution that is also applicable in practice. For instance, it would not be possible for the company to change the location and allocation decisions daily or weekly. Location decisions must be taken yearly, but flow allocations can vary, for instance, month by month. Such a setting is represented by the multi-period model (6.11)-(6.18) presented in Section 6.4.2. Iren Ambiente provided us with real data on the monthly demand of waste to be collected over the same year, denoted by q_{iht}. Monthly transport costs and emissions from sources to facilities, denoted by parameters C_{ijht} and e_{ijht} respectively, were calculated accordingly, using a flat distribution as assumed by the company. We solved Reggio Emilia instance with monthly demand by means of model (6.11)-(6.18), where location decisions are fixed and flow allocation decision are taken for every period, while satisfying monthly demand constraints for sources and yearly capacity constraints for facilities.

Figure 6.3 shows the two Pareto fronts for the single-period and the multi-period models with $\delta = 50$. We obtained better Pareto fronts moving from the single-period model to the multi-period model, finding solutions with lower costs and emissions.

To compare different Pareto fronts, we applied the hypervolume indicator (see, e.g., Cao et al., 2015 and Guerreiro et al., 2021). The hypervolume indicator is a popular measure of an approximated Pareto front's quality, used to evaluate and compare different Pareto fronts obtained, for instance, by different algorithms. Computed as the volume of the bounded space by the points of the Pareto front and a user-defined reference point in the criterion space, the hypervolume metric is known as simultaneously taking into account proximity, diversity and spread of the points of the Pareto front. However, the choice of the reference point is crucial for the correct application of the method. In our work, we applied the classical hypervolume metric with the nadir point (i.e., the point with the worst value of each objective).

We compare the two fronts of single-period and multi-period models represented in Figure 6.3 by means of the hypervolume indicator. The multi-period model gains 19.27% of hypervolume over the single-period model. Moreover, larger sets of solutions are generated by the multi-period model (23 solutions against 6 in the single-period model). Despite the longer CPU time required to solve the instance (94 seconds against 2 seconds with $\delta = 50$), the multi-period model brings a significant improvement in terms of cost and emission savings.

For the sake of conciseness, the computational results from the second case study in the Parma district are provided in Appendix 6.B. These results confirm our findings from the Reggio Emilia case study.

6.5.2 Experiment on random instances

We built a set of random instances for our CFLP, based on the real-world case studies. Instances are available at https://github.com/regor-unimore/Cost-and-CO2-minimization-in-Facility-Location. The aim is to test the scalability of our models and to show how they can be used to solve a wide range of similar applications. Details on our random generator are given in Appendix 6.C. We obtained a set of 27 random instances for the single-period model (and the corresponding set for multi-period model) divided in three groups with respect to the number of sources (i.e., small, medium, and large instances with $N = 50$, $N = 200$, and $N = 500$, respectively) as shown in Table 6.2, having nine instances each. All instances were solved for $\delta = 10$.

Table 6.2: The set of random instances.

N	M	W	Instances
50	[5, 10, 25]	[1, 3, 5]	1 - 9
200	[20, 40, 100]	[1, 3, 5]	10 - 18
400	[40, 80, 200]	[1, 3, 5]	19 - 27

Tables 6.3-6.4 report the average results obtained by the ϵ-constraint method applied to the single-period and multi-period models on the three groups of random instances. The second and third columns count the average number of solutions found and the average number of optimal ones, respectively. Column 4 indicates the average number of extreme points on the Pareto front. Columns 5 and 6 report the average building and optimization times expressed in seconds (computed considering 3 hours when the global time limit is reached). Finally, columns 7 to 9 present the average number of variables, constraints, and non-zero coefficients of the model, respectively.

Table 6.3: Average results of random instances on the single-period model.

Instances	Sol.	Opt.	Extr.	Build t.	Opt. t.	Vars	Constr	NZ
1-9	10	10	9	0.65	1.66	2,698	243	15,716
10-18	12	12	11	9.97	77.62	43,293	968	255,959
19-27	11	11	11	38.44	2043.26	173,387	1936	1,028,303

Table 6.4: Average results of random instances on the multi-period model.

Instances	Sol.	Opt.	Extr.	Build t.	Opt. t.	Vars	Constr	NZ
1-9	11	11	10	7.36	21.80	32,233	2,289	187,701
10-18	12	12	11	117.11	2222.21	483,643	9,207	3,067,944
19-27*	12	12	10	244.63	3854.60	1,058,000	15,350	6,266,855

*Results are computed on instances 19-25 because 26-27 go out-of-memory.

Our models are able to provide optimal solutions for large instances involving up to 400 sources, 200 candidate locations, and 5 waste types, showcasing strong scalability. From the single-period to multi-period models, the average optimization time increases by 13, 29, and 13 times for the multi-period model in the three groups, respectively (considering only instances 19-25 in the third group because instances 26-27 are out-of-memory for the multi-period model). The size of the model also increases significantly: on average, the number of variables, constraints, and non-zero coefficients increase by a factor of 10. On average, we find a number of solutions very close to the limit imposed by the parameter δ, in a similar way to the instance of Parma case study.

6.6 Conclusions

Urban waste management is an increasingly challenging activity for modern societies, in view of the dramatic increase of waste amount to be collected and recycled or disposed of, which has raised not only several logistic challenges, but also serious environmental issues in the public and private sectors. OR studies have recently started to address multi-objective optimization problems with economic and environmental goals with real-world applications in the waste management industry.

In this work, we have studied a real-world bi-objective capacitated facility location problem occurring in Iren Ambiente SpA, an Italian multi-utility company. The goal is to define the optimal network of intermediate transfer and final treatment facilities, also considering the opportunity of opening new facilities, with the twofold objective of minimizing the annual costs and CO_2 emissions generated by transports and facility operations. We have provided a mixed integer linear programming formulation for the single-period problem and then extended it to a multi-period setting. This extension is particularly important, since this model combines a multi-period setting with a multi-objective one for a facility location problem, an area of research which is still unexplored, as noted by Nickel and Saldanha da Gama (2019). Our models enable dividing the waste demand of each source among one or more facilities, ensuring that the total and waste-specific capacity constraints for each facility are satisfied in the overall time horizon. In the multi-period model, the waste flow allocation decisions are allowed to change period by period whereas facility opening decisions are unvaried, as required by the realistic assumptions given by the company.

We solved two real-world case studies provided by Iren Ambiente related to the districts of Reggio Emilia and Parma, focusing on one year (single-period model) and 12 months (multi-period model) to allow for seasonality (i.e., monthly changes of waste production). In both cases, minimum cost and minimum emission solutions show a significant reduction in costs and CO_2 emissions with respect to the status quo where almost no intermediate flow appears, showing the benefit of introducing intermediate transfer flows of waste within Iren Ambiente's networks.

We have developed an approximated ϵ-constraint algorithm, providing a Pareto front for each of our models and applied the hypervolume indicator to compare the quality of the fronts between each other: from the single-period model to the multi-period model, one gains around 19% and 18% of hypervolume reduction in the two case studies, respectively. Despite the greater computational effort, we have shown that the improvements obtained for the multi-period Pareto front are significant.

The models are general and can be applied to different case studies of waste collection that appear in the literature and in similar real-world contexts. Such a generality has been validated by further computational experiments on larger randomly generated instances inspired by our case studies. Specifically, we built well-shaped Pareto fronts of optimal solutions for instances with up to 400 sources, 200 locations, and five waste types within a reasonable computational time.

Interesting future research directions could address the extension of the considered problem to the related vehicle routing optimization problem of waste transport vehicles across the network, considering the overall costs and CO_2 emissions involved, again in a multi-objective setting (see, e.g., Rodrigues and Soeiro Ferreira, 2015). As future research, we are interested in further developing the multi-period setting of our problem by adding an additional set of variables and constraints to include capacity constraints on each period (instead of imposing them on the overall time horizon) where slack variables allow to keep the unused capacity not used in one period (if any) for use in the following period. This could add more flexibility to our multi-period model under the considered realistic setting affected by monthly waste fluctuations. We are also interested in applying our methods to different case studies with larger fluctuations of demand, cost, and emissions parameters over time to further observe the improvement of the multi-period model over the single-period one. Finally, we aim at developing multi-objective meta-heuristic algorithms to solve larger instances of our problem.

References

Adeleke, O. J., & Olukanni, D. O. (2020). Facility location problems: Models, techniques, and applications in waste management. *Recycling*, *5*(2), 10.

Amin Chaabane, A. R., & Paquet, M. (2011). Designing supply chains with sustainability considerations. *Production Planning & Control*, *22*(8), 727–741.

Asefi, H., Lim, S., Maghrebi, M., & Shahparvari, S. (2019). Mathematical modelling and heuristic approaches to the location-routing problem of a cost-effective integrated solid waste management. *Annals of Operations Research*, *273*, 75–110.

Cao, Y., Smucker, B. J., & Robinson, T. J. (2015). On using the hypervolume indicator to compare pareto fronts: Applications to multi-criteria optimal experimental design. *Journal of Statistical Planning and Inference*, *160*, 60–74.

Carbon Credits. (2023). Live Carbon Prices Today [https://carboncredits.com/carbon-prices-today [Accessed: 2023-03-01]].

Caselli, G., Delorme, M., Iori, M., & Magni, C. A. (2022). Mixed integer linear programming for a real-world parallel machine scheduling problem with workforce and precedence constraints. In L. Amorosi, P. Dell'Olmo, & I. Lari (Eds.), *Optimization in artificial intelligence and data sciences* (pp. 61–71). Springer International Publishing.

Colicchia, C., Creazza, A., Dallari, F., & Melacini, M. (2016). Eco-efficient supply chain networks: Development of a design framework and application to a real case study. *Production Planning & Control*, *27*(3), 157–168.

Correia, I., & Saldanha da Gama, F. (2019). Facility location under uncertainty. In G. Laporte, S. Nickel, & F. Saldanha da Gama (Eds.), *Location Science* (pp. 185–213). Springer.

Coutinho-Rodrigues, J., Tralhão, L., & Alçada-Almeida, L. (2012). A bi-objective modeling approach applied to an urban semi-desirable facility location problem. *European Journal of Operational Research*, *223*(1), 203–213.

Darmian, S. M., Moazzeni, S., & Hvattum, L. M. (2020). Multi-objective sustainable location-districting for the collection of municipal solid waste: Two case studies. *Computers & Industrial Engineering*, *150*, 106965.

DEFRA. (2019). Measuring and reporting environmental impacts: Guidance for businesses [Last updated 31 January 2019].

Ehrgott, M. (2005). *Multicriteria optimization*. Springer.

Eiselt, H. A., & Marianov, V. (2014). A bi-objective model for the location of landfills for municipal solid waste. *European Journal of Operational Research*, *235*(1), 187–194.

Erkut, E., Karagiannidis, A., Perkoulidis, G., & Tjandra, S. A. (2008). A multicriteria facility location model for municipal solid waste management in north greece. *European Journal of Operational Research*, *187*(3), 1402–1421.

Farahani, R. Z., Fallah, S., Ruiz, R., Hosseini, S., & Asgari, N. (2019). OR models in urban service facility location: A critical review of applications and future developments. *European Journal of Operational Research*, *276*(1), 1–27.

Gambella, C., Maggioni, F., & Vigo, D. (2019). A stochastic programming model for a tactical solid waste management problem. *European Journal of Operational Research*, *273*(2), 684–694.

Gendron, B., & Semet, F. (2009). Formulations and relaxations for a multi-echelon capacitated location–distribution problem. *Computers & Operations Research*, *36*(5), 1335–1355.

Ghiani, G., Laganà, D., Manni, E., Musmanno, R., & Vigo, D. (2014). Operations research in solid waste management: A survey of strategic and tactical issues. *Computers & Operations Research*, *44*, 22–32.

Ghiani, G., Laganà, D., Manni, E., & Triki, C. (2012). Capacitated location of collection sites in an urban waste management system. *Waste Management*, *32*(7), 1291–1296.

Golden, B. L., Assad, A. A., & Wasil, E. A. (2002). Routing vehicles in the real world: Applications in the solid waste, beverage, food, dairy, and newspaper industries. In *The vehicle routing problem* (pp. 245–286). SIAM.

Govindan, K., Jha, P. C., & Garg, K. (2016). Product recovery optimization in closed-loop supply chain to improve sustainability in manufacturing. *International Journal of Production Research*, *54*(5), 1463–1486.

Guerreiro, A. P., Fonseca, C. M., & Paquete, L. (2021). The hypervolume indicator: Computational problems and algorithms. *ACM Computing Surveys*, *54*(6), 1–42.

Hansen, P. (1980). Bicriterion path problems. In G. Fandel & T. Gal (Eds.), *Multiple criteria decision making theory and application* (pp. 109–127). Springer Berlin Heidelberg.

Harijani, A. M., Mansour, S., & Karimi, B. (2017). A multi-objective model for sustainable recycling of municipal solid waste. *Waste Management & Research*, *35*(4), 387–399.

Harris, I., Mumford, C. L., & Naim, M. M. (2014). A hybrid multi-objective approach to capacitated facility location with flexible store allocation for green logistics modeling. *Transportation Research Part E: Logistics and Transportation Review*, *66*, 1–22.

Laporte, G., Nickel, S., & Saldanha da Gama, F. (2019). *Location science*. Springer.

Lee, C. K. M., Yeung, C. L., Xiong, Z. R., & Chung, S. H. (2016). A mathematical model for municipal solid waste management–a case study in hong kong. *Waste Management*, *58*, 430–441.

Liu, W., Kong, N., Wang, M., & Zhang, L. (2021). Sustainable multi-commodity capacitated facility location problem with complementarity demand functions. *Transportation Research Part E: Logistics and Transportation Review*, *145*, 102165.

Modgil, S., Gupta, S., & Bhushan, B. (2020). Building a living economy through modern information decision support systems and UN Sustainable Development Goals. *Production Planning & Control*, *31*(11-12), 967–987.

Mohsenizadeh, M., Tural, M. K., & Kentel, E. (2020). Municipal solid waste management with cost minimization and emission control objectives: A case study of Ankara. *Sustainable Cities and Society*, *52*, 101807.

Nickel, S., & Saldanha da Gama, F. (2019). Multi-period facility location. In G. Laporte, S. Nickel, & F. Saldanha da Gama (Eds.), *Location Science* (pp. 289–310). Springer.

Rabbani, M., Mokhtarzadeh, M., & Farrokhi-Asl, H. (2018). A new mathematical model for designing a municipal solid waste system considering environmentally issues. *International Journal of Supply and Operations Management*, *5*(3), 234–255.

Reddy, K. N., Kumar, A., Choudhary, A., & Cheng, T. E. (2022). Multi-period green reverse logistics network design: An improved benders-decomposition-based heuristic approach. *European Journal of Operational Research*, *303*(2), 735–752.

ReVelle, C. (2000). Research challenges in environmental management. *European Journal of Operational Research*, *121*(2), 218–231.

Rodrigues, A. M., & Soeiro Ferreira, J. (2015). Waste collection routing—limited multiple landfills and heterogeneous fleet. *Networks*, *65*(2), 155–165.

Salamirad, A., Kheybari, S., Ishizaka, A., & Farazmand, H. (2023). Wastewater treatment technology selection using a hybrid multicriteria decision-making method. *International Transactions in Operational Research*, *30*(3), 1479–1504.

Saldanha da Gama, F. (2022). Facility location in logistics and transportation: An enduring relationship. *Transportation Research Part E: Logistics and Transportation Review*, *166*, 102903.
Soland, R. M. (1979). Multicriteria optimization: A general characterization of efficient solutions. *Decision Sciences*, *10*(1), 26–38.
Tirkolaee, E. B., Goli, A., Gütmen, S., Weber, G.-W., & Szwedzka, K. (2023). A novel model for sustainable waste collection arc routing problem: Pareto-based algorithms. *Annals of Operations Research*, *324*(1-2), 189–214.
Tolaymat, T., El Badawy, A., & Genaidy, R. S. A. (2015). A system-of-systems approach as a broad and integrated paradigm for sustainable engineered nanomaterials. *Science of the Total Environment*, *511*, 595–607.
Trading Economics. (2023). EU Carbon Permits [https://tradingeconomics.com/commodity/carbon [Accessed: 2023-03-01]].
Vaillancourt, K., & Waaub, J.-P. (2002). Environmental site evaluation of waste management facilities embedded into eugene model: A multicriteria approach. *European Journal of Operational Research*, *139*(2), 436–448.
Van Engeland, J., Beliën, J., De Boeck, L., & De Jaeger, S. (2020). Literature review: Strategic network optimization models in waste reverse supply chains. *Omega*, *91*, 102012.
Velázquez-Martínez, J., & Fransoo, J. (2017). Green facility location. In Y. Bouchery, C. J. Corbett, J. C. Fransoo, & T. Tarkan (Eds.), *Sustainable supply chains: A research-based textbook on operations and strategy* (pp. 219–234). Springer.
Verter, V. (2011). Uncapacitated and capacitated facility location problems. In H. A. Eiselt & V. Marianov (Eds.), *Foundations of location analysis* (pp. 25–37). Springer US.
Waltho, C., Elhedhli, S., & Gzara, F. (2019). Green supply chain network design: A review focused on policy adoption and emission quantification. *International Journal of Production Economics*, *208*, 305–318.
Wu, X., Huang, G. H., Lin, J., & Li, J. (2006). An interval nonlinear program for the planning of waste management systems with economies-of-scale effects—A case study for the region of Hamilton, Ontario, Canada. *European Journal of Operational Research*, *171*(2), 349–372.
Yu, H., & Solvang, W. D. (2017). A multi-objective location-allocation optimization for sustainable management of municipal solid waste. *Environment Systems and Decisions*, *37*(3), 289–308.

Appendix 6.A Cost and CO_2 emissions evaluation

For transport costs parameters C_{ijh}, we consider the operating costs related to manpower and vehicles, a portion of the indirect costs related to the operational service center, and a portion of operational headquarters and general costs of the company (measured in $[\frac{€}{year}]$). Transport costs parameters C'_{jkh} depend on the distance between facilities j and k, the type of waste h to be transported, the type of vehicle, the characteristics of departure and arrival facilities, the costs of operating manpower and vehicles, and another portion of indirect costs of the operational headquarters and general costs of the company (measured in $[\frac{€}{ton}]$). Variable usage costs parameters r_{jh} include all those cost components that vary in proportion to the amount of waste type h treated by facility j, including the costs for operating personnel, energy, routine maintenance of machines, and disposal of residual materials from plant operations, such as ashes from combustion and wastewater from cleaning (measured in $[\frac{€}{ton}]$). Fixed usage costs G_j, on the other hand, include rental and equipment costs for facility j, in addition to annual fixed expenses such as coordination, insurance, and other indirect costs not dependent on the amount of waste treated (measured in $[\frac{€}{year}]$).

As reported by Velázquez-Martínez and Fransoo (2017), only few studies in the facility location literature include a detailed formulation for CO_2 emission parameters. Moreover, these parameters are typically referred only to transport emissions, while the ones related to facilities are very rarely included in the models (see , e.g., Harris et al., 2014 and Reddy et al., 2022). In this regard, our work contributes to the literature by providing a detailed formulation of both types of emission parameters, which is presented in the following. This contribution emerged from our collaboration with the company's technicians and decision makers who deal with the considered facility location problem and the related case studies of the paper. They provided us with realistic approximations as well as real historical data.

Parameters e_{ijh}, indicating transport emissions from customer i to facility j for waste type h (measured in $[\frac{kgCO_2}{year}]$), are computed as follows:

$$e_{ijh} = m_{ij} d_{v(i)} W n_{ih} \tag{6.21}$$

where m_{ij} indicates the time (in hours) required by the vehicle type that serves customer i (i.e., one type for each customer, indicated as $v(i)$) to go from customer i to facility j, $d_{v(i)}$ represents the fuel consumption of vehicle $v(i)$ ([liters/hour]), W is the conversion factor ($W = 2.63$ kg CO_2 per liter of diesel fuel taken by DEFRA (2019), which is constant since we only deal with diesel vehicles in our study), and n_{ih} expresses the number of required trips over one year from customer i for waste type h. In Table 6.A.1, we report the values of fuel consumption (expressed in [liters/hour] of diesel fuel) of the different vehicles used in our case studies for the collection and transport of waste from customers to facilities (i.e., first level transport). Each customer is served by only one type of vehicle, which depends on the characteristics of the customer and the service. For example, automatic side loader compactors are huge vehicles used only for "street waste collection", which involves large stationary waste containers positioned on main roads. On the contrary, smaller rear loader compactors are used to collect waste in smaller roads, for instance, inside the historical centers of the municipalities.

Note that Table 6.A.1 includes all types of vehicles possibly used by Iren. However, only a subset of them is involved in each of the case studies presented in this work (i.e., light rear loader compactors, rear loader compactors, and automatic side loader compactors in Reggio Emilia; mini, medium, and heavy rear loader compactors in Parma). As regards the carbon emissions from facility to facility, parameters e'_{jkh}, indicating the amount of emissions for the transport of one ton of waste of type h from facility j to facility k (measured in $[\frac{kgCO_2}{ton}]$), are computed as follows:

$$e'_{jkh} = \frac{m_{jk} d_{v(j)} W}{l_{v(j)h}} \tag{6.22}$$

where m_{jk} indicates the time (in hours) required by the vehicle type that serves facility j (i.e., one type for each facility, indicated as $v(j)$) to go from facility j to facility k,

Table 6.A.1: Hourly fuel consumption of the different types of vehicles used for customer-facility transports.

Vehicle type $v(i)$	Fuel consumption $d_{v(i)}$
Mini rear loader compactor	3.55
Light rear loader compactor	5.00
Medium rear loader compactor	6.45
Rear loader compactor	7.50
Heavy rear loader compactor	8.70
Automatic side loader compactor	9.00

$d_{v(j)}$ represents the fuel consumption of this vehicle (measured in [liters/hour]), W is the conversion factor, and $l_{v(j)h}$ indicates the capacity of the loading vehicle $v(j)$ dedicated to waste type h (expressed in tons of waste). The latter depends on the size (i.e., volume) of the dedicated vehicle and on the density of the waste type. For example, plastic is a voluminous but very light waste, so the capacity of the vehicles used for collecting plastic waste are lower (and therefore unitary emissions parameters are higher) than the ones dedicated to the other waste types, which are typically heavier. In Table 6.A.2, we report the values of fuel consumption (expressed in [liters/hour] of diesel fuel) of the different vehicles used in our case studies for the collection and transport of waste from facilities to facilities (i.e., second level transport): semi-trailer and heavy semi-trailer trucks in Reggio Emilia; hooklift trucks and semi-trailer trucks in Parma. Each facility is served by only one type of vehicle, which depends on the loading system technology in use in the facility j of departure for the trip from facility j to facility k. For example, in the intermediate plants dedicated to plastic or paper waste, the material is pressed into transportable bales which are loaded directly into the semi-trailer truck for the transport to final facilities. The set of additional parameters introduced in

Table 6.A.2: Hourly fuel consumption of the different types of vehicles used for facility-facility transports.

Vehicle type $v(j)$	Fuel consumption $d_{v(i)}$
Hooklift truck	9.50
Semi-trailer truck	13.00
Haevy semi-trailer truck	15.00

this section (i.e., m_{ij}, m_{jk}, $d_{v(i)}$, $d_{v(j)}$, and n_{ih}) used for the computation of transport emissions parameters have been collected and given by the company on the considered case studies. For parameters p_{jh} and F_j related to the operating facility emissions, we estimated a different percentage of emissions for fixed and variable components, depending on the energy consumption of the operating machines in the facility, that is, the equipment used for loading, unloading, and moving waste. The greater the amount of waste collected by a single facility, the greater the environmental benefit from opening that facility. To compute the facility emissions parameters, we use historical data provided by the company as well as the support of the company's technicians. First, we estimate the unit value of CO_2 emissions produced by facility j when *fully* *operational* (i.e., working at its maximum capacity), indicated by $UEFO_j$ and expressed in $[\frac{kgCO_2}{ton}]$ as follows:

$$UEFO_j = UCFO_j W + UPFO_j W_e \tag{6.23}$$

where $UCFO_j$ (measured in $[\frac{liters}{ton}]$) and $UPFO_j$ (measured in $[\frac{kWh}{ton}]$) indicate, respectively, the fuel and electricity consumption for one ton of waste when the facility is fully operational. Such consumption components vary depending on the type of facility j for two main factors: (i) the types of waste that are treated by the facility (e.g., processes are more energy-consuming for plastic waste) and (ii) the shape of the facility (e.g., final facilities are more energy-consuming than intermediate ones since not only collection but also treatment processes are involved in the former). Each consumption component is multiplied by its conversion factor: $W = 2.63$ kgCO_2 per liter of diesel was taken by DEFRA (2019), while $W_e = 0.35$ kgCO_2 per kWh for electricity is given (based on the company electricity supply mix). We excluded other possible consumption components with no significant impact in our analysis.

Then, we obtain the total quantity of emissions of facility j when fully operational, indicated by $TEFO_j$ and (measured in $[\frac{kgCO_2}{year}]$), by multiplying the unit value of consumption by the overall annual capacity Q_j of the facility (assuming to use a facility at its maximum capacity), as follows:

$$TEFO_j = Q_j UEFO_j \tag{6.24}$$

Given the percentage η of energy consumption (and therefore CO_2 emissions) needed to keep open an *empty* facility (i.e., a facility with no waste to be treated), we compute the fixed component of carbon emissions for facility j, indicated by parameter F_j expressed in $[\frac{kgCO_2}{year}]$, as follows:

$$F_j = TEFO_j \eta \tag{6.25}$$

Finally, the variable component of carbon emissions for facility j is equal to the difference between its total amount of emissions and its fixed component of emissions. To obtain parameters p_{jh} for the variable component of emissions for each ton of waste type h and facility j (expressed in $[\frac{kgCO_2}{ton}]$), we consider a percentage of operations of facility j dedicated to waste type h, which is indicated by ω_{jh}, and the capacity Q_{jh} of facility j dedicated to waste h as follows:

$$p_{jh} = \frac{(TEFO_j - F_j)\omega_{jh}}{Q_{jh}} \tag{6.26}$$

The auxiliary parameters introduced in this section (i.e., $UCFO_j$, $UPFO_j$, η, and ω_{jh}) for the computation of carbon emission parameters related to operating facilities have been estimated by the company's technicians based on historical data and their specific knowledge of similar systems managed by the company.

The computation of cost and emissions parameters of all the real-world and randomly generated instances adopted in the article follows the description above.

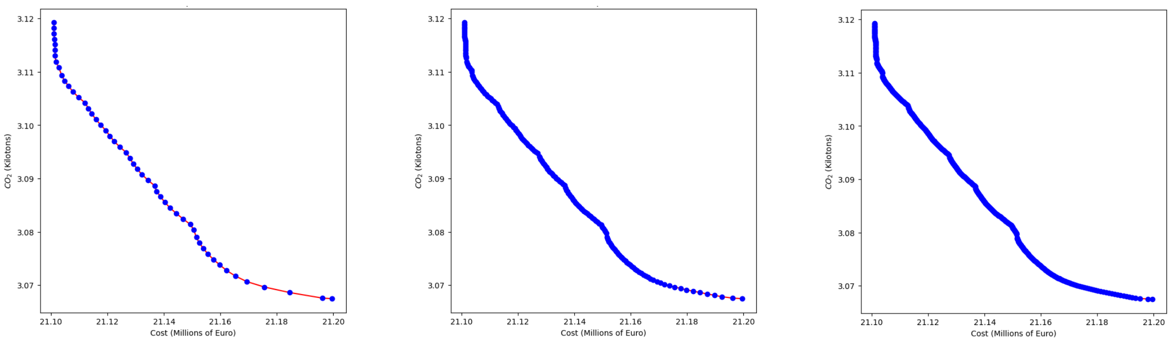
Appendix 6.B The case study of the Parma district

In this section, we study the performance of our bi-objective approaches on a second case study. The second case study provided by Iren Ambiente deals with the district of

Parma (423,000 inhabitants producing approximately 144,000 kilotons of urban waste per year), for which the "door-to-door" waste collection system is currently working in 40% of the municipalities of the district (mainly urban areas) and is going to be extended to the rest of the district (upland area). The study was motivated by the aim of optimizing the future collection network based on this new system (100% door-to-door waste collection) by opening new intermediate transfer facilities over the entire district, especially in the upland area, which is farther from final treatment plants. In this case, the estimated data provided by Iren Ambiente are based on company standards as well as historic data for a generic year of full door-to-door waste collection. We structure the analysis of results of our approximated ϵ-constraint method on Parma case study analogously to the Reggio Emilia case study.

6.B.1 Single-period model

A larger single-period model with 184,910 variables, 2,352 constraints, and 562,650 non-zero coefficients was built for Parma instance. This district has 235 sources, three existing final treatment facilities (located in Parma, Torrile, and Borgo Val di Taro) for different types of waste, one existing intermediate transfer facility in Borgo Val di Taro for organic waste, and 100 candidate intermediate transfer facilities located in Monchio delle Corti, Palanzano, Salsomaggiore Terme, and Tizzano, defined by five capacity configurations for each of the five types of waste, (paper, plastic, glass, organic, and unsorted). There is no candidate final facility. The three Pareto sets of solutions obtained using δ = 50, 200, 500 are represented in Figure 6.B.1.



(a) $\delta = 50$ (b) $\delta = 200$ (c) $\delta = 500$

Figure 6.B.1: Pareto fronts of the single-period model for Parma instance.

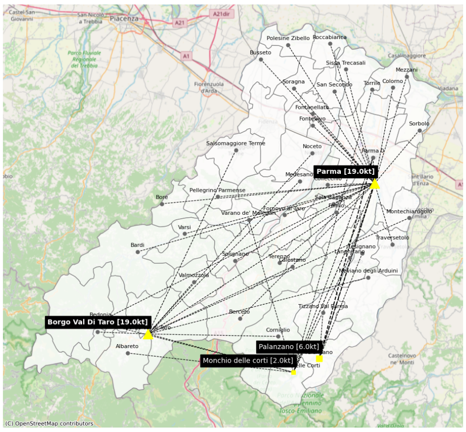
We observe much more dense sets of solutions (238 solutions and 40 extreme points on average). All solutions found are optimal. The average CPU time is 965 seconds. The larger size of the instance allows more flow variations, giving the model a broader range of solutions to be generated for the same set of selected facility locations to be opened (i.e., groups of solutions with similar opening decisions do not appear clearly from the graphs in this case, resulting in less readable fronts).

Similarly to the findings from the case study of Reggio Emilia presented in the article, a set of intermediate facilities are opened in both the minimum cost and the minimum emission solutions in Tizzano (for paper, organic, glass, and unsorted waste), Monchio delle Corti (for paper, plastic, glass and unsorted waste), and Palanzano (for plastic, organic, glass, and unsorted waste). Also, in this case, the reduction of costs and emissions on the aggregated waste of all types is significant, with emissions reduction equal to 19% and 20% in the minimum cost and minimum emission solutions respectively[3], with respect to the status quo where almost no intermediate flow is transported. The minimum cost solution has a total cost of 21.10 million euros and a total emission of 3.12 kilotons of CO_2. The minimum CO_2 solution has a total cost of 21.20 million euros and a total emission of 3.07 kilotons of CO_2.

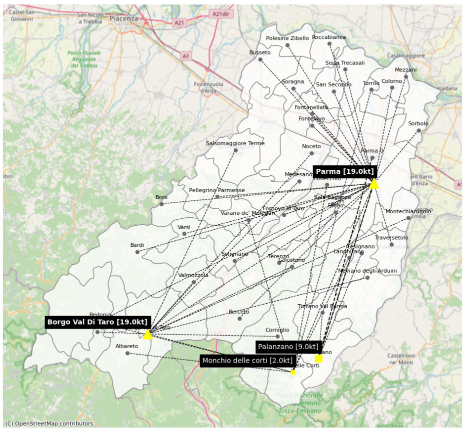
In Figure 6.B.2, we report the graphical representation of each waste type flow (i.e., paper, plastic, glass, organic, and unsorted waste) in the two solutions with minimum costs and minimum emissions, respectively. In this larger-size case study with more source points, three final existing facilities in Parma, Torrile, and Borgo Val di Taro, and five waste types, we observe different behaviors of minimum cost and minimum emission solutions for the five waste types. The waste collection model in Parma case study is door-to-door in almost half of the district. Light vehicles with rear loader systems stop in front of each building and a team of several operators get off the vehicle and handle the waste manually collecting it from the side of the street. However, depending on the type of waste, the type of vehicle and the number of operators vary. For waste types with high labor cost, such as paper and organic waste, we observe that the minimum cost solution tends to favor more first-level transports to intermediate transfer facilities, since the latter has a large economic impact on the total costs, in contrast with Reggio Emilia case study. On the other hand, we observe a different situation for plastic waste. For this particular type of waste, door-to-door collection has a lower labor cost (i.e., very light waste collected in sacks easily moved by a single operator) and a higher unit cost at the facility (i.e., light but voluminous waste which is more expensive to collect). Therefore, the minimum cost solution in this case tends to favor more direct flow from sources to final facilities.

(a) Minimum cost paper flow. (b) Minimum emission paper flow.

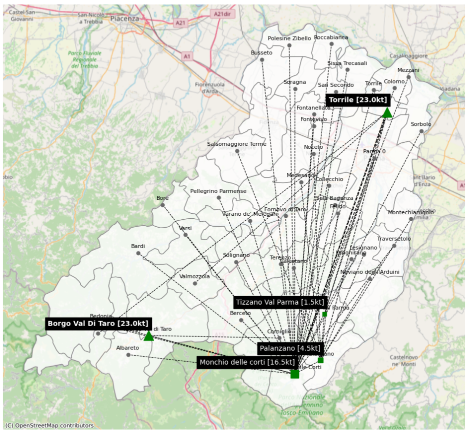
[3]Upon specific request by Iren Ambiente, we may not disclose the data on cost reduction.



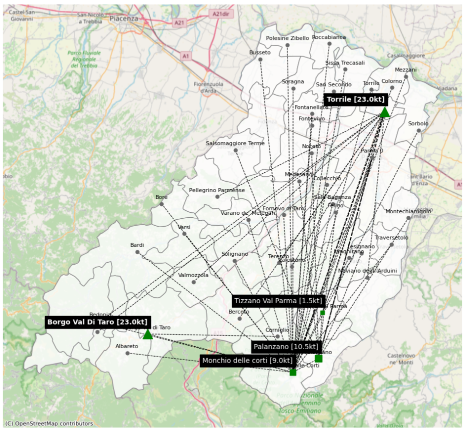
(c) Minimum cost plastic flow.



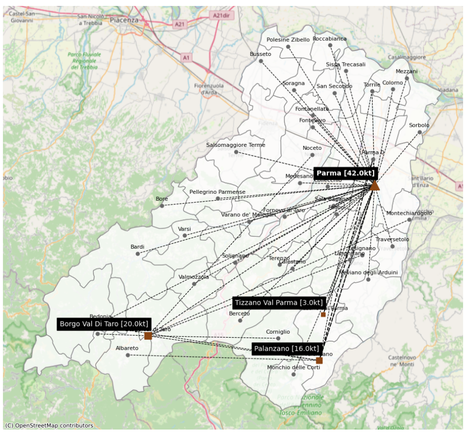
(d) Minimum emission plastic flow.



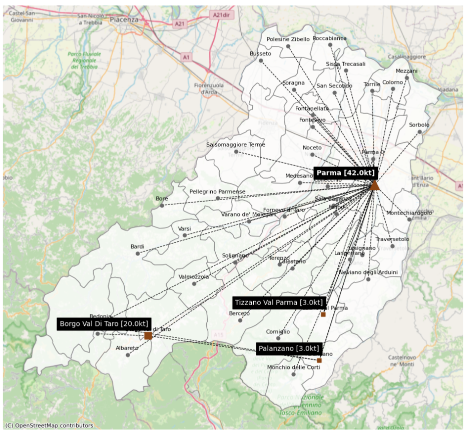
(e) Minimum cost glass flow.



(f) Minimum emission glass flow.



(g) Minimum cost organic flow.



(h) Minimum emission organic flow.

(i) Minimum cost unsorted flow. (j) Minimum emission unsorted flow.

Figure 6.B.2: Flow representation of minimum cost and minimum emission solutions for Parma case study (paper, plastic, glass, organic, and unsorted waste): triangles and squares represent final and intermediate facilities with shape size proportional to facility capacity, dots indicate the source points, and dashed lines represent the flow with thickness proportional to the amount of flow.

6.B.2 Multi-period model

The single-period and multi-period models were solved with $\delta = 50$. We obtained better Pareto fronts moving from the single-period to the multi-period model, finding solutions with lower costs and emissions, as shown in Figure 6.B.3. The multi-period model gains

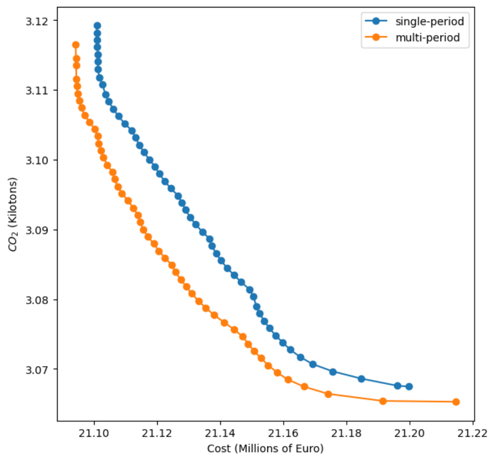


Figure 6.B.3: Comparison of the generated Pareto fronts with $\delta = 50$ for Parma case study: single-period model in blue and multi-period model in orange.

18.43% of hypervolume over the single-period model. Such improvement is substantial; therefore, it motivates the use of multi-period models.

Appendix 6.C Details on the random instances

In this section, we provide a detailed description of our random instances followed by a presentation of the complete set of results obtained by the ϵ-constraint method we implemented on the set of random instances. We recall that these instances are publicly available at https://github.com/regor-unimore/Cost-and-CO2-minimization-in-Facility-Location.

6.C.1 Random instances

First, we generated instances for our single-period model as follows:

- We define a random rectangle centered in a fixed geographic point with an area measured in km^2 defined by parameter A as limiter for the coordinates (latitude and longitude) of sources and facility locations sets N and M, which are then randomly generated.

- A proportion $M_e = 5\%$ of the generated facility locations is randomly selected to indicate existing facilities. In a similar way, final treatment facilities are randomly selected following proportion $M_f = 5\%$. To guarantee instance feasibility, the last facility is always generated as final and existing.

- Sources waste demand q_{ih} and facility capacity Q_{jh} values are randomly generated following uniform distributions limited by lower and upper bounds $[q_l, q_u] = [0, 10{,}000]$ and $[Q_l, Q_U] = [0, 40{,}000]$. The values of overall capacity Q_j are computed between 90% and 100% of the sum of the waste-specific capacities.

- The waste types are randomly chosen from the set of all five options: paper, plastic, glass, organic, and unsorted waste. Variable and fixed costs of each facility are computed using a set of previously calculated coefficients for each type of waste and (intermediate or final) facility based on real-world data provided by Iren Ambiente's case studies. Values of fixed and variable CO_2 emissions of the facilities are defined proportionally to the total capacity of the facility.

- Transportation cost and emission parameters from sources to facilities are calculated using a set of coefficients for each type of waste (obtained by analysing real-world data based on the collection method practiced for each waste type by Iren Ambiente), the waste demand, and the distance between the source and the facility. In a similar way, cost and emission parameters for the transportation between facilities are calculated using another set of coefficients derived by the case studies based on the waste type and the distance between them.

Distances are measured in km and calculated by the classical Haversine formula. Then, our instances were adapted to generate multi-period ones with $T = |12|$ (as in our real-world case studies), using a flat distribution for transportation cost and emission parameters from sources to facilities C_{ijht} and e_{ijht}, and a random normal distribution for waste demand of sources q_{iht}.

6.C.2 Detailed computational results on the ϵ-constraint method

Tables 6.C.1-6.C.2 report the results obtained by the ϵ-constraint method applied to the single-period and multi-period models on the full set of random instances. Costs are in million euros and CO_2 emissions are in kilotons. The second and third columns report the value of the two objectives for solutions of minimum costs and minimum emissions, respectively. The fourth fifth columns count the number of solutions found and the number of optimal ones, respectively. Column 6 indicates the number of extreme points on the Pareto front. Columns 7 and 8 report the building and optimization times expressed in seconds (with "T.LIMIT" indicating when the global time limit is reached). Finally, columns 9 to 11 give the number of variables, constraints, and non-zero coefficients of the model.

Table 6.C.1: Results of random instances on the single-period model (costs in M€ and CO_2 in kilotons).

Inst.	Min cost	Min CO_2	Sol.	Opt.	Extr.	Build t.	Opt. t.	Vars	Constr	NZ
1	(21.50, 6.54)	(21.65, 6.49)	12	12	8	0.08	0.20	275	67	1,563
2	(93.46, 19.85)	(94.20, 19.62)	12	12	10	0.22	0.46	815	185	4,657
3	(125.84, 23.77)	(127.49, 23.54)	12	12	12	0.43	0.53	1,355	303	7,751
4	(13.62, 5.03)	(14.09, 4.94)	12	12	9	0.19	0.43	600	82	3,468
5	(66.4, 11.06)	(67.17, 10.95)	12	12	12	0.43	1.01	1,780	220	10,342
6	(117.72, 21.77)	(119.23, 21.42)	12	12	11	0.71	1.49	2,960	358	17,216
7	(13.20, 4.65)	(13.28, 4.59)	4	4	4	0.49	0.64	1,850	127	10,768
8	(77.07, 13.45)	(79.47, 12.88)	6	6	5	1.31	2.82	5,500	323	32,150
9	(105.41, 19.70)	(108.55, 18.37)	11	11	7	1.99	7.40	9,150	519	53,532
10	(104.34, 8.93)	(105.19, 8.18)	11	11	11	1.06	1.94	4,380	262	25,698
11	(376.09, 51.36)	(378.87, 50.78)	12	12	12	3.27	5.78	13,100	738	76,970
12	(582.82, 116.57)	(592.32, 113.93)	12	12	12	5.71	10.10	21,820	1,214	128,242
13	(43.44, 15.06)	(43.71, 14.95)	12	12	11	2.32	4.49	9,520	322	56,110
14	(351.10, 40.75)	(354.67, 39.46)	12	12	11	6.92	17.93	28,480	876	168,084
15	(469.28, 83.84)	(474.66, 82.13)	12	12	12	11.29	33.54	47,440	1,430	280,058
16	(94.59, 9.60)	(97.90, 8.06)	12	12	11	8.31	43.27	29,500	502	174,478
17	(253.87, 44.06)	(261.42, 40.94)	11	11	8	19.01	180.87	88,300	1,290	522,822
18	(400.62, 67.89)	(410.43, 64.28)	12	12	12	31.81	400.64	147,100	2,078	871,166
19	(378.91, 25.29)	(381.81, 24.60)	11	11	6	4.47	8.97	17,520	522	103,393
20	(728.59, 145.94)	(747.66, 141.53)	12	12	12	13.79	29.63	52,480	1,476	310,284
21	(924.63, 159.19)	(937.97, 155.92)	12	12	12	22.75	47.15	87,440	2,430	516,941
22	(206.82, 19.83)	(210.94, 17.50)	12	12	11	9.49	24.79	38,080	642	225,500
23	(668.12, 86.26)	(680.52, 84.39)	12	12	12	28.11	74.63	114,080	1,752	676,010
24	(827.92, 138.44)	(841.11, 133.61)	12	12	11	46.55	268.93	190,080	2,862	1,126,285
25	(85.49, 32.69)	(86.17, 32.61)	12	11	12	25.64	1,668.25	118,000	1,002	699,998
26	(403.82, 91.88)	(416.11, 89.73)	12	12	12	75.25	5,466.95	353,600	2,580	2,098,772
27	(841.99, 153.59)	(868.73, 149.14)	8	2	7	119.87	T.LIMIT	589,200	4,157	3,497,545

We observe that the single-period model is able to find at least one optimal solution for every instance of the set. The multi-period model reaches the memory limit ("M.LIMIT") on the largest instances 26 and 27. This is due to the larger number of variables, constraints and non-zero coefficients. To solve larger instances, it is desirable to apply different approaches based on a meta-heuristic algorithm.

For most of our random instances, we find a number of solutions very close to the limit imposed by the parameter δ, in a similar way to the instance of Parma case study. The sole exceptions are small instances (7 and 8), having a behavior more similar to the instance of Reggio Emilia case study, which is indeed smaller than Parma instance. Such observation seems to indicate that larger instances allow more variation of the solutions, permitting a more diverse and dense resulting Pareto set.

Table 6.C.2: Results of random instances on the multi-period model (costs in M€ and CO_2 in kilotons).

Inst.	Min cost	Min CO_2	Sol.	Opt.	Extr.	Build t.	Opt. t.	Vars	Constr	NZ
1	(21.09, 6.44)	(21.22, 6.40)	12	12	12	0.93	1.52	3,245	661	18,525
2	(92.35, 19.56)	(92.92, 19.33)	12	12	12	2.97	4.32	9,725	1,967	55,543
3	(124.98, 23.50)	(125.9, 23.28)	12	12	12	4.29	6.22	16,205	3,273	92,561
4	(13.46, 4.95)	(13.67, 4.86)	12	12	12	1.73	2.95	7,090	731	41,165
5	(66.1, 10.97)	(66.99, 10.84)	12	12	12	5.32	10.44	21,250	2,167	123,433
6	(116.32, 21.33)	(117.55, 21.06)	12	12	12	8.56	17.10	35,410	3,603	205,701
7	(13.13, 4.57)	(13.19, 4.55)	12	12	8	5.18	14.66	21,925	930	128,094
8	(76.78, 13.31)	(79.16, 12.76)	5	5	5	14.15	32.82	65,725	2,732	384,128
9	(105.07, 19.48)	(108.1, 18.20)	10	10	7	23.15	106.21	109,525	4,534	640,162
10	(103.69, 8.77)	(104.88, 8.14)	11	11	10	13.28	39.80	52,340	2,660	307,474
11	(372.63, 50.38)	(374.85, 49.80)	12	12	12	38.34	104.56	156,980	7,932	922,298
12	(568.7, 112.35)	(574.06, 110.45)	12	12	11	64.24	257.27	261,620	13,204	1,537,122
13	(42.63, 14.80)	(42.91, 14.73)	12	12	12	26.39	72.59	113,800	2,929	671,527
14	(349.12, 40.10)	(352.45, 38.93)	12	12	10	79.16	578.78	341,320	8,697	2,014,335
15	(464.34, 82.31)	(469.68, 80.74)	12	12	12	133.63	1,499.86	568,840	14,465	3,357,143
16	(94.38, 9.41)	(97.00, 7.91)	12	12	12	75.55	1,503.89	352,900	3,736	2,089,270
17	(252.80, 43.54)	(260.37, 40.44)	11	11	8	233.10	5,143.18	1,058,500	10,992	6,267,198
18	(399.42, 67.05)	(409.21, 63.65)	12	11	12	390.32	T.LIMIT	1,764,100	18,248	10,445,126
19	(377.07, 24.91)	(380.45, 24.18)	11	11	6	51.54	291.67	209,800	5,329	1,238,923
20	(708.23, 139.81)	(720.56, 135.96)	11	11	11	154.52	814.98	629,320	15,897	3,720,735
21	(908.83, 154.37)	(917.67, 151.85)	12	12	12	259.41	1,334.76	1,048,840	26,465	6,199,739
22	(204.98, 19.31)	(209.51, 17.30)	12	12	12	106.04	1,330.55	456,080	5,867	2,702,425
23	(661.11, 84.55)	(671.29, 82.34)	12	12	12	320.65	2,585.96	1,368,080	17,427	8,106,785
24	(819.73, 136.10)	(832.54, 131.50)	12	12	10	537.22	9,824.26	2,280,080	28,987	13,508,325
25	(85.16, 32.58)	(85.95, 32.45)	11	11	10	283.06	T.LIMIT	1,413,800	7,481	8,391,055
26	-	-	-	-	-	M.LIMIT	-	-	-	
27	-	-	-	-	-	M.LIMIT	-	-	-	

Appendix 6.D Tailored weighted sum approach

As reported by Ehrgott (2005), "the traditional approach to solving multicriteria optimization problems of the Pareto class is by scalarization, which involves formulating a single objective optimization problem by means of a real-valued scalarizing function typically being a function of the objective functions of the multi-objective problem". The weighted sum method, indicated by the authors as the "simplest" method for multi-objective problems, defines a single-objective function by multiplying each objective by a weight and uses the vector of weights as a parameter. We propose an MILP-based weighted sum approach to solve the bi-objective problem by converting CO_2 emissions into costs and then scalarizing a single-objective function with different pairs of weights on the two original objectives. Then, we propose a greedy-based algorithm based on the weighted sum method, where we first select the facilities to be opened in a greedy way and then solve an LP model to allocate the waste flow.

6.D.1 MILP-based weighted sum method

To solve our bi-objective problem, we first apply an MILP-based weighted sum method as follows. We define our single-period and multi-period weighted objective functions Z_{WS} and Z_{WS}^m (in units of cost), respectively, as

$$Z_S = \lambda_C Z_C + \lambda_E p Z_E \tag{6.27}$$

$$Z_{WS}^m = \lambda_C Z_C^m + \lambda_E p Z_E^m \tag{6.28}$$

where λ_C and λ_E are the weights assigned, respectively, to cost and emission functions (equations (1)-(2) for the single-period model and (11)-(12) for the multi-period model of the article), and p is the market unit price of CO_2 (i.e., conversion factor of emissions into costs). Taking the cost of emitting CO_2 into account is a common practice in the current world of private and public businesses (e.g., evaluating the business carbon footprint, embracing decarbonization practices, and participating in regulated public bids). Our MILP-based weighted sum approach solves (6.27) subject to equations (3)-(10) (single-period model) and (6.28) subject to equations (7), (10), (13)-(18) (multi-period model) of the article. Assuming that the sum of weights is always equal to one (see Ehrgott, 2005), we try different combinations of weights respecting this rule between the two extreme single-objective solutions (i.e., $\lambda_C = 1.00, \lambda_E = 0.00$ and $\lambda_C = 0.00, \lambda_E = 1.00$, with both cases included). In the case of the two extreme solutions, we include an additional step after the single-objective optimization in which we look for non-dominated solutions. For instance, when optimizing costs only (i.e., $\lambda_C = 1.00, \lambda_E = 0.00$), the model can find an optimal value Z_C^* (or Z_C^{m*}) with an arbitrarily bad value of emissions Z_E (or Z_E^m). Therefore, we re-optimize the problem by minimizing emissions only (i.e., $\lambda_C = 0.00, \lambda_E = 1.00$) with the additional constraint that limits the total costs to be lower than or equal to the optimal value found in the first step, that is, Z_C^* (or Z_C^{m*}). In this way, we search for solutions with the same cost and minimum emissions. We do the same for the other extreme case (i.e., $\lambda_C = 0.00, \lambda_E = 1.00$), in which we first optimize emissions finding the optimal value Z_E^* (or Z_E^{m*}) and then we optimize costs but constraining the total value of emissions to be lower than or equal to the optimal value previously found. This additional step is not required for the intermediate solutions (i.e., $\lambda_C > 0.00$ and $\lambda_E > 0.00$). We consider each combination of weights as a possible preference of the decision maker. We solve our single-period and multi-period models for each selected pair of weights and observe the different model solutions in terms of objective function values and decision variables.

6.D.2 Greedy-based weighted sum method

We also apply a greedy-based weighted sum method in two phases: first, we select the facilities in a greedy way; then, we optimize the network flow through a Linear Programming (LP) model. This gives us the advantage of exploiting the power of LP while remaining fast since the LP model is a simple problem solvable in polynomial time. The detailed steps of our heuristic weighted sum approach for the single-period model are the following:

1. We select the facility locations to be opened by applying a greedy heuristic:
   (a) For each facility j, we initialize the total (scalarized) cost D_j to the fixed cost of opening the facility as follows: $D_j = G_j + pF_j$, where G_j is the fixed usage cost of opening facility j and pF_j is the cost of fixed CO_2 emissions for operating facility j (converted in costs by the conversion parameter p);
   (b) For each facility j, we compute the variable cost of fulfilling demand q_{ih} of waste type h of customer i considering ordered demands by increasing cost $d_{ih} = c_{ijh} + pe_{ijh}$ (i.e., total transport costs) until there is capacity Q_j (and Q_{jh} for waste type h). We sum the (transport) variable component of cost to total cost D_j;

(c) We order facilities by increasing total cost D_j weighted by total capacity Q_j (i.e., by increasing order of $\frac{D_j}{Q_j}$) to avoid penalizing larger facilities;

(d) We select and open facilities following the considered order until all the waste demand is fulfilled;

2. We take the solution of the greedy heuristic and we solve an LP which is a modification of model (1)-(10) of the article where integer location variables y are fixed, that is, $\bar{y}_j = 1$ for each facility selected in Step 1, and 0 otherwise.

We adapt this approach to the multi-period problem, by computing and ordering waste demands q_{iht} per customer i, waste type h, and time period t, and then we adapt the next steps accordingly. Note that the only modifications are in Steps 1b and 1c, since the location variables y are fixed in our multi-period model. In the following, we present the results of our weighted sum approaches on the two real cases and on the random instances. For every instance, a time limit of 1200 seconds was imposed to the solver to conclude each optimization step of the weighted sum method, and a global time limit of three hours was set for the overall weighted sum method. We adopt a unit price of CO_2 equal to $p = 100\frac{€}{tonCO_2}$, the prevailing trend in the European Emissions Trading System (see Carbon Credits, 2023 and Trading Economics, 2023).

6.D.3 Results on the case study of the Reggio Emilia district

Table 6.D.1 shows the results of the MILP-based and greedy-based weighted sum methods of Sections 6.D.1-6.D.2. We compute five trade-off solutions considering $\lambda_C = \{1.00, 0.75, 0.50, 0.25, 0.00\}$ and $\lambda_E = 1.00 - \lambda_C$. For each solution, we provide the values of the weighted sum function in million euros (M€) for the MILP-based ("MILP") and greedy-based methods ("Greedy"), as well as the percentage gap between the two ("Gap"), for both the single-period and multi-period models. We observe that

Table 6.D.1: Weighted sum method on Reggio Emilia case study (Z^m_{WS} and Z^m_{WS} in M€).

Weights		Single-period (Z_{WS})			Multi-period (Z^m_{WS})		
λ_C	λ_E	MILP	Greedy	Gap	MILP	Greedy	Gap
1.00	0.00	2.364	3.087	31%	2.361	3.061	30%
0.75	0.25	1.800	2.343	30%	1.798	2.323	29%
0.50	0.50	1.236	1.599	29%	1.234	1.586	29%
0.25	0.75	0.671	0.856	28%	0.670	0.849	27%
0.00	1.00	0.106	0.112	6%	0.106	0.112	6%

the MILP-based approach is always more than 5% better than the greedy-based one, with an overall average gap of 25% and 24% in the single-period and multi-period cases, respectively. Of special importance is the case of the trade-off solution where $\lambda_C = 0.50$ and $\lambda_E = 0.50$ (i.e., costs and emissions are equally weighted). By doubling the weights, we get the case of $\lambda_C = 1.00$ and $\lambda_E = 1.00$, which provides an evaluation of the total costs. The difference between the total costs of the two methods supplies the net benefit of the MILP-based method as opposed to the greedy-based one. We see that, in both cases of single-period and multi-period models, we get a substantial net benefit greater than 0.7 million euros. This motivates the use of the MILP-based approach instead of the simpler greedy-based one.

6.D.4 Results on the case study of the Parma district

Table 6.D.2 shows the results of the MILP-based and greedy-based weighted sum methods on Parma case study, with the same structure that has been presented for Reggio Emilia case study. The MILP-based approach is, on average, 4% better than the greedy-based one in both the single-period and multi-period cases. With the same line of reasoning of Section 6.D.3, we evaluate the net benefit of MILP-based approach corresponding to the case $\lambda_C = 0.50$ and $\lambda_E = 0.50$. Even in this case study, we observe that the MILP-based approach gives rise to significant cost savings greater than 0.5 million euros, in both cases of single-period and multi-period models.

Table 6.D.2: Weighted sum method on Parma case study (Z^m_{WS} and Z^m_{WS} in M€).

Weights		Single-period (Z_{WS})			Multi-period (Z^m_{WS})		
λ_C	λ_E	MILP	Greedy	Gap	MILP	Greedy	Gap
1.00	0.00	21.101	21.667	3%	21.094	21.650	3%
0.75	0.25	15.904	16.334	3%	15.899	16.321	3%
0.50	0.50	10.706	11.000	3%	10.703	10.992	3%
0.25	0.75	5.509	5.667	3%	5.507	5.663	3%
0.00	1.00	0.307	0.330	8%	0.307	0.329	7%

6.D.5 Results on the random instances

We applied the MILP-based and greedy-based weighted sum approaches to the set of random instances described in detail in Appendix 6.C. In Table 6.D.3, we report the summary of average results on the first 25 instances for each of the trade-off solutions in the single-period and multi-period models, respectively, using the same five combinations

Table 6.D.3: Weighted sum method on random instances: average results (Z^m_{WS} and Z^m_{WS} in M€).

Weights		Single-period (Z_{WS})			Multi-period (Z^m_{WS})		
λ_C	λ_E	MILP	Greedy	Gap	MILP	Greedy	Gap
1.00	0.00	285.234	286.733	1%	281.684	283.214	1%
0.75	0.25	215.097	216.222	1%	212.407	213.554	1%
0.50	0.50	144.959	145.710	1%	143.130	143.894	1%
0.25	0.75	74.818	75.196	1%	73.849	74.231	1%
0.00	1.00	4.543	4.585	2%	4.447	4.486	2%

of weights λ_C and λ_E used in the case studies. We removed instances 26 and 27 from the computation of summary results because they went out of memory for the MILP-based approach. We observe that our MILP-based approach is always at least 1% better than the corresponding greedy-based one and is able to solve even very large instances with 400 sources of waste. These results confirm that the former approach is more efficient than the latter and enforces the use of mathematical models for solving even large-size instances of the considered problem.

Chapter 7

Optimal Vehicle Replacement with Budget Constraints and CO_2 Emissions Minimization[1]

Abstract

In this work, we study the bi-objective parallel replacement problem with finite horizon, constant demand of assets over time periods, and budget constraints for multiple families of vehicles with multiple vehicle types for replacement. The goals are the minimization of total discounted costs, including purchasing costs for new vehicles, operating and maintenance costs for vehicles in use, net of salvage values for replaced vehicles, and the minimization of vehicular CO_2 emissions. We propose a specific dynamic program and two integer programming formulations solved with the ϵ-constraint method, providing some preliminary computational results. The proposed methods are intended to solve the real-word case studies from two Italian companies. The aim is to integrate the financial and environmental sustainability perspectives within vehicle replacement to optimize not only the economic profitability of the fleet but also the shift towards a greener fleet. The final goal is to contribute to the replacement literature with effective methods that could be applied to analogous applications.

7.1 Introduction

In the vehicle replacement problem, a company owns a fleet of vehicles and must decide in each time period whether to replace them to minimize total discounted costs. More generally, the problem is called Equipment Replacement Problem (ERP), for which we refer the reader to Hartman and Tan (2014). In this work, we focus on ERPs with finite horizon and multiple asset. Various versions of the multi-asset ERP with finite horizons are explored in the literature, including those examining (i) economies of scale (Büyüktahtakın et al., 2014) and budget constraints (Hartman, 2000) on replacement costs, which make multiple assets interdependent (i.e., parallel

[1]Caselli, G., Hartman, J., Iori, M., Magni, C. A., Zucchi, G. (2024). Optimal vehicle replacement with budget constraints and CO_2 emissions minimization (technical report presented at *POMS2023 International Conference*, Paris, France, July 18–20, 2023).

replacement) and therefore the problems more challenging to solve; (ii) technological change in asset replacement alternatives for more realistic settings (Büyüktahtakın and Hartman, 2016); (iii) environmental considerations within composite objective functions (Rajabian et al., 2021). Despite extensive literature on ERPs (see Section 7.2 below), certain practical applications remain unexplored. This includes the focus of our study. Additionally, recent research has emerged concerning environmental considerations in vehicle fleet management, prompted by governmental regulations, industrial initiatives, and growing public awareness of the impact of vehicular operations and transportation on pollution and climate change (see, e.g., Environmental Protection Agency (EPA), 2019; Kagawa et al., 2013). In this work, we address an original vehicle replacement problem in which, in every period of the given finite time horizon, a keep-or-replace decision must be made for every asset of the fleet while meeting constant demand and budget constraints with two distinct objective functions: minimizing total discounted costs and minimizing total vehicular CO_2 emissions. The problem is inspired by real-world applications and can be used to model a large variety of further applications. In these large-size case studies, the fleet of vehicles for logistics and moving services comprises multiple asset families with different features and uses, each with a selection of alternatives (e.g., diesel, gasoline, hybrid, electric), each representing a "challenger" type. Replacement decisions are made for each family, considering all the options for replacement (e.g., electrification of the fleet), to reduce costs and emissions. The families of assets are interdependent due to budget constraints. We provide general formulations to be applied to a variety of analogous replacement applications.

We directly incorporate CO_2 emissions from vehicles, while recent works in the replacement literature account for emissions as additional or external costs within a single minimum-cost objective function (see, e.g., Rajabian et al., 2021). Our bi-objective approach allows the decision maker to control both objectives by analyzing the Pareto front of "efficient" solutions (i.e., the set of optimal solutions in the two-dimensional space of objectives for which it is not possible to improve one objective without worsening the other).

We present Dynamic Programming (DP) and Integer Linear Programming (ILP) formulations, which have been utilized in the replacement literature since the fifties (Bellman, 1955) and the eighties (Vemuganti et al., 1989), respectively. Our goal is to design efficient formulations and related bi-objective solution methods capable of solving large-scale instances of the problem to optimality. Initially, we formulate the specific problem involving a single family of assets and same-type replacement decisions (i.e., one challenger). We develop a tailored DP algorithm and two alternative ILP formulations with a bi-objective ϵ-constraint method. Then, we formulate the general problem with multiple families of assets and multiple challengers for each family. In this context, the bi-objective approach enables the simultaneous minimization of costs and emissions. This is achieved not only by replacing assets at the optimal time point for cost/emission trade-off but also considering various asset replacement options (e.g., fleet electrification). We preliminarily test our models and show that ILP models are promising solution methods for our real-world case studies, while more advanced techniques are required to improve the computational performance of the DP algorithm on the general problem. Our final goal is to solve real-world instances of the problem to optimality to provide companies with cost/emission Pareto fronts of solutions to the companies to balance the financial and environmental sustainability of the replacement process.

Section 7.2 provides a concise literature review on related vehicle replacement problems. In Section 7.3, we provide a detailed description of the problems. In Section 7.4, we present our DP model with an illustrative example. In Section 7.5, we present our ILP models and the proposed bi-objective ϵ-constraint methods. Finally, Section 7.7 presents our preliminary computational results and Section 7.8 reports some concluding remarks and future directions of research.

7.2 Concise literature review

The ERP, that is, the problem of determining the optimal procedure for the replacement of old equipment by new, has been studied in the mathematical programming literature since the 1950s. Well-known applications of the ERP refer to buses, trucks, aircrafts, and machines, among many. A survey on ERP is given by Hartman and Tan (2014). In this section, we aim to briefly explore the main characteristics, solution methods, and applications of ERPs related to our problem. Finally, we focus on similar ERPs with environmental considerations.

Characteristics The classical ERP focuses on minimizing the total discounted costs associated with replacing a single asset over an infinite time horizon. These costs include purchasing a new asset, operating and maintenance (O&M) costs for the asset in use, and salvage values for selling the old asset. Later on, various ERP problems have been studied in the literature. Under technological change, one considers non stationary costs and may be encouraged to replace assets before their economic life in order to take advantage of technologically more advanced models. The ERP with non stationary costs is more challenging to solve and often simplified in the literature under the finite horizon setting (see, e.g., Fraser and Posey, 1989). Multiple asset replacements are common in practice, with two main scenarios: serial and parallel replacement. Serial replacement involves independent assets and can be solved as individual single-asset replacement problems. In more realistic settings, assets are interdependent due to one or multiple factors. Parallel replacement includes (i) economies of scale, such as fixed charges on purchases of new assets, introduced by Jones et al. (1991) and then extensively studied (see, e.g., Büyüktahtakın et al., 2014), (ii) demand constraints (see, e.g., Rajagopalan, 1998) and (iii) budget constraints (see, e.g., Karabakal et al., 1994; Karabakal et al., 2000). A general parallel replacement model is presented by Hartman (2000). Additionally, assets can be homogeneous or heterogeneous. In the homogeneous model, a group of similar assets must be replaced simultaneously, while the heterogeneous model deals with optimizing multiple types of assets simultaneously, such as fleets with different vehicle types. The latter is more complex but reflects real-world scenarios more accurately (see, e.g., Ansaripoor and Oliveira, 2018).

Solution methods DP has been early introduced in the ERP literature by Bellman (1955) and Wagner (1975). In Bellman's model, each node represents the age of the asset, which is the state space of the model, and each arc represents either a keep or replace decision. In Wagner's model, the state space of the model is the time period and the decisions are the number of periods to retain an asset. DP has been predominantly applied to serial replacement or small-scale parallel replacement problems with homogeneous assets. Hartman (2004) propose a stochastic DP algorithm for homogeneous parallel replacement and utilization with stochastic demand and economies of scale in purchase prices. The authors solve the small case with two assets. Hartman and Rogers (2006) extend the problem with continuous and discontinuous technological change with probabilistic arrivals and multiple challengers both in Bellman's and Wagner's DP models, although showing that none of them is immune from the curse of dimensionality (i.e., computing memory issue due to the large-size state space of the model) when the instance size grows greater than 20 time periods (single-asset) and 10 time periods (multi-asset).

In such cases, ILP is more frequently applied. Vemuganti et al. (1989) formulate the single-asset replacement for a fleet of same-type vehicles with constant demand and finite horizon as a minimum cost-flow network model and then extend it to multi-asset scenarios with budget constraints and fleet-size variations. Karabakal et al. (1994) and Karabakal et al. (2000) tackle multi-challenger ERPs with budget constraints and Net Present Value (NPV) maximization using ILP formulations, Branch&Bound, and approximate solution methods based on Lagrangian relaxations, achieving optimality on small-size instances and near-optimal solutions for large-size instances with up to 500 assets. Hartman (2000) propose an ILP model for homogeneous assets with constant demand, budget constraints, and minimization of total discounted costs with variable discount rates. The authors solve small instances to optimality with up to 10 periods with fixed and fluctuating demand of assets over periods. Büyüktahtakın et al. (2014) focus on heterogeneous parallel replacement problems, proving the $\mathcal{NP}$-hardness of the problems and providing valid inequalities to improve solution efficiency. Results show that `CPLEX` solver is able to solve to optimality large-size instances with 500 and 100 time periods in a few seconds.

Beyond DP and ILP, heuristic algorithms are proposed to handle larger and more complex ERP instances effectively. Wang and Nguyen (2017) introduce a pattern search-genetic algorithm for stochastic DP models with expected NPV maximization, while Rajagopalan (1998) devise an ILP-based heuristic for general replacement, expansion, and disposal problems. Chand et al. (2000) present an iterative heuristic algorithm based on decomposition.

While this overview primarily covers mathematical programming approaches, other solution methods for ERPs include simulation (Zheng and Chen, 2018), Input/Output models (Hofmann et al., 2016), analytical real-option models (Adkins and Paxson, 2017), and portfolio theory (Ahani et al., 2016).

Applications Büyüktahtakın and Hartman (2016) extends the single-asset problem with fixed charge and budget constraints in Büyüktahtakın et al. (2014) by including technological change and assets inventory decisions and apply the proposed ILP model to the case study of the US postal fleet logistics. Their model, solved by `CPLEX` solver and integrated in a Branch&Cut approach, effectively handle large-scale real-world instances exceeding 200,000 vehicles, 60 time periods, and a maximum asset age of 24 years. des-Bordes and Büyüktahtakın (2017) introduce the first multi-objective parallel replacement problem with an application in the US healthcare sector, specifically on magnetic resonance imaging machines. In their ILP models, that is intended to be general and applicable to other assets applications like aircrafts and vehicles, the authors consider multi-asset replacement of multiple challengers for each family, family-specific demand, budget constraints, fixed charge, and capacity change due to technological change and deterioration, with the objective functions of minimizing the discounted penalty cost for not satisfying the demand and the total discounted cost incurred throughout the finite planning horizon. The authors solve a small real-world case study in Kansas (US) by applying the weighted sum method with 11 different weights combinations, providing useful insights to healthcare decision makers who aim to balance penalty and costs. Uncertainty is another key consideration in ERP models. Seif et al. (2019) present a two-stage stochastic ILP model for parallel machine replacement, addressing uncertain planning horizons and demand across time periods with the objective to minimize total expected costs. Their application on excavators shows the importance of considering uncertainty in this and similar ERP applications, and provide managerial insights on the optimal economic life and total utilization level of excavators.

Environmental considerations Recent ERP models integrate environmental considerations, such as greenhouse gas (GHG) emissions and fuel efficiency, into fleet management decisions. Guerrero (2014) explores the impact of fuel-saving technologies on fleet management in the US trucking sector, offering insights for sustainability. A common setting for ERP applications with GHG emissions consideration is to model a regulating emissions trading system, under which the governmental authority gives allowances and permits to emitters, that set how much they can emit in a considered time period, and then emitters can trade (i.e., buy and sell) permits with the authority and among each other. The first one to be established was the European Union Emissions Trading Scheme (EU ETS) in 2005 (Skjærseth and Wettestad, 2016). Ansaripoor et al. (2014) and Ansaripoor and Oliveira (2018) address sustainable fleet replacement problems, optimizing vehicle leasing decisions, including Electric Vehicles (EV), to minimize the weighted average of risk and total expected cost (including CO_2 prices and emissions). Ansaripoor and Oliveira (2018) extend their model to incorporate uncertainties in CO_2 and fuel prices. Optimal leasing policies in the proposed case study include petrol or EVs as the best option depending on the presence of some technological changes, while hybrids are never chosen. Figliozzi et al. (2011) extend the multi-challenger ERP model by Hartman (2004) to include GHG emissions costs, analyzing scenarios based on real-world data from US and GHG cost from the current EU ETS to demonstrate the potential of higher fuel costs or carbon taxes for reducing energy consumption and GHG emissions.

The bus fleet replacement problem is another well-studied area with environmental considerations. Islam and Lownes (2019) present an ILP model for a bus fleet replacement with finite horizon and heterogeneous replacement (e.g., diesel, electric, hybrid buses), optimizing Life Cycle Costs (LCC) including emission costs, with significant reductions achieved through optimized replacement schedules on the case study of the public transportation department in Connecticut (US). Riechi et al. (2017) study a related problem with a case study in Spain for the urban transport fleet. Other studies examine bus fleet replacement problems from bus operators and government perspectives, considering factors such as emissions standards, subsidies, and infrastructure investments for EVs to minimize costs and environmental impacts of bus operations.

Li et al. (2015) solve the case study of public buses in Hong Kong transportation department. Zhou et al. (2021) propose a bi-objective ILP to maximize environmental equity (measured as the elimination of disadvantage of pedestrian population exposed to pollution based on spatial distances) and minimize the total investment costs, and solve a real-world case study on the transit authority of Utah (US) by means of an approximated ϵ-constraint method. Zhou et al. (2023) include external costs (i.e., noise costs, air pollution costs, GHG emissions costs and used battery recycling cost/profit) and incentives/disincentives (i.e., a one-time subsidy, annual subsidy, diesel bus tax and diesel tax) in their ILP model and solve the case study on the public bus transport operator in Singapore. They show how the different subsidies can effectively accelerate the bus fleet electrification.

EVs are introduced in replacement models for various applications. Kleindorfer et al. (2012) study the electrification of the mail and parcel distributors of La Poste, the French national postal operator. Alp et al. (2022) explore fleet electrification and infrastructure investment for freight transportation, considering congestion-related issues and emissions constraints. A numerical experiment demonstrates that electrification is cost-optimal in most scenarios, especially under dense demand conditions, and that it is convenient for a company to optimize electrification and charging infrastructures simultaneously. Stasko and Oliver Gao (2012) study a multi-asset repair/retrofit/replacement problem with stochastic maintenance and repair costs and vehicle failures, under deterministic demand and emissions regulations. The authors propose a stochastic Approximate Dynamic Programming (ADP) approach. A taxi fleet application in China is presented by Xiao et al. (2023). The taxi fleet company aims to maximize the expected total net profit by planning replacement and leasing in the heterogeneous case of multiple types of EV under a set of potential subsidy policies (i.e., Chinese cap-and-trade system) with predicted probability in the future. The authors propose a stochsatic ILP model, with one scenario for each governmental subsidy plan and solve it with `CPLEX` on a case study in China with useful insights on the more efficient EV types. Rajabian et al. (2021) examine ERP under the California and Quebec cap-and-trade markets, optimizing asset decisions while adhering to GHG emissions regulations. The goal is to minimize the sum of all (not discounted) costs, including GHG emissions bought/sold. The authors propose an ILP model and solve it with `CPLEX` on a small-size application on excavators in Ontario (Canada), showing that the model is stable with respect to carbon price, but sensitive to costs and prices. Other ERP applications with environmental concerns include electronic equipment with leasing decisions (Sharma et al., 2007) and private vehicles under scrappage programs in US (Spitzley et al., 2005). Research in this area addresses a wide range of applications, including repair/replacement problems, bus and taxi fleet management, and scrappage programs, reflecting a growing interest in incorporating environmental concerns into ERP models. However, there remains a research gap in addressing the bi-objective perspective of financial and environmental sustainability, testing its effectiveness and scalability on ERP models, and evaluating its validity and applicability in practice. Our study aims to bridge this gap in the existing literature.

7.3 Problem statement

In this work, we first study a specific vehicle replacement problem and then investigate its generalization. In the single-family-single-challenger (SFSC) problem, we are given one family of multiple assets of the same type with different age clusters. In every time period, the replacement of each asset with a new same-type asset must be evaluated to satisfy a constant demand of assets, the maximum age of assets, and budget constraints. Then, we state the multi-family-multi-challenger (MFMC) formulation, that represents the general case of multiple families of assets, each with its set of challenger types for replacement. The problems are bi-objective with total discounted costs minimization and vehicular CO_2 emissions minimization. Although the ultimate goal is to solve the general bi-objective MFMC case in the real case studies, we formulate the two problems as separate ones and present models and algorithms for each of them as distinct contributions.

SFSC problem We are given a set of time periods $j = 0, 1, \ldots, T$ (years), where $j = 0$ is the current time period (i.e., time zero) and $j = T$ is the (finite) time horizon, and a set of n assets owned by a company with a maximum age equal to N years. We assume here that $N < T$ and that assets are all of the same type but have different ages. In particular, we define N subsets of assets given at time zero, called clusters, each made of b_i assets of age i, with $i = 1, 2, \ldots, N$ such that $\sum_{i=1}^{N} b_i = n$. We also assume that (i) in every period, all assets of age N (cluster b_N) must be sold and replaced, and (ii) at the end of the time horizon ($j = T$), all assets must be salvaged and no purchases are allowed. We introduce a capital budgeting constraint u_i for each time period $i = 0, \ldots, T-1$. Every year, the company must decide whether to replace or not each asset of the fleet (i.e., vehicle replacement plan) to minimize the total discounted costs and the total vehicular CO_2 emissions, while maintaining n vehicles every year and not violating budget constraints with buying/selling operations. In the following, we assume that all replacements and outflows occur at the end of each time period (i.e., end of the year). In the total discounted costs to time zero, we consider purchase prices for new vehicles, O&M costs for keeping each vehicle in use every year, and salvage values from salvaged vehicles when they are replaced, with constant discount rate. We define (i) the purchase price of a new asset p_j purchased at the end of period j (assuming that the price of an asset increases over the periods due to continuous technological change and inflation effect), (ii) the O&M cost c_{ij} for keeping an asset in use for the $(i+1)$-th period of age from the end of period j to the end of period $j+1$ (under the assumption of variable costs over the periods, which are decreasing by time due to continuous technological change and non-decreasing by age due to deterioration), and (iii) the salvage value v_{ij} of an i-period old asset salvaged at the end of period j (assuming that salvage values are increasing by time and non-increasing by age under the same assumptions of prices). In the total CO_2 emissions, we consider only the vehicular emissions generated by the use of the vehicles. We define the emission parameters e_{ij} for using the $(i+1)$-th period of age from the end of period j to the end of period $j+1$ (under the same assumptions of O&M costs c_{ij}). The cost discounting effect is modeled by constant discount factor r $(0 \leq r \leq 1)$, while CO_2 emissions are not affected by discount effect.

MFMC problem We are given a set of families $m = 1, \dots, M$ of different assets and a set of multiple challengers (i.e., alternative types of assets valid for replacement) $l = 1, \dots, L_m$ for each family m. We are given a maximum age N_m of assets for each family m. We define initial clusters b_i^{ml} of assets of age i for each type l of the family m at time zero. Demand values are constant for each family m and are defined as $n^m = \sum_{l=1}^{L_m} \sum_{i=1}^{N} b_i^{ml}$ (i.e., the number of assets for each family must remain constant but the mix of asset types can vary for each family over time periods). Parameters p_j^{ml}, c_{ij}^{ml}, and v_{ij}^{ml} indicate purchase prices, operating costs, and salvage values of assets of age $i = 0, \dots, N_m$ in time periods (i.e., years) $j = 0, \dots, T$ for family m and type l with $m = 1, \dots, M$ and $l = 1, \dots, L_m$. Similarly, CO_2 emission parameters are indicated by e_{ij}^{ml}. The capital budget u_j remains on the overall fleet (and not family-specific) for every period $j = 0, \dots, T-1$. Each replacement decision includes (i) when each asset of each family must be replaced and (ii) which challenger to select for each replacement among the set of possible challengers for each family.

7.4 Dynamic programming for the SFSC problems

In the SFSF replacement problem, there is only one challenger available for each asset replacement. The decision is twofold for each asset, that is whether to keep or replace the asset at the end of each period. Since there are multiple assets (all of the same type), a combination of assets may either be kept or replaced. Decisions occur at the end of each time period with all assumptions made in Section 7.3. The states of the dynamic program refer to the state of each cluster of assets, defined by their same age. All assets of maximum age must be replaced with new assets assumed all of age 0. The DP determines optimal keep/replace decisions in each time period. Demand (i.e., total number of assets) and budget constraints hold implicitly in the DP as generated replacement decisions are restricted to being feasible. As we solve a finite horizon problem, a boundary condition is assigned to period $T + 1$ representing the salvage of all assets after the final period T. We propose a forward DP algorithm, in which we are given a set of time periods $j = 0, \dots, T+1$ where $j = 0$ is the starting time period or "time zero", in which the initial condition of the system is defined (number of owned assets b_i of each cluster of age $i = 1, \dots, N$) and $j = T + 1$ is the last time period which we need to include to define the final condition of the system in which all assets salvaged (i.e., empty clusters).

In the representative network of the DP, a node k in period j represents a specific state of the fleet of vehicles, that is how many assets of age i are owned at the end of period j for $i = 1, \dots, N$ and $j = 0, \dots, T+1$, and an arc (k, k') represents the keep/replace decision to move from the state of the fleet corresponding to node k in period j to the state of node k' in period $j + 1$, that is how many assets to sell and replace at the end of period j and how many to keep until the end of period $j + 1$. In our DP, arcs are defined only between pairs of nodes in following periods. We represent the label of each node k in period j with the tuple (j_k, $[b_1, b_2, \dots, b_N]$, $prev_k$, C_{jk}, E_{jk}) where: (i) j_k indicates the time period index of node k; (ii) the list $[b_1, b_2, \dots, b_N]$ represents the state of the DP, defined by the cardinality b_i of each cluster of assets by age i with $i = 1, \dots, N$; (iii) $prev_k$ is the pointer to the predecessor node of k (in period $j - 1$) in the partial path P_k from the origin up to node k; (iv) C_{jk} is the value of total costs (discounted to time 0) of the partial path P_k up to node k in period j; (v) E_{jk} is the value of total CO_2 emissions of the partial path P_k up to node k in period j. The origin of each path is the source node $k = 0$ (i.e., initialization of the graph) with label (0, $[b_1, b_2, \ldots, b_N]$, $None$, 0, 0): it is the only node in time zero with zero costs and emissions and no predecessor (i.e., "$None$"), and the state is represented by the given number of old assets. The DP algorithm works to generate all feasible non-dominated paths from the source node in period $j = 0$ to period $j = T$ and close them in period $j = T + 1$ with the known state $[0, \ldots, 0]$ (i.e., all assets salvaged at the end of the time horizon). Under common cost assumptions (non-decreasing O&M costs and non-increasing salvage values by asset age), it can be shown that the DP solutions, that are the set of optimal paths with respect to the objective functions, are optimal (Hartman, 2004). The non-dominance condition refers here to the concept of Pareto optimality in the presence of two objectives, as we will state later in Section 7.5.3. Figure 7.4.1 represents the first stage of decisions for a representative DP network with maximum age $N = 3$, where we indicate only an abbreviated version of the node labels $([b_1, b_2, \ldots, b_N], C_{jk}, E_{jk})$ since time periods and predecessors are shown by the graph (periods on top and predecessor/successors connected by linking arcs, respectively). In absence of budget constraints and considering a single cost-minimization objective

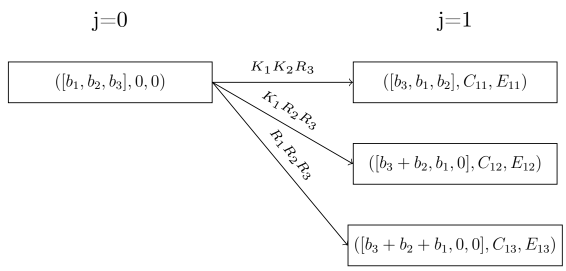


Figure 7.4.1: Network of first-stage decisions without budget constraints.

function, the forward DP algorithm proceeds period by period by: (i) visiting each node k in period j, (ii) extending node k, i.e., generating all the nodes with feasible states at the following period $j + 1$ from the state of node k in period j, (iii) evaluating the additional costs of each new feasible state (ignoring emissions here), and (iv) select only the ones with non-dominated states in the current list of nodes in period $j + 1$ to be appended to the list. In the last step of the algorithm, each node in $j = T$ is extended as a new node in $j = T + 1$ with the only feasible state with all empty clusters $[0, \ldots, 0]$ equivalent to the "all-salvaged" decision, if non-dominated in $j = T + 1$. The dominance function in the case of the single cost-minimization objective is to keep only the minimum-cost label for each feasible state. In $j = T + 1$, since all the states are the same and equal to the state with all empty clusters, the algorithm ends up with only one node with the minimum-cost label. Retrieving the path from this node back to the first period $j = 0$ through the predecessors, we obtain the optimal solution path.

In this procedure, we can assume that the same keep/replace decision is optimal for all the assets of each cluster b_i of age i since they have equivalent characteristics and there is no interdependence between assets. This property, called the no-splitting rule (NSR) in the literature, was proved by Jones et al. (1991). We also know that at each time period, there is only one feasible decision for assets of maximum age $i = N$, which is to sell the assets and buy new ones since they reached the maximum age. All the feasible states can be generated by the combination of different keep/replace decision for each cluster $b_1, \ldots, b_{N-1}$ by taking into account such rules. We can also assume that a second property is valid, which is the older cluster replacement rule (OCRR) by Jones et al. (1991), stating that older clusters are always replaced before younger ones. These two rules guarantee that feasible states for each node can be generated by replacing an increasing number of clusters, from replacing only the maximum-age one to replacing all. We indicate by R_i the decision of "replacing" cluster of age i and by K_i the decision for "keeping" cluster of age i. Then, the complete set of decisions can be indicated as $\{K_1 \ldots K_{N-1}R_N, K_1 \ldots K_{N-2}R_{N-1}R_N, \ldots, K_1R_2 \ldots R_N, R_1 \ldots R_N\}$. A feasible state corresponds to each one of these decisions. In Figure 7.4.1, we observe that for $N = 3$, we have three new nodes in time period $j = 1$, which correspond to the following keep/replace decisions: replace only three-year old asset cluster b_3, which is mandatory because it is the maximum-age cluster ($K_1K_2R_3$), replace 2-year old and three-year old clusters ($K_1R_2R_3$), replace all clusters ($R_1R_2R_3$). These three different decisions correspond to different states (i.e., assets age) and different values of costs C_{jk} and CO_2 emissions E_{jk} of partial paths up to the k-th node in period j.

In addition to NSR and OCRR, we introduce a dominance rule on different states comparison (to be checked in addition to the one on same states comparison) that can further reduce the growth in the state space and is never contradicted in the empirical testing of the algorithm. We call it Younger-and-Cheaper Dominance Rule (YCDR): given two nodes in the same period, if one defines a younger state (i.e., smaller total sum of assets age) with smaller cost than the other, the first dominates the second and the second can be discarded. This rule is based on the idea that in the rest of the path up the end of the horizon, replacing the younger fleet will be cheaper for any decisions set.

Budget constraints If budget constraints are included in the problem, we cannot assume that no cluster is going to split anymore. Thus, it could happen that it is convenient to replace only some units of a cluster to minimize costs and satisfy capital budget constraints. Therefore, the number of feasible states increases significantly. However, as shown by Hartman (2000), we can still apply the NSR and the OCRR under reasonable cost parameters assumptions and assume that at most one cluster is split in any decision, which is the youngest among the ones that are replaced. Thus, we define the number of states which we need to generate as follows: for each node k in period j, we generate the new states by evaluating the replacement of assets up to an increasing number of units of cluster i (with decreasing i from the oldest to the youngest cluster). Using the notation $R_i(a)$ to indicate the replacement of a assets of cluster of age i, the complete set of decisions becomes $\{K_1 \ldots K_{N-1}R_N,\ K_1 \ldots K_{N-2}R_{N-1}(1)R_N,\ K_1 \ldots K_{N-2}R_{N-1}(2)R_N,\ \ldots,\ K_1 \ldots K_{N-2}R_{N-1}(b_{N-1})R_N,\ \ldots,\ R_1 \ldots R_N\}$. For any new feasible state in $j + 1$, we evaluate the additional cost and check if it violates the budget constraint u_{j+1}; if not, we proceed by evaluating the dominance (considering also YCDR) and, if non-dominated, we append the node and proceed to generate new nodes from node k; otherwise, if budget is violated, we can stop generating new nodes from the current one, and move to the following node $k + 1$ (since we know that an increasing number of replacements in the next states is going to be more expensive and therefore, still violating the budget). Figure 7.4.2 shows a partial representation of first-stage decisions in the representative network with $N = 3$ and budget constraints.

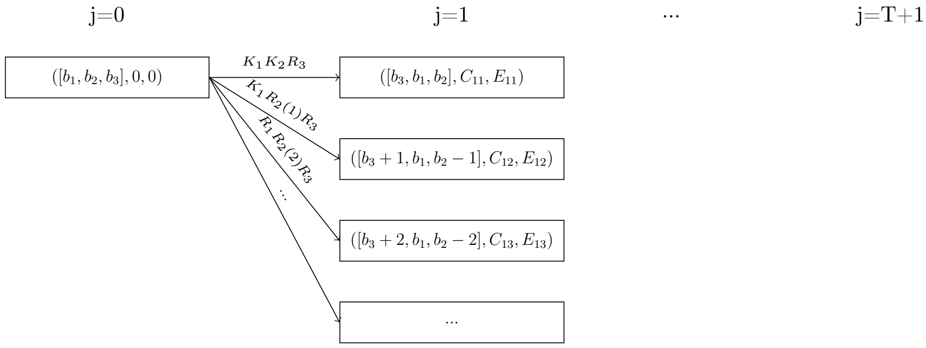


Figure 7.4.2: Network of first-stage decisions with budget constraints.

Bi-objective DP If the second objective function is included, which is the CO_2 emissions minimization, the dominance function must consider the two objectives for any new feasible state generated (after checking the budget, if considered). Note that in the intermediate case of no budget but bi-objective, the way of generating new states is equivalent to the case of budget constraints (i.e., one cluster splitting). This was empirically demonstrated by the comparison with the ϵ-constraint method applied to the ILP formulations (see Section 7.5.3), where the second objective becomes an additional set of budget constraints on emissions, thus introducing the splitting. In the bi-objective case, the number of non-dominated states increases in each stage up to $j = T$ due to the bi-objective dominance function. Consider two states in the same stage generated from different partial paths. If one has lower values for both objectives, then it dominates the other (therefore only the best one is included in the list). Otherwise, if each state is better on one objective (i.e., lower costs but higher emissions or higher costs but lower emissions), the two states are non-dominated and both must be included in the list. We extend here the YCDR to the case of younger-and-cheaper-and-greener dominance rule (YCGDR). In the last step of the algorithm (i.e., from $j = T$ to $j = T + 1$) of the bi-objective case where all assets are salvaged with the same final state for all the nodes (i.e., $[0, \dots, 0]$), multiple paths can be obtained. The set of pairs of costs and emissions values $(C_{T+1,k},\ E_{T+1,k})$ of such complete paths provides a set of points in the criteria space, that is, the Pareto front of optimal costs/emissions solutions.

7.4.1 An illustrative numerical example

We have implemented our DP algorithm in `Python` and preliminarily tested it on illustrative numerical examples. We consider one instance of example with the following input data: the time horizon is $T = 5$, the maximum age is $N = 3$, and the initial state of the fleet is given by $[2, 2, 2]$. Purchase prices are $[10, 12, 15, 20]$ in K€. O&M costs are $[[8, 7, 6, 5], [9, 8, 7, 6], [10, 9, 8, 7], [11, 10, 9, 8]]$ in K€. Salvage values are $[[6, 9, 11, 14, 19], [5, 8, 10, 13, 18], [4, 7, 9, 12, 17]]$ in K€. The discount rate is $r = 0.5$. Budgets are $[12, 12, 1000, 1000]$ in K€, and emission are $[[16, 12, 8, 4], [17, 13, 9, 5], [18, 14, 10, 6], [19, 15, 11, 7]]$ in tons of CO_2.

DP without budget constraints Since we have a bi-objective problem with costs and emissions, we must consider the one-splitting decision enumeration described above. The difference with and without budget will be in the size of the state space: indeed, budget constraint, if tight, can significantly reduce the number of nodes and allow to avoid expanding all the nodes. For graphical reasons, Figure 7.4.3 only shows the nodes of the first two and the final periods of the DP. In the first stage of the DP, five new

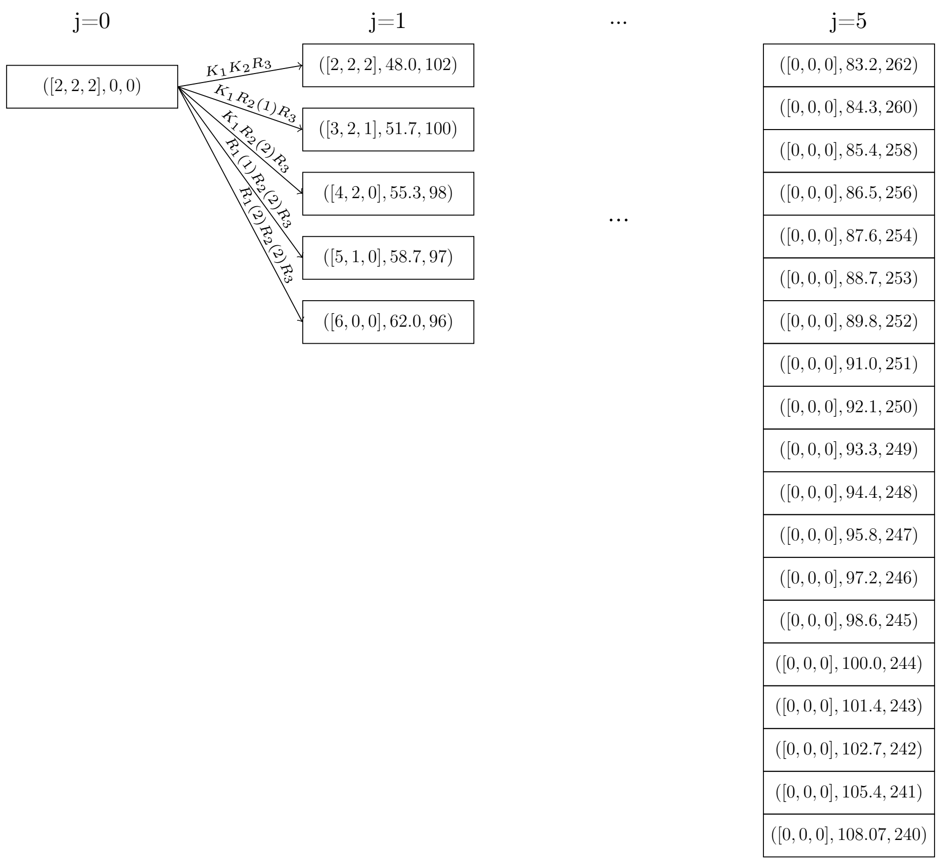


Figure 7.4.3: Illustrative example: DP without budget.

nodes are generated in $j = 1$ from the initial node in $j = 0$. The first node with state $[2, 2, 2]$ represents the decision of replacing the oldest cluster of two assets of age $i = 3$ and keeping the other two clusters, which get older by one more year (i.e., $K_1K_2R_3$). The second node with state $[3, 2, 1]$ represents the decision of replacing one asset of the cluster of age $i = 2$ and the cluster of age $i = 3$ and keeping the rest (i.e., $K_1R_2(1)R_3$). And so on. In every stage of the algorithm, a new node is appended to the new list only if its state is not yet included in the list or if it is non-dominated. New nodes in $j = 2$ are generated from the five nodes in $j = 1$ in the same way and so on. The list of nodes in $j = 5$ represents the 19 final paths which correspond to 19 optimal non-dominated solutions. The state is the same for all nodes and is equal to $[0, 0, 0]$ because we assume to salvage all assets at the end of the time horizon (i.e., in the final stage we only consider the negative costs to salvage all assets).

DP with budget constraints Figure 7.4.4 shows the same steps of the algorithm but in the case where budget constraints are considered. We observe that from $j = 0$ to

$j = 1$ only one node is generated (with state $[2, 2, 2]$) since the budget $u_1 = 12$ makes the other states infeasible (which are generated in the case without budget, as shown in Figure 7.4.3). At the final stage of the DP, only nine nodes are obtained, which

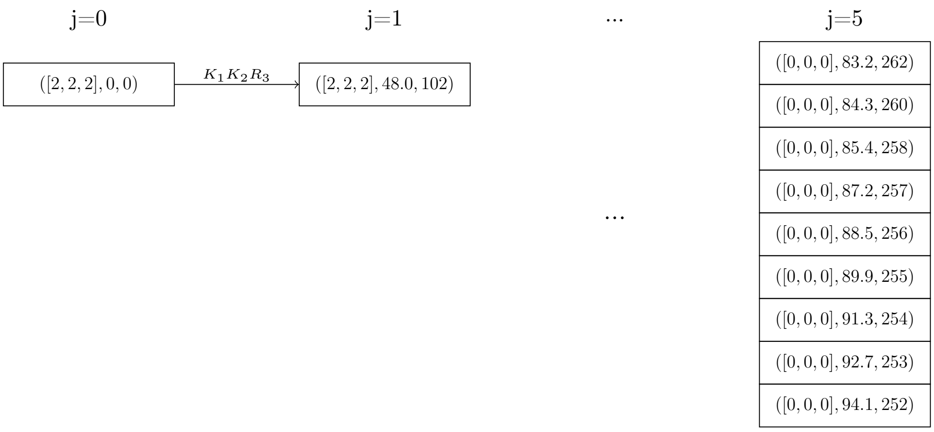


Figure 7.4.4: Illustrative example: DP with budget.

correspond to nine optimal non-dominated solutions. This numerical example shows that tight budget constraints can reduce the size of the DP network. The nine points (corresponding to total costs and CO_2 of the optimal paths) are represented in the criteria space, as shown in Figure 7.4.5, that graphically represents the Pareto front of optimal solutions of the bi-objective example.

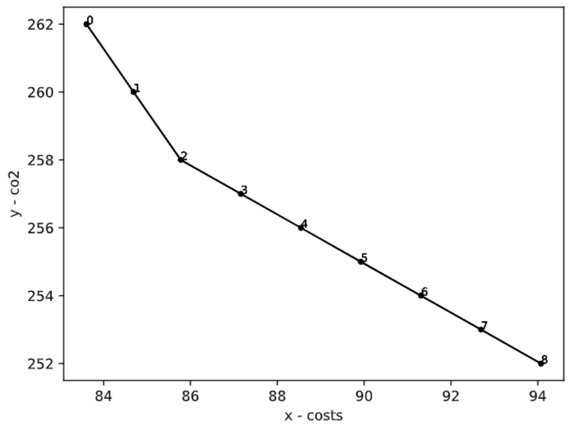


Figure 7.4.5: Illustrative example: Pareto front of optimal solutions.

The DP can be generalized to the case of multiple challengers and multiple families (i.e., MFMC problems), by introducing a matrix notation for the DP states with one vector for each challenger types, and exploring one DP table for each family of assets. However, the number of states would increase exponentially due to the multiple replacement decisions for each asset (not twofold keep/replace decision anymore), resulting in a larger representative DP network (see Bellman, 1957). The interdependence between families of assets due to overall budget constraints would further complicate the enumeration of feasible states and the definition of valid reduction rules. We leave these extension for future work (see Section 7.8).

7.5 Mathematical models for the SFSC problem

In this section, we explore the mathematical formulations and solution method for the SFSC bi-objective problem. We present a network flow ILP model (Section 7.5.1) and an alternative ILP model (Section 7.5.2). Then, we describe the bi-objective optimization approach applied to the models, that is the ϵ-constraint method (Section 7.5.3).

7.5.1 Network flow model

We define a directed network $G = (V, A)$ where the set of nodes V is given by (i) a "super" node s, which is the source node of the network supplying n units of flow (i.e., total number of assets owned at time zero), (ii) N extra nodes $s_1, \dots, s_N$, which receive the flow from the source node (i.e., i-period old assets owned at time zero with $i = 1, \dots, N$), and (iii) the set of nodes $j = 0, 1, 2, \dots, T$ representing the time periods considered in the finite time horizon. The complete set of nodes is $V = \{ s, s_1, \dots, s_N, 0, 1, \dots, T \}$. Three types of arcs are included in the set A of arcs. Any arc (s, s_i) indicates the source flow of i-period old assets owned at time zero with $i = 1, \dots, N$. Any arc (s_i, j) is a defender arc representing the decision of keeping some of i-period old assets owned at time zero until period j, with $i = 1, \dots, N$ and $j = 0, \dots, N - i$. Arcs (j, k) are the challenger arcs representing the decision of purchasing some new assets in year j and keeping them in use until year k with $j = 0, \dots, T - 1$ and $k = j + 1, \dots, min(N + j, T)$.

Cost and CO_2 emission parameters formulation We assign a unit cost (i.e., net cash outflow) $f_{jk} \geq 0$ [€] to every arc (j, k) considering (i) the purchase price of a new asset p_j purchased at the end of period j (discounted to time 0), (ii) the sum of O&M costs, where c_{ij} is the cost for keeping the asset in use for the $(i + 1)$-th period of age from the end of period j to the end of period $j + 1$ (discounted to time 0), and (iii) the salvage value v_{ij} of an asset i-period old salvaged at the end of period j (discounted to time 0). We assume valid the assumptions on cost trends made in Section 7.3. Given the discount rate r, the total discounted cost of each type of arc is the following:

$$f_{jk} = \begin{cases} 0 & \forall (s, s_i) \in A, 1 \leq i \leq N \\ -v_{i0} & \forall (s_i, 0) \in A, 1 \leq i \leq N \\ \displaystyle\sum_{l=0}^{j-1} \frac{c_{(i+l)l}}{(1+r)^{l+1}} - \frac{v_{(i+j)j}}{(1+r)^j} & \forall (s_i, j) \in A, 1 \leq i \leq N, \\ & \quad 0 < j \leq N - i \\ \displaystyle\frac{p_j}{(1+r)^j} + \sum_{l=j}^{k-1} \frac{c_{(l-j)l}}{(1+r)^{l+1}} - \frac{v_{(k-j)k}}{(1+r)^k} & \forall (j, k) \in A, 0 \leq j \leq T - 1, \\ & \quad j + 1 \leq k \leq \min(N + j, T) \end{cases} \tag{7.1}$$

In particular, each arc between super source and source nodes has no cost. Arcs from source nodes to time zero has negative cost from salvaging each asset immediately. Defender arcs have total costs from operating assets and then salvaging them. Challenger arcs have total costs for purchasing, using, and then salvaging assets.

We also assign a unit emission parameter $g_{jk} \geq 0$ [kg CO_2] to every defender/challenger arc (j, k) considering only the sum of transport emissions, where e_{ij} is the amount of

CO_2 vehicular emissions of an asset in the $(i+1)$-th period of age from the end of period j to the end of period $j+1$. No discount is applied to CO_2 emissions. Figure 7.5.1 represents a problem with $T = 4$ and $N = 3$ with source and defender arcs represented as dotted and dashed curved arrows, respectively, while challenger arcs are represented as black curved arrows.

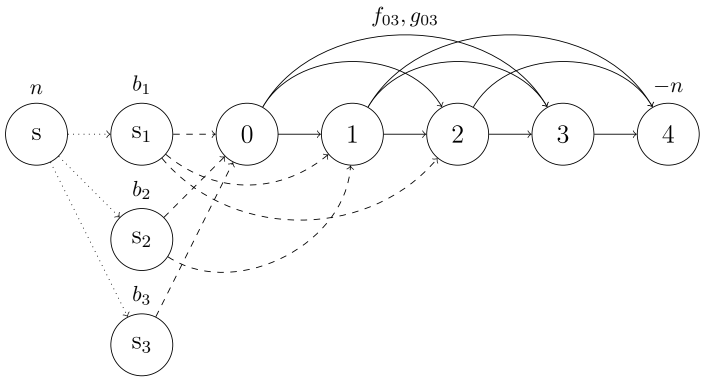


Figure 7.5.1: Graphical representation of the network flow model with $T = 4$ and $N = 3$.

Mathematical model We introduce a set of two-index non-negative integer variables x_{jk} that indicate the number of assets assigned to arc (j,k) in the solution path P with $(j,k) \in A$ and formulate our bi-objective SFSC parallel replacement problem as follows:

$$\min Z_C = \sum_{(j,k)\in A} f_{jk} x_{jk} \tag{7.2}$$

$$\min Z_E = \sum_{(j,k)\in A} g_{jk} x_{jk} \tag{7.3}$$

subject to:

$$\sum_{(j,k)\in\delta^+(j)} x_{jk} - \sum_{(k',j)\in\delta^-(j)} x_{k'j} = \begin{cases} n & \text{if } j = s \\ -n & \text{if } j = T \\ 0 & \text{otherwise} \end{cases} \tag{7.4}$$

$$x_{ss_i} = b_i \quad i = 1, \dots, N \tag{7.5}$$

$$\sum_{(j,k)\in\delta^+(j)} p_j x_{jk} - \sum_{\substack{(k',j)\in\delta^-(j)\\ 0\le k'<j}} v_{(k'-j)j} x_{k'j} - \sum_{(s_i,j)\in\delta^-(j)} v_{(i+j)j} x_{s_ij} \le u_j \tag{7.6}$$

$$j = 0, \dots, T-1$$

$$x_{jk} \ge 0, \quad x_{jk} \in \mathbb{Z} \quad (j,k) \in A \tag{7.7}$$

Equations (7.2) and (7.3) represent the objective functions that minimize the total discounted costs to time 0 and the total CO_2 emissions, respectively. Constraints

(7.4) guarantee the conservation of flow on source, sink, and transshipment nodes. Constraints (7.5) are the capacity constraints on the arcs that exit from the source (i.e., asset owned at time zero), while equations (7.6) represent the budget constraints. Finally, (7.7) state the integer domain of the variables. Note that model (7.2)-(7.3) subject to (7.4)-(7.5) is a Shortest Path Problem (SPP), which is polynomially solvable (Bradley et al., 1977) therefore, the integrality of the variables could be maintained also relaxing constraints (7.7) (i.e., defining continuous non-negative variables) in the absence of budget constraints. However, budget constraints (7.6) make the problem a resource-constrained SPP (Pugliese and Guerriero, 2013), for which the integrality of the solution variables is not guaranteed anymore.

7.5.2 Alternative integer programming model

We present an alternative ILP formulation, which is known to be efficient in the replacement literature and is an adaptation of the formulation by Hartman (2000) in which we remove inventory options and fixed charge on purchases and include a second objective function. We keep the same notation introduced in Section 7.5.1. We define three sets of non-negative integer variables: variables B_j indicates the number of new assets purchased at the end of period j ($j = 0, \dots, T-1$); variables X_{ij} indicates the number of i-period old assets kept in use from the end of period j to the end of period $j+1$ ($i = 0, \dots, N-1$, $j = 0, \dots, T-1$); variables S_{ij} indicates the number of i-period old assets salvaged at the end of period j ($i = 1, \dots, N$, $j = 0, \dots, T$). We formulate the bi-objective SFSC parallel replacement problem as follows:

$$\min Z_C = \sum_{j=0}^{T-1} \frac{p_j B_j}{(1+r)^j} + \sum_{i=0}^{N-1}\sum_{j=0}^{T-1} \frac{c_{ij} X_{ij}}{(1+r)^{j+1}} - \sum_{i=1}^{N}\sum_{j=0}^{T} \frac{v_{ij} S_{ij}}{(1+r)^j} \tag{7.8}$$

$$\min Z_E = \sum_{i=0}^{N-1}\sum_{j=0}^{T-1} e_{ij} X_{ij} \tag{7.9}$$

$$\text{s.t.} \quad B_j = \sum_{i=1}^{N} S_{ij} \quad j = 0, \dots, T-1 \tag{7.10}$$

$$B_j = X_{0j} \quad j = 0, \dots, T-1 \tag{7.11}$$

$$X_{i0} + S_{i0} = b_i \quad i = 1, \dots, N-1 \tag{7.12}$$

$$S_{N0} = b_N \tag{7.13}$$

$$X_{ij} + S_{ij} = X_{(i-1)(j-1)} \quad i = 1, \dots, N-1, \quad j = 1, \dots, T-1 \tag{7.14}$$

$$S_{iT} = X_{(i-1)(T-1)} \quad i = 1, \dots, N \tag{7.15}$$

$$S_{Nj} = X_{(N-1)(j-1)} \quad j = 1, \dots, T-1 \tag{7.16}$$

$$p_j B_j - \sum_{i=1}^{N} v_{ij} S_{ij} \le u_j \quad j = 0, \dots, T-1 \tag{7.17}$$

$$B_j, X_{ij}, S_{ij} \geq 0, \quad B_j, X_{ij}, S_{ij} \in \mathbb{Z} \tag{7.18}$$

Equations (7.8) and (7.9) represent the objective functions that minimize the total discounted costs to time 0 and the total CO_2 emissions, respectively. In this formulation, we directly include the discounting formula for total costs. Constraints (7.10)-(7.16) include the flow balance constraints: equations (7.10) guarantee that the number of assets in the systems is constant (in every period, what comes in comes out); equations (7.11) impose to keep for one period the new purchased assets; equations (7.12) state that initial assets are either salvaged or kept for one year, while equation (7.13) ensures that initial assets of maximum age are immediately salvaged; equations (7.14) state that for every period excluding the first and the last, assets are either salvaged or kept for one period, while equations (7.15) guarantee that all assets are salvaged in the last period; equations (7.16) impose that, in every period, assets of maximum age are salvaged. Equations (7.17) represent the budget constraints for every period until the last. Finally, equations (7.18) state the integer domain of the variables.

The two-dimensional network representation has age on rows and time periods on columns. Figure 7.5.2 represents a problem with time horizon $T = 4$ and maximum age $N = 3$.

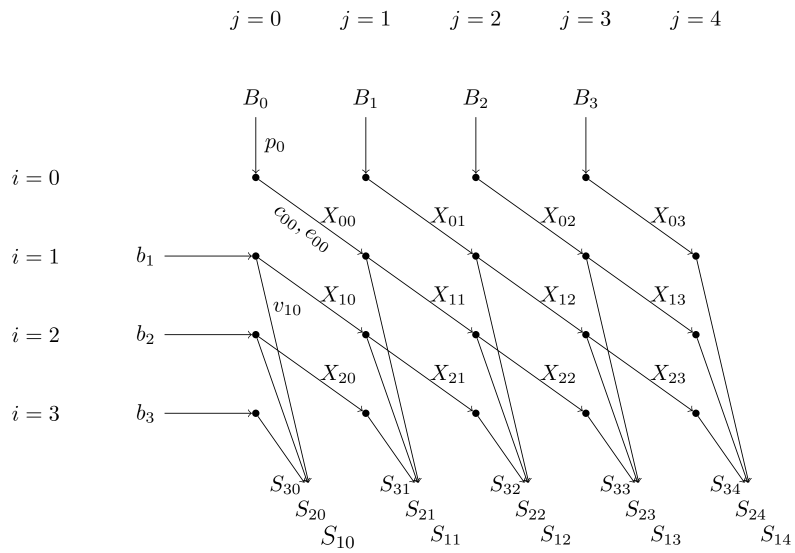


Figure 7.5.2: Graphical representation of the alternative model with $T = 4$ and $N = 3$.

7.5.3 ϵ-constraint method

To solve our bi-objective ILP models, we apply the ϵ-constraint method. Our aim is to provide the complete Pareto front of optimal solutions for costs and CO_2 emissions, from which the decision maker can choose one solution by observing the position of its corresponding point in the space of criteria and evaluating its quality with respect to the other optimal solutions.

In the case of the network flow model, we iteratively solve (7.2) subject to (7.4)-(7.7) with an additional constraint which limits the total quantity of vehicular CO_2 emissions to be, at each iteration q, lower than the value of CO_2 emissions E_{q-1} obtained by the optimal solution at the precedent iteration $q-1$. The additional constraint for the q-th iteration of the ϵ-constraint method is the following:

$$\sum_{(j,k)\in A} g_{jk}x_{jk} \leq E_{q-1} - \epsilon. \tag{7.19}$$

The algorithm stops once the first infeasible solution is found (i.e., too restrictive budget on emissions). Assuming integer values of CO_2 emissions parameters g_{jk} (for example, in tons of CO_2) and having integer variables x_{jk}, we can obtain the complete Pareto front with $\epsilon = 1$, or an approximated Pareto front with greater integer values of ϵ (i.e., finding some Pareto optimal solutions). In most practical applications, finding approximated fronts of solutions is convenient.

In the analogous ϵ-constraint method for the alternative model, we iteratively solve (7.8) subject to (7.10)-(7.18) with the following additional constraint which limits the quantity of total CO_2 emissions at iteration q:

$$\sum_{i=0}^{N-1}\sum_{j=0}^{T-1} e_{ij}X_{ij} \leq E_{q-1} - \epsilon, \tag{7.20}$$

updating the value E_{q-1} in the right-hand side at each iteration q.

7.6 Mathematical models for the MFMC problem

In this section, we present the MFMC extensions of the network flow ILP model (Section 7.6.1), the alternative ILP model (Section 7.6.2), and the ϵ-constraint method (Section 7.6.3). For the notation, we refer to Section 7.3.

7.6.1 Network flow model

We extend the definition of costs and emissions on network arcs (Section 7.5.1) to the MFMC setting, defining parameters f_{jk}^{ml} and g_{jk}^{ml} on each arc (j,k) for family m and type l. We introduce a set of four-index non-negative integer variables x_{jk}^{ml} that indicate the number of l-th type of assets belonging to family m assigned to arc (j,k) in the solution path P with $(j,k) \in A$, $m = 1,\ldots,M$, $l = 1,\ldots,L_m$ and formulate the bi-objective MFMC problem as follows:

$$\min Z_C = \sum_{m=1}^{M}\sum_{l=1}^{L_m}\sum_{(j,k)\in A} f_{jk}^{ml}x_{jk}^{ml} \tag{7.21}$$

$$\min Z_E = \sum_{m=1}^{M}\sum_{l=1}^{L_m}\sum_{(j,k)\in A} g_{jk}^{ml}x_{jk}^{ml} \tag{7.22}$$

$$\text{s.t.} \sum_{l=1}^{L_m}\sum_{(j,k)\in\delta^+(j)} x_{jk}^{ml} - \sum_{l=1}^{L_m}\sum_{(k',j)\in\delta^-(j)} x_{k'j}^{ml} = \begin{cases} n^m & \text{if } j = s \\ -n^m & \text{if } j = T \\ 0 & \text{otherwise} \end{cases} \quad m = 1,\ldots,M \tag{7.23}$$

$$\sum_{(j,k)\in\delta^+(j)} x_{jk}^{ml} - \sum_{(k',j)\in\delta^-(j)} x_{k'j}^{ml} = 0 \quad j = s_i, \quad i = 1, \dots N, \qquad m = 1, \dots, M, \quad l = 1, \dots, L_m \tag{7.24}$$

$$\sum_{m=1}^{M}\sum_{l=1}^{L_m}\left(\sum_{(j,k)\in\delta^+(j)} p_j x_{jk} - \sum_{\substack{(k',j)\in\delta^-(j)\\ 0\le k'<j}} v_{(k'-j)j} x_{k'j} - \sum_{(s_i,j)\in\delta^-(j)} v_{(i+j)j} x_{s_i j}\right) \le u_j \qquad j = 0, \dots, T-1 \tag{7.25}$$

$$x_{ss_i}^{ml} = b_i^{ml} \quad i = 1, \dots, N, \quad m = 1, \dots, M, \quad l = 1, \dots, L_m \tag{7.26}$$

$$x_{jk}^{ml} \ge 0, \quad x_{jk}^{ml} \in \mathbb{Z} \quad (j,k) \in A, \quad m = 1, \dots, M \quad l = 1, \dots, L_m \tag{7.27}$$

Equations (7.21) and (7.22) represent the objective functions to minimize the total discounted costs to time 0 and the total CO_2 emissions. Constraints (7.23) guarantee the flow conservation on source, sink, and transhipment nodes for every family of assets. Constraints (7.24) guarantee the flow conservation for defender nodes (i.e., one can only sell and not buy). Constraints (7.25) are the budget constraints. Constraints (7.26) are the capacity constraints on the arcs that exit from the source (assets owned at time zero) for every asset type of every family, while equations (7.27) state the integer domain of the variables.

7.6.2 Alternative integer programming model

We introduce three sets of non-negative integer variables for each type l of each family m: variables B_j^{ml} indicate the number of new assets purchased at the end of period $j = 0, \dots, T-1$; variables X_{ij}^{ml} indicate the number of i-period old assets kept from the end of period j to the end of period $j+1$ ($i = 0, \dots, N_m - 1$, $j = 0, \dots, T-1$); variables S_{ij}^{ml} indicates the number of i-period old assets salvaged at the end of period j ($i = 1, \dots, N_m$, $j = 0, \dots, T$). We formulate the MFMC problem as follows:

$$\min Z_C = \sum_{m=1}^{M}\sum_{l=1}^{L_m}\left(\sum_{j=0}^{T-1}\frac{p_j^{ml}B_j^{ml}}{(1+\alpha)^j} + \sum_{i=0}^{N_m-1}\sum_{j=0}^{T-1}\frac{c_{ij}^{ml}X_{ij}^{ml}}{(1+\alpha)^{j+1}} - \sum_{i=1}^{N_m}\sum_{j=0}^{T}\frac{v_{ij}^{ml}S_{ij}^{ml}}{(1+\alpha)^j}\right) \tag{7.28}$$

$$\min Z_E = \sum_{m=1}^{M}\sum_{l=1}^{L_m}\sum_{i=0}^{N_m-1}\sum_{j=0}^{T-1} e_{ij}^{ml}X_{ij}^{ml} \tag{7.29}$$

$$\text{s.t.} \sum_{l=1}^{L_m} B_j^{ml} = \sum_{l=1}^{L_m}\sum_{i=1}^{N_m} S_{ij}^{ml} \quad j = 0, \dots, T-1, \quad m = 1, \dots, M \tag{7.30}$$

$$B_j^{ml} = X_{0j}^{ml} \quad j = 0, \dots, T-1, \quad m = 1, \dots, M, \quad l = 1, \dots, L_m \tag{7.31}$$

$$X_{i0}^{ml} + S_{i0}^{ml} = b_i^{ml} \; i = 1, \dots, N_m - 1, \; m = 1, \dots, M, \; l = 1, \dots, L_m \tag{7.32}$$

$$S_{N_m 0}^{ml} = b_{N_m}^{ml}, \quad m = 1, \dots, M, \quad l = 1, \dots, L_m \tag{7.33}$$

$$X_{ij}^{ml} + S_{ij}^{ml} = X_{(i-1)(j-1)}^{ml} \quad i = 1, \dots, N_m - 1,\ j = 1, \dots, T-1, \quad m = 1, \dots, M,\ l = 1, \dots, L_m \tag{7.34}$$

$$S_{iT}^{ml} = X_{(i-1)(T-1)}^{ml} \quad i = 1, \dots, N_m,\ k = 1, \dots, M,\ l = 1, \dots, L_m \tag{7.35}$$

$$S_{N_m j}^{ml} = X_{(N_m-1)(j-1)}^{ml} \quad j = 1, \dots, T-1,\ m = 1, \dots, M,\ l = 1, \dots, L_m \tag{7.36}$$

$$\sum_{m=1}^{M} \sum_{l=1}^{L_m} \left(p_j^{ml} B_j^{ml} - \sum_{i=1}^{N_m} v_{ij}^{ml} S_{ij}^{ml} \right) \leq u_j \quad j = 0, \dots, T-1 \tag{7.37}$$

$$B_j^{ml}, X_{ij}^{ml}, S_{ij}^{ml} \geq 0, \quad B_j^{ml}, X_{ij}^{ml}, S_{ij}^{ml} \in \mathbb{Z} \tag{7.38}$$

Equations (7.28) and (7.29) represent the objective functions to minimize the total discounted costs to time 0 and the total CO_2 emissions, respectively. Constraints (7.30)-(7.36) include the flow balance constraints analogous to the SFSC model (7.8)-(7.18) but with families and types: the demand of assets in the system is constant over period for each family, but the mix of challenger types can changed over periods. Equations (7.37) represent the budget constraints and equations (7.38) state the integer domain of all the variables.

7.6.3 ϵ-constraint method

Analogously to the methodology applied to SFSC models (see Section 7.5.3), we apply the ϵ-constraint method to the bi-objective MFMC network flow model (7.21)-(7.27) with equation:

$$\sum_{m=1}^{M} \sum_{l=1}^{L_m} \sum_{(j,k) \in A} g_{jk}^{ml} x_{jk}^{ml} \leq E_{q-1} - \epsilon, \tag{7.39}$$

and to the alternative MFMC model (7.28)-(7.38) with equation:

$$\sum_{m=1}^{M} \sum_{l=1}^{L_m} \sum_{i=0}^{N_m-1} \sum_{j=0}^{T-1} e_{ij}^{ml} X_{ij}^{ml} \leq E_{q-1} - \epsilon. \tag{7.40}$$

7.7 Preliminary computational results

In this section, we preliminariliy study the computational performance of the proposed methods on randomly generated instances. All algorithms were coded in `Python 3.10.4`. The ILP models were solved with `Gurobi 10.0.2`. The tests were executed on a single thread of a virtual machine Intel(R) Xeon(R) Gold with 2.30 GHz and 20 GB of RAM memory, running under Windows 10 Pro N. A time limit of 3600 seconds per instance was imposed to each algorithm. We first investigate the impact of key instance features such as the size and the rounding parameter on cost and CO_2 emission values in the mono-objective SFSC problem to test the scalability of DP and ILP models. We then explore the behavior of bi-objective DP and ϵ-constraint method on both ILP models in the SFSC problem.

To test the effectiveness and scalability of the models and solution methods, we generate random instances based on existing literature (Büyüktahtakın et al., 2014) and certain parameters from the real case studies. In the first experiment, we examine five different combinations with increasing size, determined by the number of time periods (T), maximum age (N), total number of assets (n), and the following list of initial clusters per age class (b): (10, 2, 20,[20, 0]), (15, 4,40,[20, 15, 5, 0]), (20, 6,60, [30,15, 10,5, 0, 0]), (30, 8, 80,[30, 20, 10, 15, 5, 0, 0, 0]), and (50,10,100, [40, 20, 10,15, 10,5, 0, 0,0, 0]). For each combination, we generate 10 random instances, for a total of 50 instances. We generate random cost and emission functions as follows: at time zero, values are randomly selected within the ranges of [9,000, 11,000] for purchase price p_0 (in €), [900, 1,000] for O&M cost c_{00} (in €), and [9,000, 10,000] for vehicular emissions e_{00} (in kgCO_2). Purchase prices p_j are then generated to linearly increase over time, O&M costs and CO_2 emissions decrease over time and being non-decreasing with age. Salvage values (initially valued at v_{j0}) are defined by a 25% decrease in purchase price (p_j) and being non-increasing with age. We also ensure that the sum of O&M costs and salvage values is non-decreasing with age as required by OCRR (Jones et al., 1991). Discount rate is set to the constant value $r = 0.5$. Budgets are defined by $u_j = np_j/N$ for each period j, where n is the total number of assets, p_j is the purchase price of a unit asset in time period j, and N is the maximum age.

In each table, the first sets of columns indicates the experiment setting and instance features, while the following sets report the results on each algorithm. In Table7.7.1, "ILP1" refers to model (7.2), (7.4)-(7.7), "ILP2" refers to model (7.8), (7.10)-(7.18), and "DP" refers to the mono-objective DP with budget constraints in Section 7.4. The table shows the results on three different experiments determined by the unit rounding parameter R applied on costs coefficients (in objectives and constraints) to round up float values of discounted costs and decimal budgets to integer values. We test three cases with $R = 1, 10, 100$. For each method, we report the number of optimal solutions found within the time limit ("#opt"), the average optimal value of the objective function, that is the total discounted costs ("Obj.(€)"), and the average CPU time in seconds computed only on the instances solved to optimality ("t(s)") within the time limit of 3600s. For DP, we also report the total number of nodes generated by the algorithm ("#nodes"). Each row reports the average results on 10 instances and the "Tot" row reports the overall average on each experiment. We observe that ILP models solve every instance to optimality in less than one second, while the number of nodes in DP grows significantly as the size of instances increases, making larger instances more difficult to solve. This is in line with the ERP literature (see, e.g., Büyüktahtakın et al., 2014). DP with rounding $R = 1$ and $R = 10$ is able to solve to optimality random instances with up to $T = 15$ and $N = 4$ within the time limit of 3600s. However, we observe that in the third experiment ($R = 100$) DP solves all instances to optimality within the same time limit. On the other hand, the impact of rounding on the model coefficients (i.e., total discounted costs) must be considered, possibly testing different rounding functions and evaluating the trade-off between the scalability of DP and the accuracy of the real values (in units of €) estimations.

In our second set of experiments, we investigate the performance of our algorithms on the bi-objective SFSC problem. Table 7.7.2 shows the average performance of the ϵ-constraint method applied to ILP1 and ILP2 and the bi-objective DP in comparison on the five sets of random instances with $R = 100$ and one-hour time limit. For each

Table 7.7.1: Summary of results on mono-objective SFSC methods.

	Parameters				ILP1			ILP2			DP			
R	T	N	n	#	#opt	Obj.(€)	t(s)	#opt	Obj.(€)	t(s)	#opt	Obj.(€)	t(s)	#nodes
	10	2	20	10	10	94,642	0.001	10	94,642	0.001	10	94,642	0.020	2K
	15	4	40	10	10	123,120	0.002	10	123,120	0.001	10	123,120	1151.600	3401K
1	20	6	60	10	10	151,914	0.007	10	151,914	0.002	0	-	-	-
	30	8	80	10	10	186,880	0.014	10	186,880	0.015	0	-	-	-
	50	10	100	10	10	218,815	0.023	10	218,815	0.022	0	-	-	-
Tot*					50	155,074	0.009	50	155,074	0.008	20	108,881	575.808	1702K
	10	2	20	10	10	95,480	0.001	10	95,480	0.001	10	95,480	0.028	2K
	15	4	40	10	10	126,075	0.004	10	126,075	0.002	10	126,075	246.639	2245K
10	20	6	60	10	10	158,140	0.008	10	158,140	0.003	0	-	-	-
	30	8	80	10	10	202,045	0.019	10	202,045	0.012	0	-	-	-
	50	10	100	10	10	256,461	0.043	10	256,461	0.021	0	-	-	-
Tot*					50	167,640	0.015	50	167,640	0.008	20	110,778	123.334	1123K
	10	2	20	10	10	105,200	0.001	10	105,200	0.001	10	105,200	0.021	2K
	15	4	40	10	10	158,800	0.003	10	158,800	0.002	10	158,800	21.057	487K
100	20	6	60	10	10	237,700	0.009	10	237,700	0.002	10	237,700	116.947	1181K
	30	8	80	10	10	379,600	0.014	10	379,600	0.011	10	379,600	369.470	2336K
	50	10	100	10	10	657,350	0.033	10	657,350	0.021	10	657,350	1031.150	4714K
Tot*					50	307,730	0.012	50	307,730	0.007	50	307,730	307.729	1744K

* Totals are computed on instances solved to optimality for each method.

method, we report the average computing time ("t(s)") We observe that DP solves

Table 7.7.2: Summary of results on bi-objective SFSC methods.

Parameters				ILP1	ILP2	DP
T	N	n	#	t(s)	t(s)	t(s)
10	2	20	10	0.466	0.459	11.765
15	4	40	10	20.92	16.562	-
20	6	60	10	74.439	59.625	-
30	8	80	10	139.453	95.156	-
50	10	100	10	287.614	231.025	-

to optimality only the first 10 instances while ϵ-constraint works very well with both ILP models, being able to find the full set of optimal solutions, including the complete Pareto front of efficient solutions, within the time limit. Note that we applied a post-processing procedure to select the set of Pareto-efficient solutions for ILP models (that was preliminarily shown to be faster than optimizing the second objective for each first-objective optimization and finding only Pareto-efficient solutions during the iterative process). On the other hand, the dominance function of the bi-objective DP guarantees (when computationally possible) to reach the time horizon with all and only efficient solutions.

In this experiment, the Pareto front is a dense set of points representing many optimal solutions very close to each other in the cost/emission space and therefore not straightforward to be interpreted (i.e., to select the most desirable optimal solution). Table 7.7.3 shows the details on the Pareto fronts of the first 10 instances. Columns "Min Cost", "Min Em.", and, "#eff" show, respectively, the value of costs (in €) and emissions (in kgCO_2) of the two extreme solutions of minimum costs and minimum emissions, and the number of efficient solutions in the Pareto front. Then, the total optimization time (in s) is given for each of the three methods. In the smaller set of instances with $T = 10$, $N = 2$, and n =20, the average number of Pareto-efficient

Table 7.7.3: Illustrative results on bi-objective SFSC methods.

	Pareto front			ILP1	ILP2	DP
Instance	Min Cost (€;kgCO_2)	Min Em. (€;kgCO_2)	#eff	t(s)	t(s)	t(s)
1	(104,000;2,444,000)	(192,000;2,174,000)	101	0.35	0.36	6.46
2	(98,000;2,136,000)	(178,000;1,902,000)	141	0.74	0.71	9.24
3	(100,000;2,344,000)	(182,000;2,084,000)	101	0.36	0.19	6.29
4	(108,000;2,432,000)	(202,000;2,162,000)	101	0.35	0.33	7.74
5	(114,000;2,040,000)	(210,000;1,814,000)	179	0.72	0.93	29.62
6	(110,000;2,074,000)	(198,000;1,844,000)	101	0.31	0.16	7.01
7	(110,000;2,260,000)	(198,000;2,010,000)	101	0.39	0.3	6.67
8	(100,000;2,172,000)	(182,000;1,932,000)	101	0.3	0.39	6.67
9	(104,000;2,200,000)	(196,000;1,956,000)	121	0.38	0.51	11.4
10	(104,000;2,274,000)	(192,000;2,022,000)	178	0.76	0.71	26.55

solutions in the Pareto front is 123, with an average reduction of emissions of 248 tons of CO_2 between the two extreme solutions. The average trade-off is a maximum emission reduction of 12% in emissions at the expense of a 83% increase in costs. In larger sets of instances, the average number of Pareto-efficient solutions increases up to 2,000 for the larger set. Clearly, this results shows the limitations of the SFSC approach, in which only technological change, inflation, and deterioration effects are considered within the optimization. The MFMC problem allows the replacement choice between defenders and different challengers, that in our problem represent different types of vehicles including greener technologies with different financial and environmental coefficients. Therefore, better trade-offs are expected from the application of MFMC models.

Different approaches may be developed to improve the algorithms. To provide readable subsets of Pareto fronts in shorter time, ILP-based heuristic bi-objective methods, such as approximated ϵ-constraint (e.g., by searching for optimal solutions starting from a half-way goal in the costs/emissions space of solutions and expanding in both minimum-cost and minimum-emission directions or by detecting a reasonable distance between solutions and set ϵ values accordingly), represent a promising approach to solve larger instances effectively. On the other hand, heuristic reduction rules applied on DP may help to reduce the state space and provide a valid alternative for approximating the Pareto front of optimal solutions.

7.8 Concluding remarks and future work

In this work, we study a bi-objective parallel vehicle replacement problem with finite horizon, constant demand of assets over time periods, and budget constraints. The goals are the minimization of the total discounted costs, including purchasing costs for new vehicles, O&M costs for vehicles in use, and salvage values for replaced vehicles, and the minimization of the total vehicular CO_2 emissions. The bi-objective approach which integrates and trades off financial and environmental sustainability by including the direct CO_2 emissions generated by vehicles in the objectives is new to the vehicle replacement literature but crucial for modeling more sustainable transportation systems. The problem is inspired by two real-world applications of Italian companies that every year must evaluate the state of their heterogeneous fleets of vehicles. We have studied the specific SFSC problem with one family of assets and same-type replacement, where the decision is whether and when to replace each asset to obtain the optimal cost/emission replacement plan, and the more general MFMC problem, where there are multiple families of assets and the decision is on optimal replacement periods and types of vehicles to obtain the best vehicle mix and plan for each period. We have presented a DP approach for the SFSC problem and two alternative ILP approaches for the two problems. From the preliminary computational experiments, we observe that the DP algorithm is effective on small-size instances while struggles with dimensionality issues when the problem size increases. The approach may be improved by new reduction rules or if embedded in a heuristic procedure to avoid exploring all the states but providing a good approximation of the Pareto front of solutions. Extending the DP to the MFMC case would require alternative approaches like heuristic methods or column generation. The ϵ-constraint method applied to our ILP models is a promising solution method. For practical applications, an approximated ϵ-constraint method could be useful for the vehicle replacement decision makers. As a future development, we intend to further test the effectiveness and scalability of our models and algorithms on different instances and apply the best approach to solve the real-world case studies.

References

Adkins, R., & Paxson, D. (2017). Replacement decisions with multiple stochastic values and depreciation. *European Journal of Operational Research*, *257*(1), 174–184.

Ahani, P., Arantes, A., & Melo, S. (2016). A portfolio approach for optimal fleet replacement toward sustainable urban freight transportation. *Transportation Research Part D: Transport and Environment*, *48*, 357–368.

Alp, O., Tan, T., & Udenio, M. (2022). Transitioning to sustainable freight transportation by integrating fleet replacement and charging infrastructure decisions. *Omega*, *109*, 102595.

Ansaripoor, A. H., & Oliveira, F. S. (2018). Flexible lease contracts in the fleet replacement problem with alternative fuel vehicles: A real-options approach. *European Journal of Operational Research*, *266*(1), 316–327.

Ansaripoor, A. H., Oliveira, F. S., & Liret, A. (2014). A risk management system for sustainable fleet replacement. *European Journal of Operational Research*, *237*(2), 701–712.

Bellman, R. (1955). Equipment replacement policy. *Journal of the Society for Industrial and Applied Mathematics*, *3*(3), 133–136.

Bellman, R. (1957). *Dynamic programming.* Princeton University Press.

Bradley, S. P., Hax, A. C., & Magnanti, T. L. (1977). Network models. In *Applied mathematical programming.* Addison-Wesley.

Büyüktahtakın, İ. E., & Hartman, J. C. (2016). A mixed-integer programming approach to the parallel replacement problem under technological change. *International Journal of Production Research*, *54*(3), 680–695.

Büyüktahtakın, İ. E., Smith, J. C., Hartman, J. C., & Luo, S. (2014). Parallel asset replacement problem under economies of scale with multiple challengers. *The Engineering Economist*, *59*(4), 237–258.

Chand, S., McClurg, T., & Ward, J. (2000). A model for parallel machine replacement with capacity expansion. *European Journal of Operational Research*, *121*(3), 519–531.

des-Bordes, E., & Büyüktahtakın, İ. E. (2017). Optimizing capital investments under technological change and deterioration: A case study on mri machine replacement. *The Engineering Economist*, *62*(2), 105–131.

Environmental Protection Agency (EPA). (2019). Sources of greenhouse gas emissions [Accessed 1 February 2024]. https://bit.ly/32NGfX5

Figliozzi, M. A., Boudart, J. A., & Feng, W. (2011). Economic and environmental optimization of vehicle fleets: Impact of policy, market, utilization, and technological factors. *Transportation Research Record*, *2252*(1), 1–6.

Fraser, J. M., & Posey, J. W. (1989). A framework for replacement analysis. *European Journal of Operational Research*, *40*(1), 43–57.

Guerrero, S. E. (2014). Modeling fuel saving investments and fleet management in the trucking industry: The impact of shipment performance on ghg emissions. *Transportation Research Part E: Logistics and Transportation Review*, *68*, 178–196.

Hartman, J. C. (2000). The parallel replacement problem with demand and capital budgeting constraints. *Naval Research Logistics (NRL)*, *47*(1), 40–56.

Hartman, J. C. (2004). Multiple asset replacement analysis under variable utilization and stochastic demand. *European Journal of Operational Research*, *159*(1), 145–165.

Hartman, J. C., & Rogers, J. (2006). Dynamic programming approaches for equipment replacement problems with continuous and discontinuous technological change. *IMA Journal of Management Mathematics*, *17*(2), 143–158.

Hartman, J. C., & Tan, C. H. (2014). Equipment replacement analysis: A literature review and directions for future research. *The Engineering Economist*, *59*(2), 136–153.

Hofmann, J., Guan, D., Chalvatzis, K., & Huo, H. (2016). Assessment of electrical vehicles as a successful driver for reducing CO_2 emissions in china. *Applied Energy*, *184*, 995–1003.

Islam, A., & Lownes, N. (2019). When to go electric? A parallel bus fleet replacement study. *Transportation Research Part D: Transport and Environment*, *72*, 299–311.

Jones, P. C., Zydiak, J. L., & Hopp, W. J. (1991). Parallel machine replacement. *Naval Research Logistics (NRL)*, *38*(3), 351–365.

Kagawa, S., Hubacek, K., Nansai, K., Kataoka, M., Managi, S., Suh, S., & Kudoh, Y. (2013). Better cars or older cars?: Assessing co2 emission reduction potential of passenger vehicle replacement programs. *Global Environmental Change*, *23*(6), 1807–1818.

Karabakal, N., Bean, J. C., & Lohmann, J. R. (2000). Solving large replacement problems with budget constraints. *The Engineering Economist*, *45*(4), 290–308.

Karabakal, N., Lohmann, J. R., & Bean, J. C. (1994). Parallel replacement under capital rationing constraints. *Management Science*, *40*(3), 305–319.

Kleindorfer, P. R., Neboian, A., Roset, A., & Spinler, S. (2012). Fleet renewal with electric vehicles at la poste. *Interfaces*, *42*(5), 465–477.

Li, L., Lo, H. K., & Cen, X. (2015). Optimal bus fleet management strategy for emissions reduction. *Transportation Research Part D: Transport and Environment*, *41*, 330–347.

Pugliese, L. D. P., & Guerriero, F. (2013). A survey of resource constrained shortest path problems: Exact solution approaches. *Networks*, *62*(3), 183–200.

Rajabian, A., Ghaleb, M., & Taghipour, S. (2021). Optimal replacement, retrofit, and management of a fleet of assets under regulations of an emissions trading system. *The Engineering Economist*, *66*(3), 225–244.

Rajagopalan, S. (1998). Capacity expansion and equipment replacement: A unified approach. *Operations Research*, *46*(6), 846–857.

Riechi, J., Mácian, V., Tormos, B., & Avila, C. (2017). Optimal fleet replacement: A case study on a spanish urban transport fleet. *Journal of the Operational Research Society*, *68*(8), 886–894.

Seif, J., Shields, B. A., & Yu, A. J. (2019). Parallel machine replacement under horizon uncertainty. *The Engineering Economist*, *64*(1), 1–23.

Sharma, M., Ammons, J. C., & Hartman, J. C. (2007). Asset management with reverse product flows and environmental considerations. *Computers & Operations Research*, *34*(2), 464–486.

Skjærseth, J. B., & Wettestad, J. (2016). *EU emissions trading: Initiation, decision-making and implementation*. Routledge.

Spitzley, D. V., Grande, D. E., Keoleian, G. A., & Kim, H. C. (2005). Life cycle optimization of ownership costs and emissions reduction in us vehicle retirement decisions. *Transportation Research Part D: Transport and Environment*, *10*(2), 161–175.

Stasko, T. H., & Oliver Gao, H. (2012). Developing green fleet management strategies: Repair/retrofit/replacement decisions under environmental regulation. *Transportation Research Part A: Policy and Practice*, *46*(8), 1216–1226.

Vemuganti, R., Oblak, M., & Aggarwal, A. (1989). Network models for fleet management. *Decision Sciences*, *20*(1), 182–197.

Wagner, H. M. (1975). *Principles of operations research*. Prentice-Hall, Inc.

Wang, K.-J., & Nguyen, P. H. (2017). Capacity planning with technology replacement by stochastic dynamic programming. *European Journal of Operational Research*, *260*(2), 739–750.

Xiao, L., Zhang, J., Wang, C., & Han, R. (2023). Optimal fleet replacement management under cap-and-trade system with government subsidy uncertainty. *Multimodal Transportation*, *2*(2), 100077.

Zheng, S., & Chen, S. (2018). Fleet replacement decisions under demand and fuel price uncertainties. *Transportation Research Part D: Transport and Environment*, *60*, 153–173.

Zhou, Y., Liu, X. C., Wei, R., & Golub, A. (2021). Bi-objective optimization for battery electric bus deployment considering cost and environmental equity. *IEEE Transactions on Intelligent Transportation Systems*, *22*(4), 2487–2497.

Zhou, Y., Ong, G. P., & Meng, Q. (2023). The road to electrification: Bus fleet replacement strategies. *Applied Energy*, *337*, 120903.

Chapter 8

Bilevel Optimization with Sustainability Perspective: a Survey on Applications[1]

Abstract

Bilevel optimization, a well-established field for modeling hierarchical decision-making problems, has recently intersected with sustainability studies and practices, resulting in a series of works focusing on bilevel optimization problems involving multiple decision makers with diverse economic, environmental, and social objectives. This survey offers a comprehensive overview of sustainable bilevel optimization applications. First, we introduce the main concepts related to the nature of bilevel optimization problems and define our research boundaries. Then, we review the most relevant works published in sustainable bilevel optimization, giving a classification based on the application domains and their association with well-known operations research problems, while briefly discussing the proposed solution methodologies. We survey applications on transportation and logistics, production planning and manufacturing, waste, material, and environment management, supply chains, and disaster prevention and response. Finally, we outline a list of open questions and opportunities for future research in this domain.

8.1 Introduction

Hierarchical decision-making processes, involving interconnected and selfish players, often arise in the policy-making context (e.g., organizational entities controlling local or private entities) or due to the inherent nature of decision processes (e.g., operational decisions sequencing in production). The prominent game theory concept of *Stackelberg game* (see von Stackelberg, 1934; von Stackelberg, 1952), frequently used in economics to model hierarchical decision-making processes, has been introduced in mathematical optimization in the 70s (see Bracken and McGill, 1973), when the term *bilevel optimization* is born.

[1]Caselli, G., Iori, M., Ljubić, I. (2024). Bilevel optimization with sustainability perspective: a survey on applications (technical report).

Bilevel optimization allows to formalize decision processes where multiple decision makers are organized within a two-level hierarchy. Leader/follower decision problems permeate policy making in societal and industrial contexts, where sustainability concerns can shape the decisions of today for the survival of future generations. However, balancing economic prosperity with environmental concerns and social development is not a trivial task. Operations research studies have recently progressed on problems and solutions for *sustainable process optimization* (Jaehn, 2016; Barbosa-Póvoa et al., 2018), and bilevel optimization has been applied to sustainable practices since its first steps (see, e.g., Candler and Norton, 1977). Although the two topics are indeed connected, a clear picture on their formal interrelations is missing in the literature. Therefore, in this survey, we focus on major trends of bilevel optimization applications to problems related to sustainability, covering *economic* and *environmental* goals, *economic* and *social* goals, or all three "sustainability dimensions" together. We refer the reader to Elkington (2002) for the conceptualization of the triple bottom line accounting framework in business that has contributed to establish the definition of sustainability as a three-dimensional principle. Based on this principle, economic development should be pursued by companies and societies in a manner that benefits both the well-being of people and the health of the planet.

Since the introduction of bilevel optimization in the 70s, an extensive body of literature has emerged. Due to the notorious complexity of solving bilevel problems, the primary focus has been on developing effective solution methods (see recent surveys by Beck et al., 2023; Camacho-Vallejo et al., 2024; Kleinert et al., 2021 on heuristics and exact approaches, respectively). The application of bilevel optimization has extended across various fields such as the military industry (see Bracken and McGill, 1973 for the first bilevel application), transportation, and energy management, as documented in works by Dempe (2020b) and Sinha et al. (2018). While these works showcase bilevel applications, they (1) lack a comprehensive classification and review based on applications studied, and (2) do not focus on three dimensions of sustainability. This study aims to bridge this gap by offering a critical review of applications with sustainability perspective in bilevel optimization.

The survey provides a taxonomy of applications with sustainability perspective (see Figure 8.1.1) and analyzes the hierarchical nature of the bilevel game, identifying the players (number of leaders and followers) at each level and the competition or collaboration between them, especially in the form of an *equilibrium*. It examines the players (e.g., policy makers, companies), their decisions, their goals by detecting whether the leader and follower problems are single- or multi-objective, which of the three dimensions of sustainability (namely, economic, environmental, and social) are considered in each level, and which performance indicators are used to measure the objectives, and solution methods proposed for solving the bilevel problems.

Specifically, the study classifies sustainable bilevel applications according to their sector. We include works where either environmental or social goals are directly included in the players' objective functions (e.g., carbon emissions reduction, maximum equity in resource distribution), or the implementation of the bilevel application comprises positive impacts on the environment and society (e.g., green technology selection, natural disaster response).

We do not cover single-level optimization or other game theory concepts dealing with sustainability, nor bilevel applications without sustainable applications (e.g., bilevel

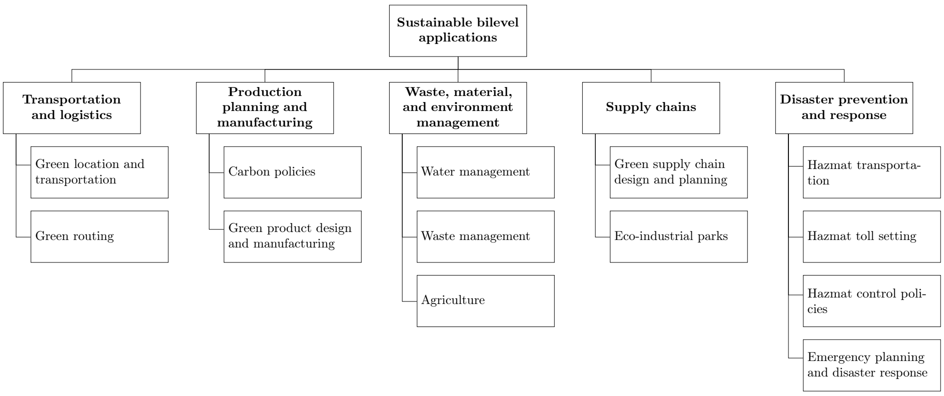


Figure 8.1.1: Sustainable bilevel problems in various sectors.

problems for supply chains optimization). We only focus on the operations research literature, excluding related fields such as operations management. Due to the limited space, we mainly focus on bilevel problems and exclude tri-level and multi-level problems, i.e., optimization problems where players are in hierarchical games of three or more levels.

The paper is structured as follows. Section 8.2 provides a brief overview on bilevel optimization, players, decisions, goals, and solution methods. In Sections 8.4-8.8, we present the following classes of bilevel problems with relevant applications with sustainability perspectives: transportation and logistics (Section 8.4); production planning and manufacturing (Section 8.5); waste, material, and environment management (Section 8.6); supply chains (Section 8.7); disaster prevention and response (Section 8.8). Finally, Section 8.9 identifies the research gaps in the existing literature and provides some guidance for future research. A full list of acronyms is provided in 8.A to improve the readability of the paper.

8.2 Overview of bilevel applications

Bilevel optimization A general optimistic bilevel optimization problem modeling a sequential game with two players (called the leader and the follower) is formulated as

$$\min_{x\in X,y} \quad F(x,y) \tag{8.1}$$

$$\text{s.t.} \quad G(x,y) \geq 0 \tag{8.2}$$

$$y \in S(x), \tag{8.3}$$

where $S(x)$ is the set of optimal solutions of the x-parameterized problem

$$\min_{y\in Y} \quad f(x,y) \tag{8.4}$$

$$\text{s.t.} \quad g(x,y) \geq 0. \tag{8.5}$$

Models (8.1)-(8.3) and (8.4)-(8.5) define, respectively, the so-called upper-level (UL) and lower-level (LL) problems. The UL variables $x \in \mathbb{R}^{n_x}$ represent the decisions of the leader, and the LL variables $y \in \mathbb{R}^{n_y}$ represent the decisions of the follower. The leader, positioned higher in the hierarchy, decides first anticipating the follower's decision outcome. The follower, lower in the hierarchy, reacts to the leader's decision afterwards. In applications studied in this paper, it is commonly assumed that the leader is optimistic meaning that, the leader controls the variables of the follower such that the follower makes the best solution for the leader among the multiple LL optimal solutions. This is the most studied version of bilevel problems since it allows for treatable single-level reformulations. On the contrary, in the pessimistic version the leader assumes that the follower chooses the worst solution for the leader. Solution methods for pessimistic models and their applications are still rare. While classical bilevel problems involve one leader and one follower, various bilevel models, classified subsequently, involve multiple leaders and/or followers in the decision process.

The class of single-leader-single-follower (SLSF) bilevel problems is well established (but still challenging) and represents several realistic situations involving two players (i.e., one leader and one follower). In the so-called multi-leader-follower bilevel problems, multiple players are involved in the UL or LL problems. The classification includes single-leader-multi-follower (SLMF), multi-leader-single-follower (MLSF), and multi-leader-multi-follower (MLMF) problems (see Figure 8.2.1). In the intermediate setting of an SLMF game, there is a single leader but multiple followers, where either the followers are independent or they play a Nash game among them, given the decision of the leader. Let y_j be the vector of decision variables of j-th follower with $j = 1, \dots, n_f$, where n_f is the number of followers, and x the leader's vector of variables. A general SLMF game in the optimistic version is formulated as

$$\min_{x \in X, y_1, \dots, y_{n_f}} \quad F(x, y) \tag{8.6}$$

$$\text{s.t.} \quad G(x, y) \geq 0 \tag{8.7}$$

$$y := (y_j)_{j=1,\dots,n_f} \text{ solves GNEP}(x), \tag{8.8}$$

where y is the vector of all decisions of all followers and GNEP(x) is the set of generalized Nash equilibra of the non-cooperative game among the n_f followers, where the objective function and the feasible set of every follower explicitly depends on the decisions on the other followers. Indicating by y_{-j} the decision variables of all followers except for those of j-th follower, the LL problem of j-th follower can be generally formulated as

$$\min_{y_j} \quad f_j(y_j, x, y_{-j}) \tag{8.9}$$

$$\text{s.t.} \quad y_j \in Y_j(x, y_{-j}), \tag{8.10}$$

where $Y_j(x, y_{-j})$ is the feasible set of j-th follower, defined by private constraints of j-th follower and shared constraints of all followers. GNEPs arise quite naturally from standard Nash equilibrium problems if the players share some common resource (e.g., a transportation link) or limitations (e.g., a common limit on the total pollution in a certain area). The MLSF and MLMF games have more complex structures and are more rare in bilevel applications. We refer the reader to Aussel and Svensson (2020) for general problem formulations and applications of MLSF and MLMF games.

Players and decisions Figure 8.2.1 provides a schematic overview of bilevel games and the players involved. In SLSF applications, the leader typically represents a central authority or a policy maker and the follower can be a local authority (e.g., the national

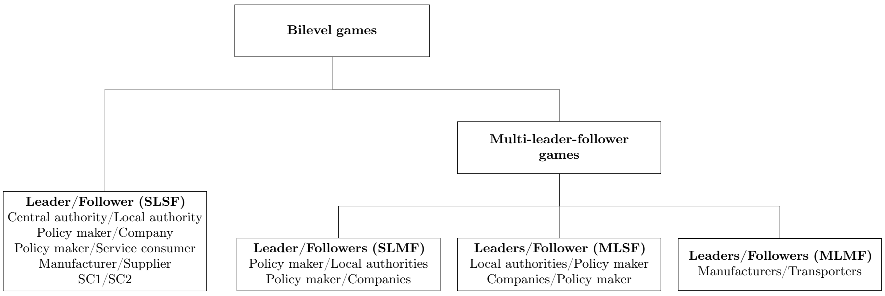


Figure 8.2.1: Taxonomy of bilevel games and players.

authority sets an investment plan and the local authority designs the network layout), a company (e.g., the government sets taxes and subsidies and the company makes production and routing plans), or an average service consumer (e.g., the decision maker optimize the network design and the user makes routing decisions). In other cases, the leader and the follower can be two similar entities hierarchically related and having conflicting goals, such as two members of the same supply chain (SC) or two SCs competing in the same market ("SC1" and "SC2"). In these cases, for instance, the former optimizes customer selection and routing decisions and the latter makes production planning decisions.

Many real-world bilevel problems are modelled as SLMF games between one leader, a policy maker, and multiple followers, companies, local authorities or end users. The policy maker sets prices, subsidies, budgets, taxes, allowances or applies other policy tools to regulate some activities, typically concerning the regulation of market competition, carbon emissions, and social welfare. In consideration of such policies, companies make their production and distribution plans or local authorities design their tactics and operational policies or citizens, generally called end users, adopt their consumption behaviours and travel habits.

The MLSF and MLMF setting are not much explored in the literature on bilevel applications due to the inherent complexity of the problem structure and the methodological limitations. In this survey, we have selected very few applications modelled in such settings, where the roles of the authority and companies or end users are reversed. Other bilevel applications of this kind are seen in the energy and gas sectors (see, e.g., Grimm et al., 2022).

Goals We analyze the three dimensions of sustainability (namely, economic, environmental, and social) within the UL and LL problems of the selected bilevel applications to identify common roles and behaviours of the game players. In Figure 8.2.2, we provide a schematic overview of UL and LL objective functions. The LL problems have typically one economic goal, such as costs minimization, profit maximization, or some utility function maximization for the followers. Sometimes the LL problem is multi-objective with two utility goals such as cost and travel time minimization in network optimization problems. Rarely, when the follower is a local authority relating with a central authority in the UL for the design of SCs or industrial parks, an environmental goal is included in the LL problem, such as pollution cost within the total cost minimization or the solely environmental cost minimization. In specific applications concerning, for instance, the hazardous material transportation, the followers are the carriers minimizing the

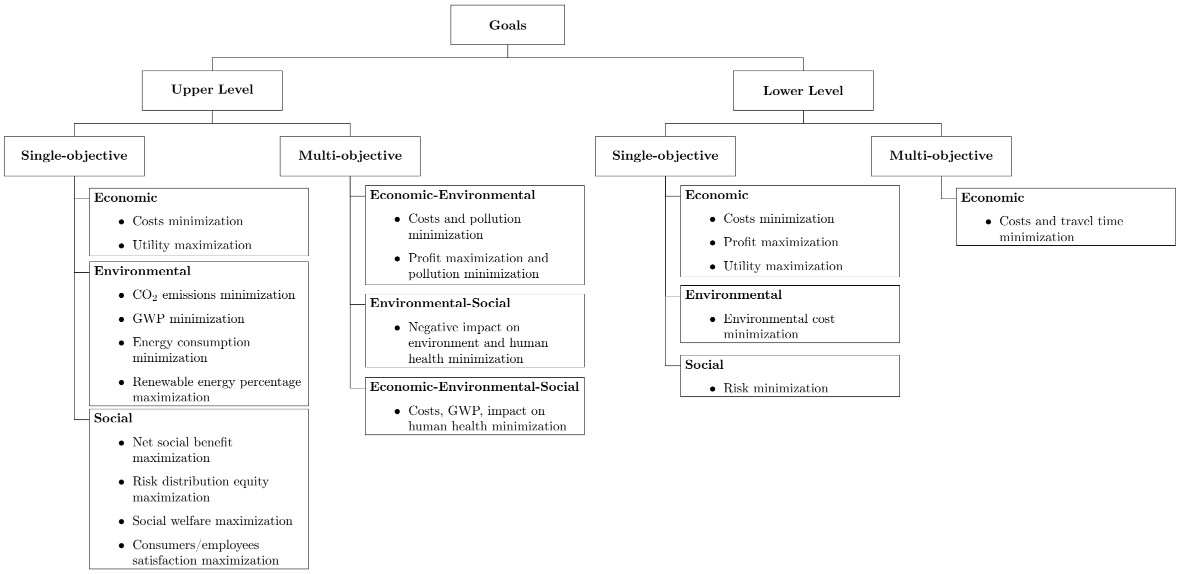


Figure 8.2.2: Taxonomy of economic, environmental, and social objective functions in bilevel applications

.

accident risk on population, that is a social goal.

On the contrary, the UL problems show a variety of single- and multi-objective functions including the three dimensions of sustainability. Very often, the leader is a policy maker whose aim is to improve the economic, environmental and social conditions of the system (e.g., a country or a region, an industrial sector, a public service). The single-level objective functions are economic (total costs minimization, total utility maximization), environmental (minimization of CO_2 emissions, global warming potential (GWP), energy consumption, and maximization of renewable energy percentage), or social (maximization of net social benefit, risk distribution equity, social welfare, consumer or employee satisfaction). Economic and environmental or environmental and social goals are often combined in bi-objective UL problems. For instance, impact on land and human health are minimized in agriculture bilevel applications. In general, pollution minimization is combined with costs minimization or profit maximization in public networks design, SC management, and carbon policy setting. In some cases, the three dimensions are all included in multi-objective UL problems. The applications and results on numerous real-world case studies collected in this survey show the great potential of bilevel optimization in embedding sustainable perspectives into industry and policy decision-making processes.

Solution approaches Bilevel optimization problems are notoriously hard to solve. For instance, Hansen et al. (1992) showed that bilevel problems where UL and LL are linear programming (LP) problems (i.e., bilevel LPs) are strongly $\mathcal{NP}$-hard. Solution methods for bilevel problems strongly depend on the structure and properties of the LL problem (e.g., continuous linear or convex, mixed-integer linear, non convex) and on the coupling between the UL and LL problems (e.g., optimistic or pessimistic version). Assuming some "nice" LL properties (convexity and constraint qualification for all possible UL decisions), an SLSF problem is usually reformulated into a single-level

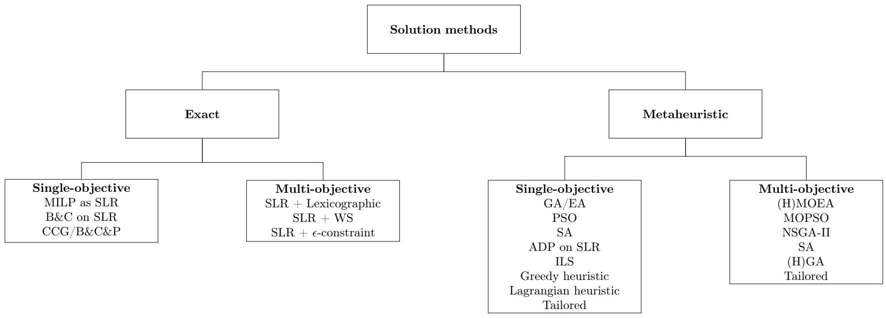


Figure 8.2.3: Taxonomy of solution methods.

problem, that is a single-level reformulation (SLR) by imposing Karush-Kuhn-Tucker (KKT) or strong duality conditions for the follower's solution y. There are various techniques for solving such obtained nonconvex nonlinear models to optimality. In the SLMF setting, the solution for the non-cooperative players in the same level is typically modeled as a Nash equilibrium, that is the solution of a Nash equilibrium problem (i.e., to find a strategic combination of decisions such that no player can gain by deviating unilaterally from it, see Dempe, 2020a) among the followers. Exact solution approaches applied to LP and MILP bilevel problems include Branch-and-Bound (B&B), Branch-and-Cut (B&C), column generation (CG) and cutting plan (CP) algorithms, Benders' decomposition (BD), the penalty function method and approximation algorithms applied to the KKT reformulation. Enumeration algorithms, including the well-known *k-th best* algorithm, are sometimes possible for optimistic bilevel problems. In most applications dealing with real-world bilevel problems, LL problems are non-convex or give intractable SLRs. Therefore, several bilevel problems are solved by metaheuristics, including genetic and evolutionary algorithms as well as (multi-objective) practicle swarm optimization (PSO; MOPSO) and non-dominated sorting genetic algorithm (NSGA-II) especially effective in the case of multi-objecitve UL and LL problems. Hybrid algorithms are also implemented combining heuristics and LP models. We refer to the surveys by Kleinert et al. (2021) and Camacho-Vallejo et al. (2024) on, respectively, mixed-integer programming and metaheuristic techniques for bilevel optimization. In bilevel problems under uncertainty, the uncertainty typically reveals after the leaders take their decision and before the followers react. For bilevel optimization under uncertainty, we refer the reader to Beck et al. (2023). Both the stochastic optimization (SO) and the robust optimization (RO) approaches are used in bilevel applications by defining, respectively, a finite set of scenarios and the deterministic bilevel equivalent problem in the SO setting and the deterministic bilevel counterpart in the RO setting. Figure 8.2.3 gives a sketch of exact and heuristic solution methods commonly adopted in the literature to solve bilevel problems.

8.3 Typical formulations

In this section, we present the mathematical formulation for basic bilevel problems focusing on the optimistic setting. Our goal is to analyze the emerging bilevel problems with real-world applications that relates to sustainability practices and present general mathematical formulations for the basic problems. These formulations can serve as a reference for researchers dealing with specific problems in the class and can be used as a starting point for generalizations. The problems are described as follows. The first is a basic linear price setting problem with one leader and one or multiple independent followers. Let T be the tax (or more generally tariff, or subsidies if negative) decision of the leader (in the UL) and $y \in \mathbb{R}^{n_1}$ and $z \in \mathbb{R}^{n_2}$ be the decision variables sets of the followers (in the LL) on taxed and untaxed activities. Then, let $c_1 \in \mathbb{R}^{n_1}$ and $c_2 \in \mathbb{R}^{n_2}$ be the LL objective function coefficients. The bilevel linear price setting problem with independent followers is formulated as

$$\max_{T \in \mathrm{T}} \quad Ty \tag{8.11}$$

$$\text{s.t.} \quad (y,z) \in \arg\min_{y,z} \quad (c_1 + T)y + c_2 z \tag{8.12a}$$

$$\text{s.t.} \quad g_1(y,z) + g_2(x,y) = b \tag{8.12b}$$

$$y, z \geq 0 \tag{8.12c}$$

$$y \in Y, z \in Z \tag{8.12d}$$

where equation (8.11) is the bilinear objective function of the leader, which maximizes tax revenues from the pricing variables T. The LL problem (8.12a)-(8.12d) typically models cost-minimization problems or routing problem.

In the joint design and pricing problem (see Brotcorne, Labbé, et al., 2008), all variables are defined on arcs of a network, where a set of commodities models the network demand, and arcs are divided in two types: under the control of the leader (i.e., tariff arcs) and otherwise (i.e., tariff-free arcs). The leader decides on the tariff as in pricing model (8.11) (vector T of variables), and on the design of the network. The latter decision is represented by the binary variables vector $x \in X$, where set X contains budget constraints and other specifications. The bilevel design and pricing problem is formulated as

$$\max_{T \in \mathrm{T},\, x \in \{0,1\}^n} \quad Ty \tag{8.13}$$

$$\text{s.t.} \quad (y,z) \in \arg\min_{y,z} \quad f(T,y,z) \tag{8.14a}$$

$$\text{s.t.} \quad g_1(y,z) \leq 0 \tag{8.14b}$$

$$g_2(x,y) \leq 0 \tag{8.14c}$$

$$y \in Y, z \in Z \tag{8.14d}$$

The followers decision variables sets on taxed and untaxed activities are y and z, generally defined on polyhedral sets Y and Z, but could also include integrality constraints. The lower-level problem (8.14a)-(8.14d) is parameterized by the UL tariff variables T, which is solved on the subnetwork resulting from the binary UL variables x. Constraints (8.14a) model the utility or payoff function of the followers. Constraints (8.14b) guarantee the feasibility of the followers' problem. Constraints (8.14c) are flow-balance constraints in the network linking leader's design with followers' decisions (i.e., the followers can use the network only after the leader has undertaken design decisions).

Bilevel pricing problems sometimes contain the so-called network equilibrium problem in the LL. We will illustrate this type of problem as an example of a transportation problem in which the LL problem is modelled as a Wardrop equilibrium. An equilibrium LL problem in transportation network aims at finding the minimum travel time (or cost) of all the followers. Typical applications arise for traffic management in congested urban areas. In equilibrium network pricing problems, followers are not independent players but interact with each other such that they are in a simultaneous equilibrium (i.e. no player can unilaterally improve their objective by changing their decision, as stated in Wardrop, 1952). This is because each driver decides to use the least duration path and therefore, all routes have the same travelling time. The user equilibrium assumes that the drivers (i.e., followers here) have a complete information about the available paths and the network flows are stable over time. We can define a general equilibrium model in the LL problem containing pricing and design UL variables. Let the graph $G = (V, A)$ represent the network, where A is the set of arcs. Variables y_a represent the flow on each arc $a \in A$ of the network, while z_c^k represent the flow of the origin-destination pair c on path k, with P_k^c being the set of paths containing pair c. As in previous models, vectors T and x indicate UL variables for pricing and design decisions, respectively. The LL results in a user equilibrium model generally formulated as

$$\min_{y,z} \sum_{a \in A} \int_0^{y_a} f_a(\omega, T) d\omega \tag{8.15a}$$

$$\text{s.t.} \quad y_a = \sum_{c \in C} \sum_{\substack{k \in K_c: \\ a \in P_k}} z_c^k \quad \forall a \in A \tag{8.15b}$$

$$d_c = \sum_{k \in K_c} z_c^k \qquad \forall c \in C \tag{8.15c}$$

$$y \leq Mx \tag{8.15d}$$

$$y_a \geq 0 \quad \forall a \in A \tag{8.15e}$$

$$z_c^k \geq 0 \quad \forall c \in C, \forall k \in K. \tag{8.15f}$$

Objective function (8.15a) integrates the time function for each follower over each arc flow (i.e., the longer the congestion, the longer the travel time). Constraints (8.15b) bounds the arc flows y_a to the flow of the paths passing through the arc $a \in A$. Constraints (8.15c) guarantee demand d_c satisfaction for each commodity $c \in C$. Constraints (8.15d) link UL and LL variables (i.e., the followers use an arc only if the leader designs it). Constraints (8.15e)-(8.15f) state the domain of the LL variables. A more general approach for modelling equilibrium constraints is found in Brotcorne, Marcotte, Savard (2008).

These three formulations cover the majority of applications collected in this work. Governmental policies for reducing the environmental impact of, for instance, industrial production and supply chains, and improving social welfare are modeled as pricing problems. Network pricing decisions are crucial for airline, freight, and urban transportation, as well as telecommunication and service industries, especially in modern markets where intense price competition from economic deregulation on one hand, and environmental and social concerns from sustainable development regulations and practices on the other hand, coexist with network modifications.

8.4 Transportation and logistics

Network design is a major class of combinatorial optimization problems represented conceptually as the selection of a subset of links in a graph, with numerous application developments in transportation and logistics (see Cordeau et al., 2021). The selected works address bilevel network design applications in sustainable transportation and logistics. We include studies related to "green" extensions of commonly encountered problems in this domain, like the Facility Location Problem (FLP) and the Vehicle Routing Prolem (VRP).

8.4.1 Green location and transportation

In transportation bilevel problems, it is common that the government or network operator acts as the leader making decisions related to network design, in particular price setting, while the users of the network act as the followers, making individual route choices. See also Section 8.8 for other transportation bilevel problems applied to disaster prevention and response. Given the growing importance of sustainability in transportation, recent bilevel problems in this field deal with sustainability, either directly or indirectly. Sustainable transportation, in our context, refers to alternative transport modes that aim to reduce carbon emissions, such as bicycles. Table 8.4.1 provides a classification of the referenced works based on their application, the type of bilevel game and whether the LL problem is modelled as an equilibrium problem ("Eq."), players and objectives in the UL and LL problems, and the proposed solution method.

Highway network design and pricing

Ben-Ayed et al. (1992) and Sinha et al. (2015) introduce transportation bilevel problems focused on highway network design. These problems fall under the SLSF class, where the government acts as the leader and the average user serves as the follower. Ben-Ayed et al. (1992) address rural highway systems in developing countries. The government, responsible for network flow and capacity decisions, aims to minimize overall system costs, including environmental and social damage costs. The user determines the routes to maximize their individual utility functions. The authors develop an iterative heuristic algorithm and solve a realistic instance from a case study in Tunisia. Sinha et al. (2015) study the extended problem with multi-objective UL and LL problems. The government in the UL aims at maximizing revenues and minimizing pollution by defining highway tolls, whereas the user in the LL seeks to minimize total costs and travel times by optimizing routes. A Pareto-optimal front of decisions balancing revenues and pollution is given for the UL problem, and a LL Pareto-optimal front balancing travel cost and time is given for any UL optimal decision. The authors propose a multi-objective evolutionary algorithm (MOEA) and solve an illustrative example. They also extend the problem to the SLMF version, where the decisions of multiple independent followers are modelled in multiple LL problems.

Angulo et al. (2014) propose a bilevel SLMF model for the location of new corridors to a highway network. The leader, the network designer, takes location decisions with the aim of minimizing total costs (including environmental ones), while the followers

Table 8.4.1: Summary of papers on bilevel applications focused on green bilevel location and transportation.

Application	References	Bilevel game		Players		Objectives		Method
		Class	Eq.	UL	LL	UL	LL	
Highways network design and pricing	Ben-Ayed et al. (1992)	SLSF		Government	Average road user	Min total cost	Max utility	HEUR
	Sinha et al. (2015)	SLSF/ SLMF		Government	Average road user/Users	**MO**: Max revenue; Min pollution	**MO**: Min total cost; Min total travel time	MOEA
	Angulo et al. (2014)	SLMF		Government	Road users	Min total cost	Min travel time	PSO+DP
	Wang et al. (2014)	SLMF	✓	Tolls DM	Road users	**MO**: Min travel time; Min emissions; Min impact on health	**MO**: Min travel cost; Min travel time	NSGA-II
	Wen and Eglese (2016)	SLMF	✓	Tolls DM	Road users	**MO**: Min emissions; Min travel time	Min total cost	SA
	Zhao et al. (2016)	SLMF	✓	Tolls DM	Road users	Min emissions	Min total travel time	EA+AON
Bike lanes network design	Sohn (2011)	SLMF	✓	Network designer	Motorists and cyclists(or government)	**MO**: Min motorists time and min cyclists time(or max motorist speed)	Min travel time	NSGA-II
	Bagloee et al. (2016)	SLMF	✓	Network designer	Motorists and cyclists	Min total travel time	Min travel time	B&B
	S. Liu et al. (2022)	SLMF		Network designer	Cyclists	Max total utility	Max utility	SLR
	Gaspar et al. (2015)	SLMF	✓	Network designer	Motorists, cyclists, and bus users	Max number of cyclists	Min travel time	HEUR
	Rashidi et al. (2016)	SLMF	✓	City planner	Travellers	Min total travel cost (incl. safety)	Min travel cost (incl. safey)	HEUR
Green facility location	Chen et al. (2020)	SLMF	✓	EV service provider	EV drivers	**MO**: Min construction cost and min total drivers time	Min time	SLR+IA
	Hu et al. (2022)	SLSF		Government	GSI consumer	**MO**: Min distance; Min cost; Min pollution	**MO**: Max efficiency; Max density	MOPSO
	Gang et al. (2015)	SLMF		Government	Enterprises	**MO**: Min pollution; Min total cost	Min total cost	PSO
	Zhou et al. (2023)	SLMF	✓	Network designer	Users	**MO**: Max transfer flow; Max transit priority	Min travel time	HEUR
Air passenger transportation	Qiu, Xu, Xie, et al. (2020)	SLMF	✓	Goverment	Airlines	Max net social benefit	Max profit	GA

are the users deciding on the road and the transport system (highway or road). As a solution approach, the authors propose an PSO algorithm for the leader's problem. A modified Dijkstra algorithm is presented to solve the LL problem of shortest paths for origin-destination pairs under the assumption of uncongested highways (i.e., costs not dependent on traffic flow).

Wang et al. (2014) propose an SLMF toll pricing problem on a road network with multiple objectives in both UL and LL. The leader, responsible for setting tolls, minimizes the total travel time of the system, the total vehicle emissions, and the negative impact on human health. The LL problem is formulated as a bi-objective user equilibrium among the travellers, who aim at minimizing travel costs and travel times. The authors propose a metaheuristic algorithm, where the multi-objective UL problem is solved by the modified NSGA-II, and the LL problem is solved by a quasi-Newton method, and solve a small illustrative example.

Wen and Eglese (2016) present an SLMF pricing model to minimize CO_2 emissions in a small road network, where the toll pricing strategy is determined in the UL and the dynamic equilibrium between the road users is defined in the LL. An illustrative example solved by simulating different toll pricing strategies and users' behaviour reveals the impact of various pricing strategies on costs and CO_2 emissions.

In the SLMF bilevel problem proposed by Zhao et al. (2016), the leader aims at setting an optimal greenhouse gas (GHG) emissions charge scheme for travellers, who decide on their travel mode (e.g., cars, motorcycles, bus) to minimize their total travel costs in a user equilibrium model. The goal of the leader is a GHG emissions target. The solution method integrates an evolution algorithm for the UL and the "all-or-nothing" algorithm for the LL traffic assignment. The authors show managerial insights from a numerical example of application.

Bike lanes design

Bike lanes are an alternative transport mode that indirectly reduce carbon emissions from vehicles. In bilevel bike lanes networks optimization problems there are multiple followers, the network users, interacting with the leader, the network designer, and with each other in a network equilibrium problem. Sohn (2011) deal with road diet network design problems (i.e., how to dedicate road portions exclusively to cyclists without affecting motorists). They present two multi-objective bilevel models, each with a different UL problem and goals: (i) in the fixed mode-share model, motorists and cyclists cannot change their travel mode and the UL goals are to minimize the total travel time of motorists and cyclists; (ii) in the variable mode-share model, motorists and cyclists are allowed to change their travel mode and the UL goals are to minimize motorists travel time and to maximize the average speed of the remaining motorists after the implementation of the road diet (i.e., government's goal). The LL is modeled as a user equilibrium assignment problem between motorists and cyclists. NSGA-II algorithm is applied to a small test network to find an approximated Pareto front of non-dominated solutions for each model.

Bagloee et al. (2016) tackle bike lanes network design in congested areas, considering the total users' travel time minimization in the UL (actually defined as the sum of all the involved disutilities such as, e.g., travel time, waiting time, petrol, pollution), and including both cyclists and motorists in a multiclass user equilibrium traffic flow model at the LL. The authors present an effective B&B algorithm that is able to solve a real-world instance of the problem with around 1000 nodes and 3000 arcs.

S. Liu et al. (2022) present a bilevel bike lanes network design problem in which the leader designs the bike network to maximize the total utility, while the followers (the cyclists) make individual route choices to maximize their own utility. The authors provide an MILP SLR, which is able to solve a real-world large-scale case study in China. They also show how the introduction of cyclists' route choices (i.e., the bilevel model) impacts the solution with respect to the single-level model that only maximizes cyclists' utility function.

Gaspar et al. (2015) propose another SLMF bilevel model for optimizing cycling networks where the followers include car users, bus users, and cyclists. The leader decides on the type of cycle lanes to design for each street of the network to maximize the number of cyclists. The LL problem is a modal-split assignment problem, in which users decide their optimal route and travel mode. The proposed GA is applied to a real-world medium-size case study of the city of Santander (Spain) with around 1500 candidate links for bicycle lanes.

Bike lanes are also connected to sidewalks and crosswalks, which are similarly designed with bilevel optimization. Rashidi et al. (2016) explore pedestrian safety in traffic management through bilevel optimization. They formulate an SLMF bilevel problem in which the city planner acts as the leader, making decisions on the location of sidewalks and crosswalks. The travellers, as followers, decide on their transportation mode (i.e., automobile, public transport, or walking) and routes. The LL problem is modeled as an equilibrium traffic assignment problem. Two heuristic methods are presented to solve small illustrative examples, showing that pedestrian safety can be improved and costs can be reduced.

Green facility location

In the FLP, the objective is to strategically position a set of facilities to minimize the cost of meeting the demands of a specific group of customers (see Laporte et al., 2019). In this bilevel problem, the leader typically decides on the location and size of the facilities that could be used by the followers. Chen et al. (2020) formulate the bilevel problem of optimal location and capacity planning of electric vehicle (EV) charging stations. In the UL, the service provider of EV sharing systems assign facilities and determine their capacities with the objective of minimizing the total cost of construction and EV drivers' travel and charging waiting time. In the LL, drivers optimize their routes and charging facility choices to achieve equilibrium behaviour (i.e., minimum total travel time and waiting time for all the drivers). The authors solve the proposed nonlinear SLR with an iterative algorithm on different illustrative examples (up to around 30 nodes, 80 links, and 40 candidate locations).

Hu et al. (2022) model a facility location and size problem for general education infrastructures, such as schools, as a multi-objective SLSF bilevel problem. The facility locations are decided by the government in the UL problem with three social, economic, and environmental goals (i.e., minimization of distance, costs, and environmental pollution), while the optimal facility size is decided by the users in the LL problem with two goals (i.e., maximization of service efficiency and service density). The authors adopt an MOPSO algorithm and solve a medium-size real-world case study in China, providing interesting managerial insights.

Several sustainable bilevel applications deal with industrial parks optimization problems. We dedicate below a specific section to these problems (see Section 8.7.2), and focus here only on industrial parks FLPs. Gang et al. (2015) include green goals in an FLP for stone industrial parks, districts of stone enterprises located close to each other to share infrastructures and control costs and pollution from, for instance, dust and water consumption. In this problem, the leader in the UL is the local government, who aims at minimizing emissions pollution and total development and operating costs, whereas the followers in the LL are the stone enterprises in conflict with each other but in cooperation with the leader to minimize their total cost subject to a constraint on maximum emission. The authors develop a PSO algorithm based on the concept of satisfactory level of objectives for both the UL and LL and solve a case study in China, showing a 20% reduction for all the objectives of both the leader and the followers.

Transit-oriented development is a urban planning approach focused on creating compact and mixed-use urban areas closely linked with mass transit stations to efficiently integrate jobs, housing, and services, aiming for more sustainable and accessible cities. A bilevel FLP in this field is presented by Zhou et al. (2023), who consider the optimal placement of TOD stations in a Chinese city. They present a bilevel model with a multi-objective UL problem and a multi-modal network equilibrium in the LL. They develop a metaheuristic algorithm and solve a large-size real-world instance. They show how TODs can reduce carbon emissions in urban areas and how they impact multimodal networks.

Air passenger transportation

Qiu, Xu, Xie, et al. (2020) present an SLMF bilevel model for determining the optimal carbon tax incentive policy for air passenger transportation fossil fuel saving and carbon emissions reduction. The leader is the government, setting the carbon tax and allocating carbon tax incentive subsidies to the airlines for maximizing the net social benefit of the policy and to increase the system efficiency. Airlines are the followers, developing their air transport planning to maximize their profit. A genetic algorithm (GA) is proposed to solve a case study in China. Results show that a carbon tax incentive policy has the potential to save fuel and improve environment under proper conditions.

8.4.2 Green routing

In the VRP, routes must be optimized to visit a set of customers at minimum total travel cost (see Cordeau et al., 2007). Despite several versions of the classical VRP with the inclusion of green objectives or constraints have been explored in the literature, not many, to the best of our knowledge, have been approached with bilevel optimization (see Table 8.5.1).

The Pollution Routing Problem (PRP) considers time windows and vehicle speeds to reduce carbon emissions in the road freight transport sector (Bektaş and Laporte, 2011). Nath et al. (2018) study a bilevel multi-objective PRP where the depot acts as the leader and vehicles as followers. In the UL, the customers are assigned to vehicles to minimize both fuel consumption and total travel distance. In the LL, vehicle routes are determined to minimize individual travel distance. The authors employ a metaheuristic algorithm based on NSGA-II for the UL and a GA for each vehicle in the LL, yielding improved results compared to NSGA-II applied to a single-level multi-objective PRP.

Qiu, Xu, Ke, et al. (2020) present another bilevel PRP, incorporating a carbon pricing system. The authority acts as the leader and sets a carbon tax and a carbon subsidy to a freight transport company based on actual carbon emissions and emissions reductions, respectively, with the aim of minimizing the company's carbon emissions. The company acts as the follower and aims at reducing overall costs, including fuel, emissions, and operating costs, through routing scheduling decisions. The authors propose heuristic method based on PSO and Adaptive Large Neighborhood Search (ALNS) algorithms for the UL and LL, respectively. Results based on real-world UK instances of the problem reveal that carbon pricing initiatives can significantly reduce emissions with a relatively modest increase in costs.

Other green VRPs deal with public transport systems. Parvasi et al. (2017) and Calvete, Galé, Iranzo, Toth (2023) study bilevel problems related to the school bus routing optimization with students' choice consideration. In Parvasi et al. (2017), the leader in the UL is the public transportation company deciding bus stations location and bus routes to maximize profit. Profit depends on the estimated number of students attracted in the system. The students are the independent followers in the LL problems, who decide whether to take which bus at which station to minimize their own cost. The authors propose two hybrid meta-heuristic algorithms based on GA, simulated annealing (SA), and tabu search (TS) heuristics effectively applied on random instances.

In Calvete, Galé, Iranzo, Toth (2023), the leader of the SLMF bilevel problem is the authority who decides the routes, given the bus stops, and the number of students assigned to each stop considering students/stops accessibility and preferences. The goal of the leader is to minimize the total cost. The students, in the LL problem, decide which bus stop to select. The LL objective is to minimize the total value of preferences (the smaller the value the higher the preference). The authors provide an SLR based on duality theory and a metaheuristic algorithm solved on benchmark instances showing good performances.

8.5 Production planning and manufacturing

Governments' environmental policies and public awareness across industrial entities are driving the introduction of green technologies and sustainable industrial practices (see Tables 8.5.2 and 8.5.3). In bilevel frameworks, either the government imposes a carbon policy and the industry responds or multiple industrial entities interact with each other to reduce the environmental impact of industrial outputs, typically in the UL, while seeking efficiency in both UL and LL.

8.5.1 Carbon policies

Governmental policies and industrial reactions can be generally modelled as bilevel non-linear price setting problems. Carbon policies to control industrial activities pollution are mainly direct and indirect taxes, subsidies, and tradable permits. Sinha et al. (2013) study an SLSF bilevel problem to optimize government's environmental regulatory decisions on a mining project. The government, the leader, decides on the optimal

Table 8.5.1: Summary of papers on bilevel applications focused on green routing.

Application	References	Bilevel game		Players		Objectives		Method
		Class	Eq.	UL	LL	UL	LL	
Pollution Routing Problem	Nath et al. (2018)	SLMF		Depot	Vehicles	**MO:** Min fuel consumption; Min total travel distance	Min travel distance	NSGA-II+GA
	Qiu, Xu, Ke, et al. (2020)	SLSF		Authority	Company	Min CO_2 emissions	Min total costs	HEUR
School bus routing	Parvasi et al. (2017)	SLMF		Public transportation company	Students	Max profit	Min cost	HA
	Calvete, Galé, Iranzo, Toth (2023)	SLMF		School	Students	Min total cost	Max preference	HEUR

Table 8.5.2: Summary of papers on bilevel applications focused on carbon policies on production planning.

Application	References	Bilevel game		Players		Objectives		Method
		Class	Eq.	UL	LL	UL	LL	
Carbon policies	Sinha et al. (2013)	SLSF		Government	Mining company	**MO:** Max revenues; Min pollution	Max profit	HMOEA
	Sanjeet Singh and Bhattacharya (2018)	SLSF		Manufacturer/Seller	Seller/Manufacturer	Max profit	Max profit	SLR
	Hong et al. (2017)	SLMF	✓	Local government	Firms	Max social welfare	Max profit	PDP + BSA + GA
	Almutairi and Elhedhli (2014)	SLSF		Government	Industry	Min production and target deviation weighted difference	Max surplus	SLR

Table 8.5.3: Summary of papers on bilevel applications focused on green product design and manufacturing.

Application	References	Bilevel game		Players		Objectives		Method
		Class	Eq.	UL	LL	UL	LL	
Green product design	Yanxia Wu and Cheng (2020)	SLMF		Platform operator	Service demanders	**MO:** Max environmental, economic, social perf.	**MO:** Min time; Min price; Max quality	(H)MOPSO
	Ma et al. (2018)	SLSF		Manufacturer	Supplier	Max utility/cost ratio	Min total costs	GA
	Zeng et al. (2020)	SLMF		Station owner	Station users	Max net revenue	Max utility	SLR + CCG
	Zhu et al. (2020)	SLSF		Designer	Designer	**MO:** Min fuel consump.; Min GHG em.; Min net present cost	Min fuel consump.	MOPSO + RCA

tax rate for each time period to maximize total revenues from the project (including taxes) and minimize pollution produced by the mining company. The company, the follower, decides on the extraction amount for each time period to maximize profit. The authors propose a hybrid multi-objective evolutionary algorithm (HMOEA) to solve a case study in Finland, providing a valuable approximate revenues/pollution Pareto front of solutions.

Sanjeet Singh and Bhattacharya (2018) present two SLSF bilevel models for off-shore manufacturing contracts optimization with pollution control. In the first model, the leader is a seller firm in a developed country deciding the retail price of a product in its market, and the follower is another firm in a developing country that manufactures the product with a lower manufacturing cost and decides on the transfer price to the seller. Both firms maximize their after-tax net profits, where a green tax and an import duty are paid, respectively, by the manufacturer and the seller to their governments. In the second model, on the contrary, the leader is the manufacturer and the follower is the seller. The authors obtain MILP and MINLP reformulations based on KKT conditions and solve them to optimality. A case study on electronics goods between a seller in US and a manufacturer in China shows that the bilevel models allow to improve the contract for both parties under pollution control policies, even when the seller is the follower.

Hong et al. (2017) introduce an SLMF bilevel model for a carbon emissions cap-and-trade scheme. The local government acts as the leader, setting the emission trading scheme with the aim of maximizing social welfare of firms including economic and environmental benefits. Firms act as the followers, making production planning and technology selection decisions and trading emission allowances among each other to maximize their profit while respecting the government's emission reduction policy. The followers' problem is modelled as a Cornout equilibirum model. The authors propose an HA that combines a polynomial dynamic programming (PDP) algorithm for followers' decision, a binary search algorithm (BSA) for the equilibrium, and a GA for the leader's decision. A numerical example in China shows the effectiveness of a cap-and-trade scheme to induce carbon emissions reduction.

In the SLSF bilevel model proposed by Almutairi and Elhedhli (2014), the government in the UL determines the tax rate on an industry to minimize the weighted difference between production quantity and deviation from the target emission factor. The follower is the industry, determining production quantities and selecting fuel types to maximize the sum of consumers' and producers' surpluses, considering the carbon tax payment on the deviation from the target emission factor. The authors present a KKT reformulation approach and apply it to solve a cement industry case study in Canada. Results show that moderate tax rates can bring large reductions in the overall emissions and the optimal tax rate reduces the quantity supplied by high emitters and increases the quantity supplied by low and average emitters.

8.5.2 Green product design and manufacturing

Green practices are spreading across different industrial sectors. Yanxia Wu and Cheng (2020) assess the economic, environmental, and social sustainability of cloud manufacturing services by proposing a multi-objective SLMF bilvel model. The aim is to satisfy customers' service requirements with the minimal resource consumption and the lowest possible environmental impact. In the UL, the platform operator decides which demanded services to fulfill and how, in order to maximize the three dimensions of sustainability: environmental performance (e.g., energy consumption, emissions, and percentage of energy from renewables), economic performance (e.g., flexibility and profitability), and social performance (e.g., employee learning, employee satisfaction, and consumer satisfaction). In the LL, the service demanders determine the operational strategy for service composition to optimize the three dimensions of service quality, that is to minimize time and price, and to maximize quality. The authors propose a hybrid PSO algorithm and show that it is more efficient that the classical PSO on the single-level multi-objective model.

Ma et al. (2018) study an SLSF bilevel problem for green product design in a supply chain where incentives are given to suppliers to reduce carbon emissions. The manufacturer acts as the leader, who drives the product sustainability by paying the carbon tax and giving carbon incentives to the supplier. The manufacturer decides on production modules and aims to maximize the ratio between customer utility function and total costs. The supplier acts as the follower, deciding on product modules production to minimize net production costs. The authors propose a GA and show an illustrative example on notebook computers, showing the effectiveness of their carbon incentive-based bilevel model in improving both the product profit and the carbon emission.

EVs represent a valid alternative to fuel vehicles due to their operating efficiency and absence of direct emissions, provided that renewable resources are effectively available. Zeng et al. (2020) introduce a robust SLMF bilevel model for the design and planning of a public Plug-in EVs charging station, where the leader is the station owner and the followers are the station users. The leader decides on the renewable energy and energy storage capacity and retail price to maximize the net revenue from the station. The users maximize their utility by deciding on their charging-energy demand, after uncertainty on renewable availability, wholesale energy price, and number of Plug-in EVs arrivals realizes. The authors propose a KKT reformulation for the model and a column-and-constraint generation (CCG) algorithm to solve a case study. They compare the results of the bilevel approach with a classical single-level one and show the benefits of the former to increase renewable availability and decrease energy costs.

Zhu et al. (2020) integrate optimal component sizing and energy management for hybrid EVs manufacturing in a multi-objective bilevel model. The leader is represented by the hybrid EV designer making decisions on vehicle component sizing to minimize fuel consumption, GHG emissions, and the net present cost. The follower is represented by another EV designer making decisions on the energy management system with the goal of minimizing the fuel consumption. The solution method combines MOPSO at the UL and a heuristic rule-based control approach (RCA) at the LL. The results on an illustrative example show that the bilevel approach is able to find optimal solutions resulting in less fuel consumption, less GHG emission, and less net present cost compared to the conventional single-level optimization.

8.6 Waste, material, and environment management

We explore a selection of works leveraging bilevel optimization in waste, material, and environment management. We survey water and waste management, agriculture applications, and various problems for disaster prevention (hazardous material management) and disaster response (emergency planning and relief logistics).

8.6.1 Water management

Water is a primary source of life on our planet and its management has consistently posed complex environmental and societal challenges. Multiple stakeholders are involved in the water management process and different issues may arise. We cover water scarcity, water pollution, and water system network optimization (see Table 8.6.1).

Water scarcity

Anandalingam and Apprey (1991) introduce an insightful problem of conflict resolution in international rivers. They present a comparative analysis of different hierarchies in multi-level Stackelberg games involving an arbitrator (specifically, the United Nations) and two countries (India and Bangladesh) who compete for the use of Ganges river waters for hydroelectric power, irrigation, and flood protection. Each player aims to maximize their net benefit. In the bilevel hierarchy, the arbitrator is the leader and the two countries act as followers of equal status, in conflict with both each other and the leader. In the three-level hierarchy, the arbitrator again assumes the leader role, but the two followers are engaged in a leader-follower hierarchical relationship. Both models are solved by SLRs based on the penalty function method. The authors show interesting insights on the illustrative case study of Ganges river. First, higher objective function values are obtained under the three-level model, owing to substantial subsidies from the arbitrator covering a significant portion of the system costs. Second, in the three-level model, it is mostly indifferent who is the leader and who is the follower between the two counties in the second-level hierarchy.

More recently, Calvete, Galé, Iranzo, Mateo (2023) present an SLMF bilevel formulation to address the optimal allocation of water to demand points that are in conflict with each other due to water shortage. The leader is represented by the government or the local authority deciding on water allocation to demand points and establishing associated fees. In the LL, multiple followers are represented by demand points managers. The followers make decisions regarding water distribution within their specified areas. The leader pursues environmental and social objectives, including minimizing the total water deficit, maximizing user satisfaction, and minimizing water prices for users. Meanwhile, each follower at the LL seeks to maximize their individual net economic return. The bilevel model is reformulated as a single lexicographic multi-objective MILP model and tested effectively on several instances representing different water system scenarios.

Water pollution

Addressing water pollution is another critical concern, particularly in the context of UN Sustainable Development Goal (SDG) 6, which emphasizes the importance of

Table 8.6.1: Summary of papers on bilevel applications focused on water management.

Application	References	Bilevel game		Players		Objectives		Method
		Class	Eq.	UL	LL	UL	LL	
Water scarcity	Anandalingam and Apprey (1991)	SLMF		United Nations	Countries	Max economic net benefit	Max economic net benefit	SLR
	Calvete, Galé, Iranzo, Mateo (2023)	SLMF		Government	Water demand points managers	**MO:** Min water deficit, Max satisfaction, Min water price	Max economic net benefit	SLR+Lexic.
Water pollution	Zhao et al. (2013)	SLMF		Lake authority	Local authorities	Min pollution reduction cost	Min pollution and transfer cost	SLR
	He et al. (2023)	SLMF		Lake authority	Local authorities	Max distribution equity	Min environmental cost	SLR

Table 8.6.2: Summary of papers on bilevel applications focused on waste management.

Application	References	Bilevel game		Players		Objectives		Method
		Class	Eq.	UL	LL	UL	LL	
Urban waste	Caramia and Pizzari (2022)	SLSF		Central authority	Local authority	**MO:** Min land-use stress, Min impact on health	Min transportation cost	SLR+WS
Hazardous waste	Amouzegar and Moshirvaziri (1999)	SLMF		Central authority	Firms	Min total costs	Min cost	HEUR

Table 8.6.3: Summary of papers on bilevel applications focused on agriculture.

Application	References	Bilevel game		Players		Objectives		Method
		Class	Eq.	UL	LL	UL	LL	
Agricultural policy instrument	Candler and Norton (1977)	SLMF		Policy maker	Producers	**MO:** Max employment, income, production	Max profit	Exact
	Önal et al. (1995)	SLMF	✓	Policy maker	Farmers	Max total revenue	Max surplus	SLR, HEUR
	Whittaker et al. (2017)	SLMF		Policy maker	Farmers	**MO:** Max total profit, Min nitrogen in land	Max profit	HGA
	Barnhart et al. (2017)	SLMF		Policy maker	Farmers	**MO:** Max total profit, Min fertilizer pollution	Max profit	MOEA
Agricultural supply chain	Albornoz and Vera (2023)	SLSF		Producer	Wholesaler	Max profit	Min cost	SLR

clean water. Specifically, the lake ecological environments deterioration has become a significant global problem. In the works of Zhao et al. (2013) and He et al. (2023), SLMF bilevel programming formulations are employed to model the hierarchical structure of authorities responsible for lake water pollution control. In the UL, there is one leader, the lake authority. In the LL, there are multiple followers, the sub-regional authorities.

In Zhao et al. (2013), the leader strategically sets tax rates to minimize the total pollution reduction cost while ensuring compliance with environmental quality standards. If the amount of pollutant generated by each region exceeds the national standard, the region should pay a fee proportional to the pollutant quantity transferred and the transfer tax rate. Pollution reduction costs are set by the lake authority. The followers, in turn, make decisions regarding their pollution reduction tactics, encompassing both pollution reduction and transfer costs minimization.

In He et al. (2023), the leader allocates load permits to the followers to achieve maximum equity in water distribution. The followers implement their compensation tactic (i.e., pay for extra pollution or be rewarded for emissions reduction) aiming to minimize their total environmental costs. Both models are solved by means of the KKT reformulation approach and applied to solve the case study of Taihu Basin (China). The results show the effectiveness of this hierarchical system in reducing water pollution.

8.6.2 Waste management

Authorities and citizens are compelled to assess the economic efficiency and environmental impact of waste due to its global steady volume increase and the high pollution rate of its treatment process. Within the waste management process, the national or regional authority is responsible for strategic decisions such as facility location for waste treatment, and local authorities or private companies deal with operational decisions such are routing for waste collection, leading to bilevel programming models (see Table 8.6.2).

Urban waste

Municipal Solid Waste Management (MSWM) encompasses a set of practices aimed at collecting and recycling solid wastes generated by citizens, involving a complex network of facilities and related stakeholders, which must be efficiently designed.

In Caramia and Pizzari (2022), the global and local authorities serve as the leader and the follower responsible for waste network design and transportation decisions, respectively. The problem involves location, capacity, and transportation decisions on the levels of collection centers, sorting facilities, landfills and incinerators. The goal of the leader is to minimize the average land-use stress (i.e., environmental goal) and the impact on public health (i.e., social goal). The follower aims at minimizing the total transportation costs. The authors present two bilevel formulations under both the optimistic and pessimistic assumption, along with corresponding SLRs with the weighted sum (WS) method for dealing with the two UL objectives. They apply these formulations to a real-world case study concerning the municipal waste of an urban area in Thailand, showing that the results of bilevel programming are more realistic than the ones obtained with a bi-objective approach.

Hazardous waste

Hazardous waste includes all kinds of dangerous and “special” sources of waste, typically generated by industrial entities, such as chemicals, batteries, and electronics.

Motivated by a case study in California (US), Amouzegar and Moshirvaziri (1999) introduce a capacity allocation and facility location SLMF bilevel model for hazardous waste management. In this model, the central authority assumes the leader role, while the firms engaged in the waste management process act as followers. The leader controls allocation and location decisions by setting prices and taxes for harmful policies, with the aim of minimizing total regional costs. The followers devise policies to minimize their costs (including taxes). The authors propose a two-phase heuristic algorithm that initially solves a relaxed single-level model, excluding the follower’s objective, and subsequently solves the followers’ problem.

8.6.3 Agriculture

Bilevel programming finds several applications in agriculture, particularly in public policy-making and environmental policies. Several studies focus on optimizing policy instruments, such as taxes, credits, and incentives, with environmental and social goals, anticipating the responses of farmers. The bilevel problems are typically SLMF and are solved by means of SLRs or heuristic algorithms (see Table 8.6.3).

Policy instruments

In their pioneering work, Candler and Norton (1977) define multi-level programming models in the economic policy context. They call “policy problem” and “behavioral problem”, respectively, the UL and LL problems. The former represents the decision problem of policy makers who set policy instruments (e.g., tax rates, incentives). The latter is the decision problem of decentralized agents (e.g., firms, consumers, households) who optimize their economic behavior. The authors propose a linear bilevel model application for a Mexican agricultural production region, where policy makers set subsidies, prices, budgets and taxes on materials and water with societal objectives (e.g., farm employment, income and production maximization) and producers decide on production to maximize their profit. The application (solved by a “hand-made” algorithm inspired by the simplex method) illustrates the different trade-offs between employment and production and shows the potential of bilevel optimization for impacting economic policies.

In Önal et al. (1995), the authors address the optimal allocation of agricultural credits to farm groups in Indonesia. In the UL, the government, acting as the leader, allocates credits with the goal of maximizing the total value of agricultural output of a selected area. The LL problem is a market equilibrium problem for the followers, the farmers, whose goal is to maximize the surplus of producers plus consumers of the agricultural market. The bilevel problem is reformulated as a single-level MILP problem and solved using a heuristic method based on penalty and CP method. Results on a small real-world application show how reallocating credits, in particular decreasing credits allocation to large farmers in favor of small farmers, would pursue both agricultural market growth and equity (net income of small farms increased by 40% while only reducing by 4% the net income of large farms).

Similar conclusions are shown by Whittaker et al. (2017), who study the problem of setting green tax levels on fertilizer use in lands. The government, as leader, determines tax rates to maximize total profit and minimize nitrogen presence in land. Farmers, as followers, make production decisions based on tax rates. The study proposes a bilevel optimization approach solved by a hybrid genetic algorithm (HGA) and applies it to the case study of Calapooia watershed (Oregon, US). Results show that bilevel optimization is effective for geographical targeting agri-environmental policies and open up to the possibility of including additional social objectives in the multi-objective UL.

A similar application of the agri-environmental policy targeting problem for the Raccoon watershed (Iowa, US) is presented by Barnhart et al. (2017). The authors study a deterministic problem with the assumption of the same tax rate for all farmers. They propose three different EAs to identify the Pareto-optimal set of policies for the UL decision maker (the policy maker, who aims at minimizing pollution from fertilizer and maximizing the total profit of farmers). The best method is able to provide a well-established Pareto front of solutions for a large real-world instance with more than 1000 farmers. The authors also study the robust version of the problem considering the production uncertainty, and show more realistic Pareto fronts of solutions at the expense of the leader's objectives.

Agricultural supply chain optimization

Albornoz and Vera (2023) study a stochastic SLSF bilevel problem to address the selective harvest planning problem in a supply chain. In this scenario, the producer serves as the leader, making decisions related to harvest planning and scheduling (including harvest zone selection, quantities, and workforce planning) with the goal of profit maximization. The follower, represented by the wholesaler, determines the quantities to purchase from the producer while minimizing the cost associated with unsatisfied demand. Stochastic yields, demand, and prices are considered in the model. The single-level MILP reformulation of the stochastic model is used to solve a real-world instance for grapes harvesting in Chile. Results show that the bilevel model allows a 10% increase of producer profits compared to a two-stage method where two separate models are solved in a hierarchical perspective.

8.7 Supply chains

Supply Chain Network Design (SCND) is the field related to facility location decisions of a real-world Supply Chain (SC). In recent years, the field has evolved to incorporate environmental and social concerns. In the literature there are several formulations, applications and solution methodologies for sustainable SCND, including bilevel optimization. We review applications of green SC design and planning (see Table 8.7.1) and industrial parks (see Table 8.7.2) related to environmental goals within the SC, namely the reduction of CO_2 emissions and materials waste.

Table 8.7.1: Summary of papers on bilevel applications focused on green supply chain design and planning.

Application	References	Bilevel game		Players		Objectives		Method
		Class	Eq.	UL	LL	UL	LL	
Green supply chain	Avval et al. (2023)	SLMF		Government	SCs decision makers	Min carbon cap	Min cost	ILS
	Camacho-Vallejo et al. (2022)	SLSF		SC distributor	SC manufacturer	**MO:** Max profit; Min CO_2 emissions	Min costst	HEUR
	Camacho-Vallejo et al. (2023)	SLSF		SC distributor	SC manufacturer	**MO:** Max profit; Min CO_2 emissions	Min costst	HEUR
	Cantú et al. (2021)	SLSF		SC designer	SC operational decision maker	**MO:** Min cost; Min GWP	**MO:** Min cost; Min GWP	MOEA+LP
	Cantú et al. (2023)	SLSF		SC designer	SC operational decision maker	**MO:** Min cost; Min GWP	**MO:** Min cost; Min GWP	MOEA+LP
	Ghomi-Avili et al. (2021)	SLSF		SC1	SC2	**MO:** Max profit cost; Min CO_2 emissions	Max profit	SLR+ϵ-constr.
	Golpîra et al. (2017)	SLSF		SC1	SC2	**MO:** Max profit cost; Min CO_2 emissions	Max profit	SLR+ϵ-constr.

Table 8.7.2: Summary of papers on bilevel applications focused on EIPs.

Application	References	Bilevel game		Players		Objectives		Method
		Class	Eq.	UL	LL	UL	LL	
Materials, water, and energy exchange	Ramos et al. (2016b);Ramos et al. (2016a)	SLMF		EIP authority	EIP companies	Min total water consumption	Min cost	SLR
		MLSF		EIP participants	EIP authority	Min cost	Min total water consumption	SLR
	Liberona Henríquez (2022)	SLSF		EIP Authority	EIP participants	Min total water consumption	Min cost	SO
Utility share	Ramos et al. (2018)	SLMF		EIP authority	EIP participants	Min total CO_2 emissions	Min cost	SLR
		MLSF		EIP participants	EIP authority	Min cost	Min total CO_2 emissions	SLR
Carbon incentives	Gu et al. (2020)	SLMF		EIP authority	EIP participants	Max revenue	Min cost	Iterative primal dual

8.7.1 Green supply chain design and planning

In certain scenarios, the government acts as the leader of the bilevel problem influencing the regulations of multiple SCs, whereas the followers are the SC decision makers, namely the managers of the firms that have production and inventory facilities and deliver manufactured products to customers. In Avval et al. (2023), an SLMF bilevel problem is used to establish a cap-and-trade system for supply chains. The government acts as the leader, setting a carbon cap and carbon allowances to encourage SCs to adopt green technologies and reduce carbon emissions. The SCs decision makers act as followers in a Stackelberg game, taking optimal operational and tactical decisions according to their carbon allowances and trading allowances among each other. The government's objective is to minimize the carbon cap to stimulate the SCs to adopt green technologies, while SCs aim to minimize their own total costs. An iterated local search (ILS) is employed to solve small instances of the problem.

In other scenarios, different stakeholders within the same SC can be involved in a bilevel problem, with one acting as the leader and the other as the follower. In Camacho-Vallejo et al. (2022), the leader is a distribution company and the follower is a manufacturing company of the same SC. The distribution company acquires commodities from the manufacturing company and makes decisions on customer selection, routes, and vehicle types with two objectives: profit maximization and CO_2 emissions minimization. The manufacturing company makes the production plan and aims to minimize production and shipping costs without exceeding a maximum pollution rate. A bi-objective TS metaheuristic algorithm is used to find well-approximated emissions/profit Pareto fronts for instances with up to 1000 customers, 7 plants, and 80 vehicles.

An analogous bilevel model is presented in Camacho-Vallejo et al. (2023) in the context of SC restructuring (i.e., manufacturing and service level reduction) after the COVID-19 pandemic and consequent business closures. The authors propose a multi-start heuristic algorithm which is able to solve medium-size instances with up to 700 customers, 60 vehicles and a few manufacturing plants. Results emphasize the strategic significance of achieving a balance between the two UL objectives (profit maximization and CO_2 emissions minimization) for the supply chain leader (distribution company).

Cantú et al. (2021) study a sustainable SCND problem involving energy source, production, transportation and storage decisions of a hydrogen SC, which constitutes a key factor for future energy sources. A bi-objective MILP model minimizing the total daily cost and the GWP of the SC is reformulated as an SLSF bilevel problem. In the UL, the SC designer makes facility location decisions, while production and transportation decisions are addressed in the LL by the operational decision maker of the SC. A hybrid solution strategy combining a multi-objective evolutionary algorithm (MOEA) for the UL problem and a linear program (LP) for the LL problem is proposed and compared with the classical ϵ-constraint method applied to the single-level MILP model. The results on six real-world instances in France show that the bilevel method is able to provide better approximated cost/emissions Pareto fronts of solutions.

Cantú et al. (2023) include additional characteristics and non-linear investment cost functions in the problem they studied in Cantú et al. (2021). They propose an MINLP model and its related bilevel reformulation. An adaptation of their previous hybrid solution strategy (MOEA for the UL and LP for the LL) is shown to be valid since all the non-linearities are included in the UL problem. The robustness of the solution method is proved by tests on the same case study in France.

In the third bilevel SCND scenario we consider, two supply chains can be involved in a hierarchical relationship: one SC acts as the leader (SC1) and the other as the follower (SC2), with uncertainty typically considered in the problem formulation. Ghomi-Avili et al. (2021) present a closed-loop SCND problem (i.e., SC with a reverse flow of products for recycle) with two SCs competing on the same product market in a bilevel framework under demand uncertainty and disruption. The leader (SC1) decides on the facility locations, supplier selection, demand satisfaction, retail price, inventory management, and reverse material flow to maximize profit and minimize CO_2 emissions. The follower (SC2) decides on their demand satisfaction and retail price to maximize profit. A bi-objective KKT reformulation is give for the stochastic bilevel model. To deal with uncertainty, an RO approach is proposed. The resulting robust bi-objective single-level model is solved by means of the ϵ-constraint method. They solve a real-world case study in Tehran (Iran). A similar setting and solution approach are also proposed in Golpîra et al. (2017) for a green opportunistic SCND problem.

Other applications of SCND problems include relief logistics (see, e.g., Safaei et al., 2018) and have been addressed in Section 8.8.4.

8.7.2 Eco-industrial parks

An "Eco-Industrial Park" (EIP) is defined as a system of companies which are located geographically close enough so that materials and energy exchanges are possible and aim to minimize energy and raw materials use, minimize waste, and build sustainable economic, ecological and social relationships like a SC of strictly connected members. Since 2000, and more extensively since 2010, several studies have addressed EIPs optimization problems mainly by means of multi-objective optimization and game theory (see Boix et al., 2015). More recently, bilevel programming has been introduced in the field for modeling two primary types of cooperation among companies in an EIP: (i) the exchange of materials, water, and energy and (ii) the sharing of water and energy units.

Ramos et al. (2016b) introduce a bilevel optimization model for the design and optimization of industrial water networks. They propose two original multi-leader-follower games involving the EIP authority regulating water exchanges and the companies. In the SLMF game, the leader is the authority, which aims at minimizing the total water consumption of the whole system, while the followers are the companies, each of which minimizes its individual total costs. In the MLSF game, the roles are reversed but the goals remain the same. However, in the former case the priority is given to environmental concerns, while the latter prioritizes the economic benefit of the companies in the EIP. The two bilevel problems are both tackled by solving their KKT reformulations using non-convex and non-linear programming solvers. A small literature example with three companies shows that the bilevel approach (in both SLMF and MLSF cases) is able to advantage all the three companies (in terms of water consumption), while the multi-objective optimization approach only favours two of them. Similar results have been shown in Ramos et al. (2016a).

More recently, Liberona Henríquez (2022) formulates the stochastic EIP bilevel model for the water exchange network on a directed graph, where the EIP authority designs and operates the system while the EIP participants consume water and reintroduce partially contaminated water in the system. The amount of contaminant produced by every EIP participant is uncertain in the design phase of the EIP water exchange network (UL problem). The leader is the authority, which aims at minimizing the total water consumption of the whole system, while the followers are the companies, which minimize their total costs. The companies can buy fresh water and receive partially polluted water from other companies. The bilevel problem is solved as an SO problem with recourse.

For EIP utility share, we only mention the relevant work by Ramos et al. (2018), who propose a multi-leader-follower game for the design of the utility network (i.e., heat exchange) and introduce the concept of environmental authority (the EIP authority with the goal of minimizing the equivalent CO_2 consumption from utility consumption). The companies minimize their total annualized costs. Again, the authors formulate the SLMF and MLSF games and then solve their KKT reformulations.

Incentives for emissions reduction and energy-saving practices are also gaining attraction within EIPs. Gu et al. (2020) introduce energy price incentives for economic and environmental benefits of the partners of an EIP under the Chinese real-time multi-energy system. An SLMF bilevel model is proposed, where the leader is the authority of the energy system setting the prices with carbon emissions constraints and an economic goal (i.e., maximum revenue) and the followers are the energy users participating in the EIP, whose goal is to minimize costs. The authors propose an iterative primal dual procedure to find global optimal solutions. The case studies show that this energy price system can positively affect both the economic benefit of the energy users and the environmental impact of the EIP.

8.8 Disaster prevention and response

Disaster prevention include the transportation of hazardous material (called "hazmat" hereafter), that encompasses substances such as flammable and explosive materials, posing significant human and environmental risks (see Table 8.8.1), tolls (see Table 8.8.3), and other governmental policy tools (see Table 8.8.3) introduced to address and mitigate these risks. Disaster response include location, supply, and material distribution problems for emergency planning and relief logistics (see Table 8.8.4).

8.8.1 Hazmat transportation

In the domain of hazardous material transportation, road segment selection and network design are pivotal aspects. The main problem in the field is the Hazardous Materials Transportation Network Design Problem (HTNDP), which has been addressed with several bilevel programming approaches.

In the seminal work by Kara and Verter (2004), the authors formulate the HTNDP under a bilevel framework where the leader is the government authority and the followers are the carriers. The leader determines road segments to be closed for carriers to minimize total risk (expressed as the population exposure to an hazmat incident), while followers make route choices to minimize their individual total traveled distances. The bilevel model is reformulated as a single-level MILP model. Results on a medium-sized case study in Western Ontario (Canada) show that a win-win situation exists

Table 8.8.1: Summary of papers on bilevel applications focused on hazmat transportation.

Application	References	Bilevel game		Players		Objectives		Method
		Class	Eq.	UL	LL	UL	LL	
Hazmat transportation	Kara and Verter (2004)	SLMF		Central authority	Carriers	Min total risk	Min distance	SLR
	Gzara (2013)	SLMF		Central authority	Carriers	Min total risk	Min distance	HCP
	Erkut and Alp (2007)	SLMF		Central authority	Carriers	Min total risk	Min distance	GIA
	Erkut and Gzara (2008)	SLMF		Central authority	Carriers	**MO:** Min total risk, Min total cost	Min distance	HEUR
	Bianco et al. (2009)	SLSF		Meta-local authority	Regional authority	Max risk equity	Min total risk	HEUR
	Taslimi et al. (2017)	SLMF		Central authority	Carriers	Min max risk	Min transportation cost	SLR, HEUR
	Xin et al. (2013)	SLMF		Central authority	Carriers	Min total risk	Min distance	RH
	Sun et al. (2016)	SLMF		Central authority	Carriers	Min total risk	Min distance	RO + HEUR
	X. Liu and Kwon (2020)	SLMF		Central authority	Carriers	**MO:** Min setup cost, Min risk	Min transportation cost	CP+BD
	Taslimi et al. (2017)	SLMF		Central authority	Carriers	Min max risk	Min transportation cost	SLR, HEUR
	Esfandeh et al. (2018)	SLMF		Central authority	Carriers	Min risk	Min transportation cost	HEUR

Table 8.8.2: Summary of papers on bilevel applications focused on hazmat toll setting.

Application	References	Bilevel game		Players		Objectives		Method
		Class	Eq.	UL	LL	UL	LL	
Hazmat toll setting	Marcotte et al. (2009)	SLMF		Government	Carriers	Min travel risk and cost	Max utility	SLR
	Assadipour et al. (2016)	SLMF		Government	Carriers	**MO:** Min travel risk, Min travel cost	Min cost	HA
	López-Ramos et al. (2019)	SLMF	✓	Road network operator	Vehicles	Max profit	Min cost	SLR

Table 8.8.3: Summary of papers on bilevel applications focused on hazmat control policies.

Application	References	Bilevel game		Players		Objectives		Method
		Class	Eq.	UL	LL	UL	LL	
Hazmat control policies	Chiou (2016)	SLMF	✓	Road network operator	Vehicles	Min total travel delay	Min risk	Iterative cutting plane
	Chiou (2017)	SLMF	✓	Road network operator	Vehicles	Min total travel delay	Min risk	Iterative cutting plane
	Amouzegar and Moshirvaziri (1999)	SLMF		Central authority	Firms	Min total costs	Min cost	HEUR
	Bhavsar and Verma (2022)	SLSF		Government	Railroad operator	Min risk	Min cost	SLR

Table 8.8.4: Summary of papers on bilevel applications focused on emergency planning and response.

Application	References	Bilevel game		Players		Objectives		Method
		Class	Eq.	UL	LL	UL	LL	
Emergency planning and disaster response	Y. Liu and Luo (2012)	SLMF	✓	Policy maker/Emergency manager	Evacuees	Min total evacuation cost	Min perceived travel time	GA
	Yi et al. (2017)	SLMF	✓	Gvernment emergency management agency	Residents	Min weighted travel time and risk	Min travel time	HEUR
	Safaei et al. (2018)	SLSF		Relief SC designer	Relief SC operational decision maker	Min total relief SC cost	Min total supply risk	SLR
	Gutjahr and Dzubur (2016)	SLMF	✓	Aid-providing organization	Beneficiaries	**MO:** Min opening cost; Min uncovered demand	Min weighted travel and unmet demand cost	ϵ-constraint + B&B + FWA
	Li and Teo (2019)	SLSF		Emergency command center	Distribution centers administrator	Max road network accessibility	**MO:** Max centers satisfaction; Min total delivery time	HGA
	Gao (2022)	SLSF		Relief network designer	Relief network operational decision maker	Min total dissatisfaction level	Min total expected transportation time	SLR

for government and carriers, since both the risk and total distance decrease when a shipment-specific regulation for road segment selection is applied to the design of the network. In Gzara (2013), the authors provide an exact cutting plane (CP) method and a family of valid cuts for the bilevel HTNDP.

In Erkut and Alp (2007), the authors restrict the general HTNDP by Kara and Verter (2004) to the case where only one route between any given origin and destination is allowed, resulting in a minimum risk hazmat tree network problem. They propose a greedy insertion algorithm (GIA) that iteratively adds paths to an optimal tree network, which is found by solving the SLR of the problem. They test the algorithm on a case study in Ravenna (Italy), consisting of a large network of around 100 nodes. They find different greedy solutions with variable risk and cost with respect to the unregulated solution, one of which reduces risk by 60% with only a 6% increase in costs.

In Erkut and Gzara (2008), the authors extend the multi-commodity HTNDP by Kara and Verter (2004) to the case of an undirected network and propose a bi-objective UL model where the leader (the government) aims at minimizing both costs and risk. Different risks measures are proposed for a heuristic solution method. Results on the case study of Ravenna as well as on random instances reveal that the heuristic is effective and time-efficient, both for the single-objective and the bi-objective problem.

A different bilevel HTNDP, where the leader and the follower are, respectively, a local and a regional authority, is explored by Bianco et al. (2009). The goal is to maximize risk equity across regions for the local authority and minimize total risk for the regional authority. The study provides an iterative heuristic method, demonstrating its effectiveness on a case study in Rome (Italy).

Addressing risk uncertainty is essential in hazmat transportation, where several external factors such as climate and road conditions can impact the hazmat risk. In the uncertain HTNDP, an interval of risk values is given for each arc of the network. In Xin et al. (2013), intervals of risk values are introduced in the model proposed by Kara and Verter (2004) such that each commodity on each network arc has its own set of possible values of risk. The leader is the government, whose goal is minimize the total risk chosen by the carriers, the followers, who in turn minimize their cost. The authors provide a robust heuristic (RH) approach based on the minimax regret criterion that is used to find the robust risk shortest path for each commodity. The robust network is compared with the deterministic one on a small-size real-world case study in China for the transportation of solid and gas hazmat, showing good-quality solutions.

Sun et al. (2016) present two robust bilevel models for the HTNDP, wherein the risk and risk uncertainty associated with each network arc are defined for each arc of the network as intervals of values. The distinction is made between homogeneous and heterogeneous risk scenarios for all shipments in the two models. Their models incorporate an uncertainty budget, denoting the total number of arcs exposed to uncertainty for all shipments. A heuristic algorithm based on a Lagrangian relaxation is proposed and tested on the Ravenna case study from Kara and Verter (2004) and on larger instances for the city of Barcelona (Spain).

X. Liu and Kwon (2020) combine the robust HTNDP and hazmat facility location under a bilevel framework, introducing uncertainty for hazmat risk and transportation demand. The leader decides which facilities to open and which road segments to close with the goal of minimizing the facility setup cost and the hazmat exposure risk. The followers decides on their routes to minimize transportation costs, as in the classical HTNDP. The worst-case scenario is considered using an RO approach. The author adopt the CP method by Gzara (2013), combine it with a Benders' decomposition (BD), and include uncertainty. The resulting exact method is proven to be effective compared to the solution of the SLR by Gurobi. These studies collectively contribute to the understanding and optimization of the HTNDP, addressing environmental and social perspectives. Other HTNDP extensions include, for example, equity risk (see Taslimi et al., 2017) and time-dependent road closure policies (see Esfandeh et al., 2018).

A network interdiction problem (I) is a bilevel problem where the leader and the follower have diametrically opposite objectives. The leader (the "attacker") takes interdiction actions to maximize the minimum objective that the follower (the "defender") can obtain in solving its optimization problem. It is a resource-constrained network interdiction problem when the attacker has limited interdiction actions. We refer the reader to the survey by Smith and Song (2020).

8.8.2 Hazmat toll setting

Toll setting in hazmat transportation is an alternative tool for policy makers and road network operators to contain the hazmat risk exposure for road users. Marcotte et al. (2009) compare the bilevel HTNDP model by Kara and Verter (2004), in which road segments are closed to hazmat carriers, with their toll setting model, where tolls are applied to roads for hazmat carriers. The leader (government) sets tolls on roads to minimize a combination of travel risk and cost, while followers (carriers) decide their shipment routes to minimize their utility (defined by minimum distance and risk). The study reveals that the toll setting model gives a lower optimal risk than network design model when the same hazmat is carried by more than one carriers.

In their application on rail-truck intermodal transportation, Assadipour et al. (2016) introduce a bi-objective bilevel model where the government (the leader) deters the carriers from using certain rail intermodal terminals by assigning a toll to each hazmat container passing through them, and the followers (the carriers) select the routes to minimize their costs. The UL problem is bi-objective: the risk is minimized as well as the total toll cost (to make the toll policy attractive). They address the problem with an HA where a GA for multi-objective optimization is combined with the optimal solution of the LL problem. A case study in US is solved by comparing the toll setting model with the network design model (where the government closes some terminals instead of setting tolls). Different results suggest that a two-stage procedure (toll setting first and network design after) obtains promising solutions.

Network design and toll setting are sometimes integrated into a unique bilevel settings. In the SLSF model by López-Ramos et al. (2019), the leader is a road network operator, who maximizes its profit (toll income minus costs from roads construction and risk exposure to hazmat transportation) and the followers are represented by the vehicles deciding on routes to minimize their travel costs (in a user equilibrium problem). The authors reformulate and linearize their mixed-integer non linear bilevel problem as a single-level MILP, which is then heuristically solved on benchmark instances.

8.8.3 Hazmat control policies

Other bilevel applications for hazmat transportation deal with control policies. Chiou (2016) introduces non-linear bilevel models for hazmat transportation with signal-controlled road networks. The aim is to regulate hazmat traffic by setting signals so that total travel cost and public risk exposure can be minimized simultaneously in the UL. The LL problem involves a traffic equilibrium. The authors propose an iterative cutting plane method based on bundling subgradients from previous iterations. Chiou (2017) extend the problem to incorporate travel demand uncertainty.

Addressing pollution control policies, Amouzegar and Moshirvaziri (1999) present a bilevel model for hazardous waste management (see Section 8.6.2). In this model, prices and taxes are set by the central authority (leader) on pollution levels to mitigate the harmful policies of the firms (followers).

Subsidies emerge as another effective tool for hazmat transportation risk control. Bhavsar and Verma (2022) introduce an SLSF model, where the government (the leader) offers subsidies to induce the railroad operator (the follower) to take alternative routes that are away from high-risk network links. The government aims at minimizing risk, while the railroad operator minimizes their total net cost. The authors present a KKT reformulation, applied to a real case in US. They demonstrate that even modest subsidies can result in significant risk reduction.

8.8.4 Emergency planning and disaster response

Y. Liu and Luo (2012) study an SLMF bilevel network optimization problem for the emergency evacuation process, in which signalized and uninterrupted flow intersections in the evacuation network must be optimally located for traffic crossing-elimination and signal control. In the UL problem, the policy maker or emergency manager decides the intersections location to minimize the total evacuation cost. In the LL problem, modelled as a stochastic user equilibrium problem, evacuees make evacuation routes decisions to minimize their perceived travel cost. The authors propose a GA and solve a case study in China showing total evacuation time reductions up to 40%.

Another critical issue in evacuation is evacuation orders. Yi et al. (2017) present a bilevel model with a similar structure (SLMF with dynamic traffic equilibrium in the LL), where the leader in the UL is the government emergency management agency deciding when and where to issue orders to minimize a weighted sum of total travel times and risks. The followers in the LL are the residents deciding if, when, and where to evacuate (routes) to minimize their travel time. The UL is a multi-stage stochastic program with hurricane occurrence uncertainty. The authors propose a heuristic algorithm based on Lagrangian relaxation through scenario decomposition and solve a large case study in North Carolina (US) for different hurricane scenarios, showing the benefit of constructing a contingent evacuation policy.

Natural disaster require efficient relief supply chains to provide relief commodities to communities, including food, clothing, and medicines, among many. Safaei et al. (2018) propose a robust bilevel SLSF model to optimize the supply/demand process of relief commodities under uncertainty. In the UL, the leader decides the locations of transshipment transfer depots close to the disaster areas for collecting commodities from central warehouse and transfer decisions to the disaster areas. The UL goal is to minimize the total relief supply chain costs. In the LL, the transfer of commodities from suppliers to the central warehouse are determined to minimize the total supply risk over all suppliers while satisfying demand. The authors employ the KKT conditions to obtain an SLR solvable by commercial solvers. Results on flood disaster real scenarios in Iran demonstrate the effectiveness of the proposed method.

Other works deal with post-disaster situations. Gutjahr and Dzubur (2016) study the FLP of distribution centers for a natural disaster response humanitarian SC as an SLMF bilevel problem. The leader is the aid-providing organization, making location opening decisions to minimize the total opening cost and total uncovered demand. The followers are the beneficiaries making supply decisions (i.e., in which distribution center to go and with what demand) in a traffic user equilibrium model which minimizes the weighted sum of travel cost and cost for not being supplied with certainty. The authors propose an exact approach based on the ϵ-constraint method for the bi-objective UL, B&B, and the Frank-Wolfe algorithm (FWA) for the LL Wardrop equilibrium. Medium-size instances for a case study of rural communities in Senegal are solved to optimality, showing reductions of unmet demand up to 40%.

Li and Teo (2019) formulate the multi-period bilevel road network repair work scheduling and relief logistics problem for earthquake disaster relief. In the UL, the leader is the emergency command center making the restoration strategy (i.e., repair crew assignment and routing) for each period to maximize the road network accessibility. In the LL, the follower is the distribution centers administrator, who make relief logistics and relief material delivery decisions according to the each period UL strategy to maximize satisfaction of the centers (given a definition of maximum relative satisfaction degree) and minimize the total delivery time. The proposed HGA is able to solve a case study in China in a short computational time, which is crucial for post-disaster optimization, with a resulting 15% level of road repair at the end of the time horizon.

Another problem in post-disaster relief is the multi-commodity rebalancing problem, that allows to rebalance surplus and shortage of relief commodities over the different over the transportation network to satisfy the potential demand at all relief centers. Gao (2022) present a stochastic bilevel model for the relief multi-commodity rebalancing problem under demand and transportation network availability uncertainty. In the UL, the incoming and outgoing shipments at relief centers are determined to achieve fairness between the centers by minimizing the total dissatisfaction level. In the LL, the routing decisions are made to minimize the expected total transportation time. The authors solve the SLR on a earthquake case study in China showing applicable and effective decision-making results.

8.9 Conclusions and future research directions

Bilevel optimization models hierarchical decision-making processes involving multiple actors. The field rapidly expands in both solution methodologies and applications, approaching different real-world situations involving policy makers, industrial entities, and end users. In alignment with contemporary public and private concerns surrounding the “3P” framework, that stands for “profit, planet, people”, the paper analyzes the literature on bilevel applications with sustainability perspectives, considering the three-dimensional aspect of the term. By delving into the literature, we identify bilevel applications that incorporate at least two dimensions of sustainability – economic, environmental, and social – within the upper or lower levels of the hierarchical game structure. Particularly, we study the players of these bilevel games, their decisions and goals, along with the proposed solution methods, and the resulting insights on real-world case studies for transportation and logistics (Section 8.4), production planning and manufacturing (Section 8.5), waste, water, and material management (Section 8.6), and supply chains (Section 8.7). Applications cover renowned mathematical programming problems, such as assignment, location, and routing problems among many, which have only recently been extended to incorporate environmental and social optimization.

In bilevel applications, leaders typically emerge as authorities designing service systems for the private/public domain or companies optimizing their businesses to provide consumers with products. Optimization problems encompass diverse decision-making processes, ranging from carbon policies to green vehicle and route selection, waste reduction, environmental preservation, clean energy production, health risk mitigation from hazardous materials, to emergency logistics optimization. A recurring pattern observed in our analysis is that the upper-level problem, typically led by an authority, incorporates green and social goals alongside economic ones. Conversely, the lower-level problem of end users usually prioritizes efficiency or profitability. Numerous case studies covering industrial and public environments across various countries in Europe, America, Asia, and Africa showcase the potential of bilevel optimization in achieving sustainability goals. Positive outcomes include reduced carbon emissions, lowered energy consumption combined with an increased renewable energy mix, and maximized social benefits measured in terms of people's preferences and health risk mitigation.

Despite the wealth of applications, there remain aspects that can be further investigated. We sketch possible directions for future research in the field of bilevel optimization with sustainability perspectives:

1. Given the variety of routing problems studied in the literature, the field of bilevel green routing can be further explored. The few works reported in the paper open room for future works on hierarchical routing decisions, such as the introduction of green policies and vehicles selection for routing plans with reduced carbon emissions.

2. Future research should focus on developing frameworks that integrate environmental and social perspectives into the lower level of bilevel optimization models, addressing issues such as environmental awareness, community well-being, social equity, and the broader societal impacts of business decisions. This would allow to investigate the role of end users, including companies, consumers, and citizens in shaping sustainable practices, acknowledging not only external influences like public policies but also the individual motivations and values that drive decision-making.

3. The exploration of the social dimension in bilevel optimization is still young. Researchers should investigate how decisions, especially of the leader of the bilevel game, impact various social aspects, including equity, accessibility, and well-being considerations, as well as stakeholder engagement and corporate responsibility.

This involves the incorporation of more social indicators and metrics into bilevel goals.

4. In particular, fairness in transportation has been neglected so far in bilevel optimization applications for transportation and logistics. Interesting future research should investigate the modeling of fairness-related requirements and objective functions to consider leader's social optimum that is also fair in terms of, for instance, the difference between the maximum and minimum travel time of followers.

5. Other sectors, such as healthcare and medicine, should be explored with bilevel optimization.

6. Another important sector that, due to the large number of possible applications, remained out of scope of this study, is namely electricity and energy markets. Future research will be dedicated to the analysis of green energy applications.

In summary, bilevel optimization has already made substantial contributions to sustainable decision-making, and we expect that this collection of works not only consolidates existing knowledge but also lays the foundation for future developments in the research field and support practitioners in more structured and holistic decisions.

References

Albornoz, V. M., & Vera, P. I. (2023). Coordinating harvest planning and scheduling in an agricultural supply chain through a stochastic bilevel programming. *International Transactions in Operational Research*, *30*(4), 1819–1842.

Almutairi, H., & Elhedhli, S. (2014). Carbon tax based on the emission factor: A bilevel programming approach. *Journal of Global Optimization*, *58*, 795–815.

Amouzegar, M. A., & Moshirvaziri, K. (1999). Determining optimal pollution control policies: An application of bilevel programming. *European Journal of Operational Research*, *119*(1), 100–120.

Anandalingam, G., & Apprey, V. (1991). Multi-level programming and conflict resolution. *European Journal of Operational Research*, *51*(2), 233–247.

Angulo, E., Castillo, E., García-Ródenas, R., & Sánchez-Vizcaíno, J. (2014). A continuous bi-level model for the expansion of highway networks. *Computers & Operations Research*, *41*, 262–276.

Assadipour, G., Ke, G. Y., & Verma, M. (2016). A toll-based bi-level programming approach to managing hazardous materials shipments over an intermodal transportation network. *Transportation Research Part D: Transport and Environment*, *47*, 208–221.

Aussel, D., & Svensson, A. (2020). A Short State of the Art on Multi-Leader-Follower Games. In S. Dempe & A. Zemkoho (Eds.), *Bilevel Optimization: Advances and next challenges* (pp. 53–76). Springer.

Avval, A. E., Dehghanian, F., & Pirayesh, M. (2023). Joint optimization of multiple supply chains under cap-and-trade regulation: A bi-level programming model and solution algorithm. *Asia-Pacific Journal of Operational Research*, *40*(02), 2250019.

Bagloee, S. A., Sarvi, M., & Wallace, M. (2016). Bicycle lane priority: Promoting bicycle as a green mode even in congested urban area. *Transportation Research Part A: Policy and Practice*, *87*, 102–121.

Barbosa-Póvoa, A. P., da Silva, C., & Carvalho, A. (2018). Opportunities and challenges in sustainable supply chain: An operations research perspective. *European Journal of Operational Research*, *268*(2), 399–431.

Barnhart, B., Lu, Z., Bostian, M., Sinha, A., Deb, K., Kurkalova, L., Jha, M., & Whittaker, G. (2017). Handling practicalities in agricultural policy optimization for water quality improvements. *Proceedings of the Genetic and Evolutionary Computation Conference*, 1065–1072.

Beck, Y., Ljubić, I., & Schmidt, M. (2023). A survey on bilevel optimization under uncertainty. *European Journal of Operational Research*, *311*(2), 401–426.

Bektaş, T., & Laporte, G. (2011). The pollution-routing problem. *Transportation Research Part B: Methodological*, *45*(8), 1232–1250.

Ben-Ayed, O., Blair, C. E., Boyce, D. E., & LeBlanc, L. J. (1992). Construction of a real-world bilevel linear programming model of the highway network design problem. *Annals of Operations Research*, *34*, 219–254.

Bhavsar, N., & Verma, M. (2022). A subsidy policy to managing hazmat risk in railroad transportation network. *European Journal of Operational Research*, *300*(2), 633–646.

Bianco, L., Caramia, M., & Giordani, S. (2009). A bilevel flow model for hazmat transportation network design. *Transportation Research Part C: Emerging Technologies*, *17*(2), 175–196.

Boix, M., Montastruc, L., Azzaro-Pantel, C., & Domenech, S. (2015). Optimization methods applied to the design of eco-industrial parks: A literature review. *Journal of Cleaner Production*, *87*, 303–317.

Bracken, J., & McGill, J. T. (1973). Mathematical programs with optimization problems in the constraints. *Operations Research*, *21*(1), 37–44.

Brotcorne, L., Labbé, M., Marcotte, P., & Savard, G. (2008). Joint design and pricing on a network. *Operations Research*, *56*(5), 1104–1115.

Brotcorne, L., Marcotte, P., & Savard, G. (2008). Bilevel programming: The montreal school. *INFOR: Information Systems and Operational Research*, *46*(4), 231–246.

Calvete, H. I., Galé, C., Iranzo, J. A., & Mateo, P. M. (2023). A decision tool based on bilevel optimization for the allocation of water resources in a hierarchical system. *International Transactions in Operational Research*, *30*(4), 1673–1702.

Calvete, H. I., Galé, C., Iranzo, J. A., & Toth, P. (2023). The school bus routing problem with student choice: A bilevel approach and a simple and effective metaheuristic. *International Transactions in Operational Research*, *30*(2), 1092–1119.

Camacho-Vallejo, J.-F., Corpus, C., & Villegas, J. G. (2024). Metaheuristics for bilevel optimization: A comprehensive review. *Computers & Operations Research*, *161*, 106410.

Camacho-Vallejo, J.-F., Dávila, D., & Nucamendi-Guillén, S. (2023). A hierarchized green supply chain with customer selection, routing, and nearshoring. *Computers & Industrial Engineering*, *178*, 109151.

Camacho-Vallejo, J.-F., López-Vera, L., Smith, A. E., & González-Velarde, J.-L. (2022). A tabu search algorithm to solve a green logistics bi-objective bi-level problem. *Annals of Operations Research*, *316*(2), 927–953.

Candler, W., & Norton, R. (1977). *Multi-level programming and development policy* (tech. rep. No. SWP 258). World Bank Group. Washington, D.C.

Cantú, V. H., Azzaro-Pantel, C., & Ponsich, A. (2021). A novel matheuristic based on bi-level optimization for the multi-objective design of hydrogen supply chains. *Computers & Chemical Engineering*, *152*, 107370.

Cantú, V. H., Ponsich, A., Azzaro-Pantel, C., & Carrera, E. (2023). Capturing spatial, time-wise and technological detail in hydrogen supply chains: A bi-level multi-objective optimization approach. *Applied Energy*, *344*, 121159.

Caramia, M., & Pizzari, E. (2022). Novel bilevel formulations for waste management. *Discrete Applied Mathematics*.

Chen, R., Qian, X., Miao, L., & Ukkusuri, S. V. (2020). Optimal charging facility location and capacity for electric vehicles considering route choice and charging time equilibrium. *Computers & Operations Research*, *113*, 104776.

Chiou, S.-W. (2016). A bi-objective bi-level signal control policy for transport of hazardous materials in urban road networks. *Transportation Research Part D: Transport and Environment*, *42*, 16–44.

Chiou, S.-W. (2017). A risk-averse signal setting policy for regulating hazardous material transportation under uncertain travel demand. *Transportation Research Part D: Transport and Environment*, *50*, 446–472.

Cordeau, J.-F., Klibi, W., & Nickel, S. (2021). Logistics network design. In T. G. Crainic, M. Gendreau, & B. Gendron (Eds.), *Network design with applications to transportation and logistics* (pp. 599–625). Springer International Publishing.

Cordeau, J.-F., Laporte, G., Savelsbergh, M. W., & Vigo, D. (2007). Vehicle routing. In C. Barnhart & G. Laporte (Eds.), *Transportation* (pp. 367–428, Vol. 14). Elsevier.

Dempe, S. (2020a). *Bilevel optimization: Advances and next challenges*. Springer International Publishing.

Dempe, S. (2020b). Bilevel optimization: Theory, algorithms, applications and a bibliography. In S. Dempe & A. Zemkoho (Eds.), *Bilevel optimization: Advances and next challenges* (pp. 581–672). Springer International Publishing.

Elkington, J. (2002). *Cannilbals with forks: The triple bottom line of the 21st century*. Oxford Press.

Erkut, E., & Alp, O. (2007). Designing a road network for hazardous materials shipments. *Computers & Operations Research*, *34*(5), 1389–1405.

Erkut, E., & Gzara, F. (2008). Solving the hazmat transport network design problem. *Computers & Operations Research*, *35*(7), 2234–2247.

Esfandeh, T., Batta, R., & Kwon, C. (2018). Time-dependent hazardous-materials network design problem. *Transportation Science*, *52*(2), 454–473.

Gang, J., Tu, Y., Lev, B., Xu, J., Shen, W., & Yao, L. (2015). A multi-objective bi-level location planning problem for stone industrial parks. *Computers & Operations Research*, *56*, 8–21.

Gao, X. (2022). A bi-level stochastic optimization model for multi-commodity rebalancing under uncertainty in disaster response. *Annals of Operations Research*, *319*(1), 115–148.

Gaspar, I., Benavente, J., Bordagaray, M., Alonso, B., Moura, J. L., & Ibeas, Á. (2015). A bilevel mathematical programming model to optimize the design of cycle paths. *Transportation Research Procedia*, *10*, 423–432.

Ghomi-Avili, M., Tavakkoli-Moghaddam, R., Jalali Naeini, S. G., & Jabbarzadeh, A. (2021). Competitive green supply chain network design model considering inventory decisions under uncertainty: A real case of a filter company. *International Journal of Production Research*, *59*(14), 4248–4267.

Golpîra, H., Najafi, E., Zandieh, M., & Sadi-Nezhad, S. (2017). Robust bi-level optimization for green opportunistic supply chain network design problem against uncertainty and environmental risk. *Computers & Industrial Engineering*, *107*, 301–312.

Grimm, V., Nowak, D., Schewe, L., Schmidt, M., Schwartz, A., & Zöttl, G. (2022). A tractable multi-leader multi-follower peak-load-pricing model with strategic interaction. *Mathematical Programming*, *195*, 605–647.

Gu, H., Li, Y., Yu, J., Wu, C., Song, T., & Xu, J. (2020). Bi-level optimal low-carbon economic dispatch for an industrial park with consideration of multi-energy price incentives. *Applied Energy*, *262*, 114276.

Gutjahr, W. J., & Dzubur, N. (2016). Bi-objective bilevel optimization of distribution center locations considering user equilibria. *Transportation Research Part E: Logistics and Transportation Review*, *85*, 1–22.

Gzara, F. (2013). A cutting plane approach for bilevel hazardous material transport network design. *Operations Research Letters*, *41*(1), 40–46.

Hansen, P., Jaumard, B., & Savard, G. (1992). New branch-and-bound rules for linear bilevel programming. *SIAM Journal on scientific and Statistical Computing*, *13*(5), 1194–1217.

He, L., Yao, L., & Sabev Varbanov, P. (2023). A bi-level optimization approach to reduce the pollution burden of lake water with ecological compensation. *Ecological Indicators*, *151*, 110334.

Hong, Z., Chu, C., Zhang, L. L., & Yu, Y. (2017). Optimizing an emission trading scheme for local governments: A stackelberg game model and hybrid algorithm. *International Journal of Production Economics*, *193*, 172–182.

Hu, Z., Wang, L., Qin, J., Lev, B., & Gan, L. (2022). Optimization of facility location and size problem based on bi-level multi-objective programming. *Computers & Operations Research*, *145*, 105860.

Jaehn, F. (2016). Sustainable operations. *European Journal of Operational Research*, *253*(2), 243–264.

Kara, B. Y., & Verter, V. (2004). Designing a road network for hazardous materials transportation. *Transportation Science*, *38*(2), 188–196.

Kleinert, T., Labbé, M., Ljubic, I., & Schmidt, M. (2021). A survey on mixed-integer programming techniques in bilevel optimization. *EURO Journal on Computational Optimization*, *9*, 100007.

Laporte, G., Nickel, S., & Saldanha da Gama, F. (2019). *Location science*. Springer.

Li, S., & Teo, K. L. (2019). Post-disaster multi-period road network repair: Work scheduling and relief logistics optimization. *Annals of Operations Research*, *283*(1-2), 1345–1385.

Liberona Henríquez, G. (2022). *Applications of stochastic and bilevel optimization to network problems* [Doctoral dissertation, Universidad de Chile].

Liu, S., Shen, Z. M., & Ji, X. (2022). Urban bike lane planning with bike trajectories: Models, algorithms, and a real-world case study. *Manufacturing & Service Operations Management*, *24*(5), 2500–2515.

Liu, X., & Kwon, C. (2020). Exact robust solutions for the combined facility location and network design problem in hazardous materials transportation. *IISE Transactions*, *52*(10), 1156–1172.

Liu, Y., & Luo, Z. (2012). A bi-level model for planning signalized and uninterrupted flow intersections in an evacuation network. *Computer-Aided Civil and Infrastructure Engineering*, *27*(10), 731–747.

López-Ramos, F., Nasini, S., & Guarnaschelli, A. (2019). Road network pricing and design for ordinary and hazmat vehicles: Integrated model and specialized local search. *Computers & Operations Research*, *109*, 170–187.

Ma, S., Chen, S., & Cai, X. (2018). An incentive-based bi-level optimization model for collaborative green product line design. *2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 981–985.

Marcotte, P., Mercier, A., Savard, G., & Verter, V. (2009). Toll policies for mitigating hazardous materials transport risk. *Transportation Science*, *43*(2), 228–243.

Nath, R., Rauniyar, A., Muhuri, P. K., & Shukla, A. K. (2018). A novel bilevel formulation for pollution routing problem. *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, 586–562.

Önal, H., Darmawan, D. H., & Johnson, S. H. (1995). A multilevel analysis of agricultural credit distribution in east java, indonesia. *Computers & Operations Research*, *22*(2), 227–236.

Parvasi, S. P., Mahmoodjanloo, M., & Setak, M. (2017). A bi-level school bus routing problem with bus stops selection and possibility of demand outsourcing. *Applied Soft Computing*, *61*, 222–238.

Qiu, R., Xu, J., Ke, R., Zeng, Z., & Wang, Y. (2020). Carbon pricing initiatives-based bi-level pollution routing problem. *European Journal of Operational Research*, *286*(1), 203–217.

Qiu, R., Xu, J., Xie, H., Zeng, Z., & Lv, C. (2020). Carbon tax incentive policy towards air passenger transport carbon emissions reduction. *Transportation Research Part D: Transport and Environment*, *85*, 102441.

Ramos, M., Boix, M., Aussel, D., Montastruc, L., & Domenech, S. (2016a). Optimal design of water exchanges in eco-industrial parks through a game theory approach. In Z. Kravanja & M. Bogataj (Eds.), *26th european symposium on computer aided process engineering* (pp. 1177–1182, Vol. 38). Elsevier.

Ramos, M., Boix, M., Aussel, D., Montastruc, L., & Domenech, S. (2016b). Water integration in eco-industrial parks using a multi-leader-follower approach. *Computers & Chemical Engineering*, *87*, 190–207.

Ramos, M., Rocafull, M., Boix, M., Aussel, D., Montastruc, L., & Domenech, S. (2018). Utility network optimization in eco-industrial parks by a multi-leader follower game methodology. *Computers & Chemical Engineering*, *112*, 132–153.

Rashidi, E., Parsafard, M., Medal, H., & Li, X. (2016). Optimal traffic calming: A mixed-integer bi-level programming model for locating sidewalks and crosswalks in a multimodal transportation network to maximize pedestrians' safety and network usability. *Transportation Research Part E: Logistics and Transportation Review*, *91*, 33–50.

Safaei, A. S., Farsad, S., & Paydar, M. M. (2018). Robust bi-level optimization of relief logistics operations. *Applied Mathematical Modelling*, *56*, 359–380.

Sanjeet Singh, N. H., & Bhattacharya, A. (2018). Offshore manufacturing contract design based on transfer price considering green tax: A bilevel programming approach. *International Journal of Production Research*, *56*(5), 1825–1849.

Sinha, A., Malo, P., & Deb, K. (2015). Transportation policy formulation as a multi-objective bilevel optimization problem. *IEEE Congress on Evolutionary Computation, CEC 2015, Sendai, Japan, May 25-28, 2015*, 1651–1658.

Sinha, A., Malo, P., & Deb, K. (2018). A review on bilevel optimization: From classical to evolutionary approaches and applications. *IEEE Transactions on Evolutionary Computation*, *22*(2), 276–295.

Sinha, A., Malo, P., Frantsev, A., & Deb, K. (2013). Multi-objective stackelberg game between a regulating authority and a mining company: A case study in environmental economics. *2013 IEEE Congress on Evolutionary Computation*, 478–485.

Smith, J. C., & Song, Y. (2020). A survey of network interdiction models and algorithms. *European Journal of Operational Research*, *283*(3), 797–811.

Sohn, K. (2011). Multi-objective optimization of a road diet network design. *Transportation Research Part A: Policy and Practice*, *45*(6), 499–511.

Sun, L., Karwan, M. H., & Kwon, C. (2016). Robust hazmat network design problems considering risk uncertainty. *Transportation Science*, *50*(4), 1188–1203.

Taslimi, M., Batta, R., & Kwon, C. (2017). A comprehensive modeling framework for hazmat network design, hazmat response team location, and equity of risk. *Computers & Operations Research*, *79*, 119–130.

von Stackelberg, H. (1934). *Marktform und gleichgewicht.* Springer.

von Stackelberg, H. (1952). *The theory of the market economy.* Oxford University.

Wang, J., Ehrgott, M., Dirks, K. N., & Gupta, A. (2014). A bilevel multi-objective road pricing model for economic, environmental and health sustainability. *Transportation Research Procedia*, *3*, 393–402.

Wardrop, J. G. (1952). Some Theoretical Aspects of Road Traffic Research. *Proceedings of the institution of civil engineers*, *1*(3), 325–362.

Wen, L., & Eglese, R. (2016). Minimizing co2e emissions by setting a road toll. *Transportation Research Part D: Transport and Environment*, *44*, 1–13.

Whittaker, G., Färe, R., Grosskopf, S., Barnhart, B., Bostian, M., Mueller-Warrant, G., & Griffith, S. (2017). Spatial targeting of agri-environmental policy using bilevel evolutionary optimization. *Omega*, *66*, 15–27.

Xin, C., Letu, Q., & Bai, Y. (2013). Robust optimization for the hazardous materials transportation network design problem. In P. Widmayer, Y. Xu, & B. Zhu (Eds.), *Combinatorial optimization and applications* (pp. 373–386). Springer International Publishing.

Yanxia Wu, G. J., & Cheng, Y. (2020). Cloud manufacturing service composition and optimal selection with sustainability considerations: A multi-objective integer bilevel multi-follower programming approach. *International Journal of Production Research*, *58*(19), 6024–6042.

Yi, W., Nozick, L., Davidson, R., Blanton, B., & Colle, B. (2017). Optimization of the issuance of evacuation orders under evolving hurricane conditions. *Transportation Research Part B: Methodological*, *95*, 285–304.

Zeng, B., Dong, H., Sioshansi, R., Xu, F., & Zeng, M. (2020). Bilevel robust optimization of electric vehicle charging stations with distributed energy resources. *IEEE Transactions on Industry Applications*, *56*(5), 5836–5847.

Zhao, L., Xu, X., Gao, H. O., Wang, J., & Xie, Y. (2016). A bi-level model for ghg emission charge based on a continuous distribution of travelers' value of time (vot). *Transportation Research Part D: Transport and Environment*, *47*, 371–382.

Zhao, L., Li, C., Huang, R., Si, S., Xue, J., Huang, W., & Hu, Y. (2013). Harmonizing model with transfer tax on water pollution across regional boundaries in a china's lake basin. *European Journal of Operational Research*, *225*(2), 377–382.

Zhou, Y., Li, L., & Zhang, Y. (2023). Location of transit-oriented development stations based on multimodal network equilibrium: Bi-level programming and paradoxes. *Transportation Research Part A: Policy and Practice*, *174*, 103729.

Zhu, J., Chen, L., Wang, X., & Yu, L. (2020). Bi-level optimal sizing and energy management of hybrid electric propulsion systems. *Applied Energy*, *260*, 114134.

Appendix 8.A Acronyms

Table 8.A.1: Acronyms

Acronym	Meaning	Acronym	Meaning
ADP	Approximate dynamic programming	ILS	Iterated local search
B&B	Branch-and-Bound	KKT	Karush–Kuhn–Tucker reformulation
B&C	Branch-and-Cut	LL	Lower level
BD	Benders decomposition	LP	Linear programming
BSA	Binary search algorithm	MLMF	Multi-leader-multi-follower
CCG	Column-and-constraint generation	MLSF	Multi-leader-single-follower
CG	Column generation	MOEA	Multi-objective evolutionary algorithm
CP	Cutting plane	MOPSO	Multi-objective practical swarm optimization
CSR	Corporate social responsibility	MSWS	Municipal solid waste management
DP	Dynamic programming	PRP	Pollution routing problem
EA	Evolutionary algorithm	PSO	Practical swarm optimization
EIP	Eco-industrial park	RH	Robust heuristic
EPEC	Equilibrium problem with equilibrium constraints	RO	Robust optimization
EV	Electric vehicle	SA	Simulated annealing
FLP	Facility location problem	SCND	Supply Chain network design
FWA	Frank-Wolfe algorithm	SDG	Sustainable development goal
GA	Genetic algorithm	SLMF	Signle-leader-multi-follower
GIA	Greedy insertion algorithm	SLR	Single-level reformulation
GWP	Global warming potential	SLSF	Single-leader-single-follower
HA	Hybrid algorithm	SO	Stochastic optimization
HCP	Heuristic cutting plane	TS	Tabu search
HEUR	(Meta)heuristic	UL	Upper level
HGA	Hybrid genetic algorithm	VRP	Vehicle routing problem
HMOEA	Hybrid multi-objective evolutionary algorithm	WS	Weighted sum
HTNDP	Hazardous materials transportation network design problem		

Chapter 9

Bilevel Optimization for Green Energy Problems[1]

Abstract

In deregulated electricity markets, decision makers engage in hierarchical interactions to balance economic interests with market demand and environmental concerns. Over the last decade, bilevel optimization has emerged as a valuable tool for modeling such complex problems. This paper offers an overview of bilevel optimization applications in green energy, particularly emphasizing environmental cost reduction and the promotion of renewable energy systems. We analyze bilevel players and their decisions and objectives, along with solution methods and outcomes, particularly in integrated energy systems, renewable energy markets, and emissions trading and support schemes. This survey consolidates existing literature and provides direction for future investigations in sustainable energy optimization.

9.1 Introduction and methodology

In deregulated electricity markets, various decision makers, including market operators and generation/transmission companies, interact in a hierarchical manner to pursue their own economic benefit while simultaneously addressing market demand and adhering to multiple constraints (e.g., reliability, environmental restrictions). A vast literature has consequently emerged on energy and electricity bilevel applications. We refer the interested reader to Gabriel et al. (2012) and Wogrin et al. (2020) for discussion on general bilevel applications on energy, electricity, and natural gas markets primarily focused on efficiency and profitability.

Only in recent years, technological and innovation processes have started prioritize the environmental sustainability of energy systems to combat the climate crisis. Different sources of energy, especially renewables, are integrated and combined in new systems to increase availability, produce cleaner energy, and reduce waste and emissions. Following this trend, in energy and electricity markets, a number of applications of bilevel optimization problems with environmental sustainability perspectives have emerged over

[1] Caselli, G., Iori, M., Ljubić, I. (2024). Bilevel optimization for green energy problems (technical report).

the last decade. To the best of our knowledge, a survey focused on these applications is absent in the operations research and mathematical programming literature. This work aims to fill this gap by providing a general overview of bilevel optimization applications in the green energy sector, particularly those focused on minimizing environmental costs of energy systems or promoting the development of renewable energy systems with positive environmental impacts.

We direct readers to Caselli et al. (2024) for a comprehensive survey on bilevel applications in other sectors, in which we cover a variety of applications of bilevel optimization, namely transportation and logistics, production planning and manufacturing, waste, material, and environment management, supply chains, and disaster prevention and response. We adopt a triple bottom line perspective that emphasizes economic, environmental, and social dimensions of sustainability (see Elkington, 2002). Due to the significance and breadth of the topic, we treat the energy sector separately as a second survey. We employ a similar methodology for analyzing the literature in this survey. For each of selected work, we identify players, decisions, and goals of the bilevel games, along with mathematical formulations, solution methods, case studies, and results presented by various authors.

Furthermore, readers are encouraged to refer to the introductory sections of Caselli et al. (2024) for a general overview of common bilevel game types, players, goals, and solution approaches. In this work, we only report the formulations of two general classes of bilevel problems which cover the majority of applications that we survey, namely the single-leader-single-follower (SLSF) and single-leader-multi-follower (SLMF) problems.

A general SLSF bilevel optimization problem under optimistic assumption models a sequential game with two players, called the leader and the follower. Variables $x \in \mathbb{R}^{n_x}$ represent the decisions of the leader, while variables $y \in \mathbb{R}^{n_y}$ represent the decisions of the follower as follows:

$$\min_{x \in X, y} \quad F(x, y) \tag{9.1}$$

$$\text{s.t.} \quad G(x, y) \geq 0 \tag{9.2}$$

$$y \in S(x), \tag{9.3}$$

where $S(x)$ is the set of optimal solutions of the x-parameterized problem

$$\min_{y \in Y} \quad f(x, y) \tag{9.4}$$

$$\text{s.t.} \quad g(x, y) \geq 0. \tag{9.5}$$

Models (9.1)-(9.3) and (9.4)-(9.5) define, respectively, the so-called upper-level (UL) and lower-level (LL) problems. In the hierarchical setting of bilevel problems, the leader, holding a higher position, initiates decisions by anticipating the outcomes of the follower's choices. The follower, occupying a lower hierarchical level, responds to the leader's decision. In the scenarios explored within this paper, the optimistic variant is commonly applied, implying that the leader controls the follower's variables to yield the most favorable solution for the leader among the various LL optimal solutions. The optimistic variant of bilevel problems garners the most attention in research since it allows for treatable single-level reformulations.

In SLMF games, there is a single leader but multiple followers. Either the followers are independent or they play a Nash game among them (Lampariello et al., 2020), given the decision of the leader. The vector of variables y_j represents the decision of j-th follower with $j = 1, \ldots, n_f$ (n_f is the number of followers). The vector of variables x represents the leader's decision. A general SLMF game in the optimistic version is formulated as follows:

$$\min_{x \in X, y_1, \ldots, y_{n_f}} \quad F(x, y) \tag{9.6}$$

$$\text{s.t.} \quad G(x, y) \geq 0 \tag{9.7}$$

$$y := (y_j)_{j=1,\ldots,n_f} \text{ solves GNEP}(x). \tag{9.8}$$

Vector y represents all decision variables of all followers and GNEP(x) is the set of generalized Nash equilibra of the non-cooperative game among the n_f followers, where the objective function and the feasible set of every follower explicitly depends on the decisions on the other followers. Vector y_{-j} represents the decision variables of all followers except for those of j-th follower. Given that $Y_j(x, y_{-j})$ represents the feasible set of j-th follower, defined by private constraints of j-th follower and shared constraints of all followers, the LL problem of j-th follower can be generally formulated as:

$$\min_{y_j} \quad f_j(y_j, x, y_{-j}) \tag{9.9}$$

$$\text{s.t.} \quad y_j \in Y_j(x, y_{-j}). \tag{9.10}$$

GNEPs arise quite naturally from standard Nash equilibrium problems in energy networks and electricity grids.

The survey aims to identify the players involved in green energy bilevel games, and the players' decisions and goals, to detect whether leader and follower problems are single- or multi-objective, economic and/or environmental, and which objectives are optimized, and the proposed solution methods. Specifically, we classify sustainable bilevel applications in the green energy sector, covering works where green goals are included in players' objective functions or the implementation of bilevel applications comprises positive environmental impacts, such as emission reduction or renewable energy expansion. We encompass bilevel problems related to (i) integrated energy systems, where multiple energy sources are regulated and coordinated by authorities and service providers, (ii) investors, producers, and users of renewable energy markets, and (iii) emissions trading and support schemes used by governments with companies and users to mitigate climate change. We do not cover bilevel applications in the energy and electricity markets without sustainable applications, for which we refer the reader to Wogrin et al. (2020).

The reminder of this paper is structured as follows. In Sections 9.2-9.4, we present the following areas of green energy applications: integrated energy systems (Section 9.2); renewable energy markets (Section 9.3); emissions trading and support schemes (Section 9.4). Section 9.5 identifies research gaps in the existing literature and offers guidance for future research.

9.2 Integrated energy systems

Integrated energy systems (IES) refer to the comprehensive planning and operation of the energy system as a whole by combining multiple energy sources, such as fossil

Table 9.2.1: Summary of papers on bilevel applications focused on integrated energy systems.

Application	References	Bilevel game		Players		Objectives		Method
		Class	Eq.	UL	LL	UL	LL	
Integrated energy systems	Song et al. (2021)	SLSF		Electricity decision maker	Natural gas decision maker	**MO:** Min operating costs; Min emissions	Min operating costs	NSGA-II + GA
	Li et al. (2017)	SLSF		Electricity decision maker	Natural gas decision maker	Min operating costs	Min operating costs	SLR
	Feijoo and Das (2015)	SLSF		MGs decision maker	SG decision maker	Min operating costs	Min social cost	SLR
	Liu and Li (2015)	SLSF		SG decision maker	Users	Min electric power generation and thermal emissions costs	**MO:** Min compensation cost incentive cost, and emissions equivalent cost	NSGA-II
	Zhong et al. (2023)	SLMF	✓	Energy hub manager	Drivers and users	Min costs + Max hub benefit	Max profit	SLR + CCG
	Fan et al. (2023)	SLSF		Storage and power decision maker	Equipment power decision maker	Min storage costs	Min operating costs	SLR

Table 9.2.2: Summary of papers on bilevel applications focused on renewable energy markets.

Application	References	Bilevel game		Players		Objectives		Method
		Class	Eq.	UL	LL	UL	LL	
Renewable energy markets	Baringo and Conejo (2014)	SLMF	✓	Wind investor	Wind energy producers	Min investment costs minus revenues	Max social welfare	SLR
	Maurovich-Horvat et al. (2015)	SLMF	✓	Wind investor	Wind energy producers	Max profit	Max profit	SLR
	Wang et al. (2018)	SLMF	✓	System planner	Wind energy producers	Max weighted wind penetration minus investment costs	Min dispatch cost	SLR + CCG
	Bard et al. (2000)	SLSF		Government	Agricultural sector	Min costs	Max profit	HEUR
	N. Zhao and You (2019)	SLMF		Government	Diary firms	**MO:** Min intervention; Min unit generation costs	Max net present value	GO
	Shi et al. (2020)	SLMF		Government	Disposa plants	**MO:** Min costs; Min emissions	Max net profit	HEUR

fuels, renewable energy, and energy storage, establishing efficient and coordinated distribution and utilization methods. The objective is decarbonization, energy efficiency, affordability and reliability of the energy system.

Li et al. (2017) and Song et al. (2021) propose bilevel models for the energy dispatch of integrated electricity and natural gas systems with wind energy penetration. In both models, the leader in the UL problem is the electricity system decision maker, who determines power outputs from different sources. Meanwhile, the follower in the LL problem is the natural gas system decision maker, who decides gas supply and charge/discharge. In Li et al. (2017), both players minimize solely operating costs. The authors obtain a single-level MILP reformulation by KKT conditions and solve two case case studies on Belgian gas systems. In Song et al. (2021), the UL problem is bi-objective since the leader minimizes operating costs and carbon emissions. The follower minimizes operating costs. The authors present an interactive solution algorithm based on NSGA-II and GA. Both works demonstrate the effectiveness of integrated systems in enhancing wind power accommodation and reducing gas consumption and emissions.

Bilevel dispatch models are also applied to smartgrid (SG) and microgrid (MG) systems. SGs optimize large-scale electricity grids by technological monitoring and control systems to reduce losses and better integrate renewable energy sources. MGs offer independent power to small-scale systems via distributed energy resources like generators and solar panels. MGs can be considered as generators (when selling electricity to the smart grid) or loads (when buying electricity from the SG). Feijoo and Das (2015) model the hourly interaction between a set of MGs and a central SG as an SLSF bilevel model. In the UL, at the beginning of any hour, the MGs wind/solar operational plan is defined for all remaining hours of the horizon to minimize the total MGs operating costs, including production, maintenance, storage, and trading power with the SG. MGs are not modeled as independent players in an equilibrium, but as entities governed by a single decision maker. In the LL, the SG decision maker defines the dispatch to MGs for the current hour to minimize the total social cost, including the electricity generation cost and the social cost of carbon minus the MGs benefit. The authors apply the KKT reformulation to a sample SG with 37 MGs, showing interesting market equilibrium insights. By including the social cost of carbon in the model, a higher MG renewable generation reduces the average electricity price and increases the demand in the SG, even in the peak hours.

Liu and Li (2015) model the interaction between an SG dispatching thermal units and the final users for energy savings and emissions reduction in China. In the UL, the leader is the SG decision maker, who dispatch thermal units to minimize electric power generation and carbon emissions costs. The LL problem defines users' power demand with three minimization objective functions, namely the compensation cost for interruptible loads, the incentive cost for incentive loads, and the equivalent cost for load emissions. The authors propose an iterative algorithm based on NSGA-II, where the stopping criteria is the maximum emission tolerance. Results on a case study in China show that the proposed approach can effectively reduce emissions both from the generation side and the demand side.

Zhong et al. (2023) introduce an SLMF bilevel model for the design of an energy hub that is, a low-carbon and cost-effective IES in which renewable sources of energy (e.g., solar, wind) combined with electricity sources and storage systems serve the household, industrial, and electric vehicles demand of the energy grid. The leader is the energy hub manger setting the selling and incentive prices of the CO_2 trading system to minimize total costs (including emissions cost) and maximize the benefit of the hub, while the followers are the electric vehicles drivers and system users who decide the multi-energy loads to maximize their profit. The KKT conditions allow to obtain the single-level model, to which a robust optimization (RO) approach is applied to introduce renewable energy generation uncertainty. The model is solved by a column-and-constraint generation (CCG) method on a case study in China showing an improvement in the energy hub efficiency.

Fan et al. (2023) study the optimization of an IES with multi-energy storage and uncertainty due to wind power generation. They present a bilevel model to consider both IES early planning and late operations. The UL and LL problems of the proposed SLSF bilevel model deal with, respectively, the storage and power planning strategy to minimize the storage allocation cost and the equipment power operational decisions to minimize the operating costs. The authors apply an RO approach to the SLR model and solve an illustrative example proving the stability of the IES against wind power uncertainty.

9.3 Renewable energy markets

The energy transformation from fossil fuels to renewable energy is inevitable to achieve carbon emission reduction goals. Bilevel optimization has been applied to wind power systems planning as well as bioenergy and bioelectricity, new sources of energy obtained from more sophisticated processes. Baringo and Conejo (2014) formulate a bilevel model for strategic wind power investment planning. In the UL, the leader is the wind investor, who determines the size and site of new wind units and the production to be sold in the electricity market to maximize the expected profit and minimize the investment cost. In the LL, there is one clearing market problem for each operating condition, that is a wind power production scenario. A finite set of conditions is preliminarily generated by a K-means algorithm. The LL problem determines the production prices to maximize social welfare. The author present the KKT reformulation, which is then linearized to obtain an MILP model. Solving the MILP on benchmark instances show the profitability of the model for the wind investor.

In the same wind investment context, Maurovich-Horvat et al. (2015) formulate an SLMF stochastic bilevel model. The leader is the investor making investment decisions for profit maximization. At the LL, producers act as followers in a Cournot oligopoly (i.e., each producer competes on the amount of energy to produce) or in perfect competition to maximize their profit after wind availability uncertainty realizes. The authors propose a KKT reformulation, obtaining a linear two-stage stochastic optimization (SO) model that they then test on small instances. They show that the expected generation from renewable sources increases significantly with a Cournot oligopoly.

Wang et al. (2018) study another stochastic bilevel model for wind power investment planning problem including topology control (i.e., to switch off some existing electricity lines to accommodate wind power transmission lines). The UL problem models the wind farms siting and sizing decisions of the system planner, who aims to maximize the weighted wind penetration minus investment costs. The LL problem is a market clearing problem to minimize the dispatch cost under topology control for each scenario of electricity demand and wind intensity. The authors propose a CCG algorithm for the SLR of the problem, and use it to to efficiently solve test instances to optimality, showing that adopting topology control on a small number of lines can help reduce wind energy curtailment and improve wind penetration level.

Bard et al. (2000) present an SLSF bilevel model for biofuel production from farm crops. The leader is the government, setting tax credits for each final product or biofuel that industry can produce to minimize costs subject to a given level of land dedicated to nonfood crops. The petro-chemical industry producing biofuel is neutral in this game. The follower is represented by the agricultural sector, which selects the best mix of crops to grow as well as the percentage of land to set aside to maximize profits. The authors propose a grid-search based heuristic algorithm to solve a case study in France.

N. Zhao and You (2019) propose a multi-objective SLMF bilevel model for bioelectricity generation from diary industry waste, that is generally defined as "waste-to-energy" process. The leader is the government, who determines incentive policies (disposal fee, subsidy, and refund) under a bioelectricity generation target to minimize total government intervention and its unit cost on generating a target amount of bioelectricity. The followers are the diary farms, who decide independently on conversion technologies adoption to maximize their net present values. The authors propose a tailored global optimization (GO) algorithm based on reformulation, linearization, and decomposition. They solve a large-size case study in New York State (US), showing that incentive policies can effectively promote bioelectricity generation. They also show that a combination of the three selected policies may be more efficient in minimizing total government intervention, and that there exist a trade-off between total government intervention and unit cost of the government on bioelectricity generation.

Shi et al. (2020) study a similar multi-objective SLMF bilevel model for the waste-to-energy process in the context of kitchen waste for the production of biodiesel, biogas, power and heat from different processes. The leader, in the UL, is the local government selecting kitchen waste disposal plants and giving them subsidies for bioenergy generation to minimize total costs and total carbon emissions. In the LL, the disposal plants act as followers in competition to maximize their net profit, given by production profits plus subsidies minus costs. The authors propose a tailored heuristic algorithm that is applied to a case study in China to obtain valuable solutions and show, under a scenario analysis, that soft subsidies combined with advanced technology can effectively impact waste-to-energy processes.

9.4 Emissions trading and support schemes

Emission trading systems (ETSs) and support schemes (e.g., carbon taxes, tariffs, and subsidies) create financial incentives and regulatory mechanisms for regulated entities to reduce their emissions by either investing in cleaner technologies or purchasing addi-

Table 9.4.1: Summary of papers on bilevel applications focused on emissions trading and support schemes.

Application	References	Bilevel game		Players		Objectives		Method
		Class	Eq.	UL	LL	UL	LL	
Emissions trading and support schemes	Kainuma et al. (1999)	SLMF		Policy maker	C Energy consumers	Min CO_2 emissions	Min costs	HEUR
	Huang and Xu (2020)	SLMF	✓	Authority	Coal-fired power plants	**MO:** Max total profit; Max sludge utilization; Min emissions	Max profit	WS + SLR
	Huang and Xu (2023)	SLMF		Authority	Coal-fired power plants	Max tax revenues	Min fuel, carbon, investement costs	HEUR
	J. Zhao et al. (2024)	SLMF	✓	Authority	Coal-fired power plants	Max tax revenues	Max profit	SLR
	Yan et al. (2024)	SLMF	✓	Authority	Coal-fired power plants	**MO:** Max social benefit, carbon subsidies equality, emissions reduction	Max profit	ϵ-constraint + SLR
	Martelli et al. (2020)	SLSF		Government	Multi-energy system decision maker	Min costs	Min investment, operating costs	Hybrid
	Taha et al. (2014)	SLMF	✓	Policy maker	Power generation companies	Max social welfare	Max profit	SLR
	Pineda et al. (2018)	SLMF	✓	Policy maker	Power generation companies	Max social welfare	Max profit	SLR
	Hua et al. (2022)	SLMF		Policy maker	Power generation companies/Consumers	**MO:** Carbon revenue neutrality; Carbon emissions reduction	Max profit/Min energy bills	HEUR
	Fanzeres et al. (2015)	SLSF		Trading company	Energy producers	Max profit	Max revenue	SLR

tional permits if they exceed their allocated limit. ETSs are used by governments and other authorities to mitigate climate change and reduce air pollution while promoting economic efficiency and energy security. Since they multiple players in hierarchical relationships, several authors address related optimization problems by bilevel programming. Kainuma et al. (1999) study an SLMF bi-level problem for sustainable end-use energy. The leader is the policy maker setting carbon taxes on different energy types (e.g., oil, coal, gas, solar) and different subsidies on carbon-emitting technologies (e.g., automobiles, air conditioning, boilers) to minimize CO_2 emissions. The followers are the consumers of energy services (e.g., transportation, heating, lighting) who aim at minimizing their costs. The authors propose a heuristic algorithm and use it to solve a case study in Japan. Results show that high carbon taxes or, alternatively, low carbon taxes combined with subsidies could be effective options for reducing CO_2 emissions in Japan.

Huang and Xu (2020) and Huang and Xu (2023) propose SLMF bilevel games between the authority (leader) and coal-fired power plants (followers) for clean energy production from coal. In Huang and Xu (2020), the authority sets carbon emissions quota allocation towards co-combustion of coal and sewage sludge to maximize economic benefits (total power plants profits) and sewage sludge utilization and minimize carbon emissions. The plants acts as followers in the competitive carbon trading market, making co-combustion decisions to maximize their profits. Fuzzy numbers are used in the model to represent uncertain parameters of power plants operations (e.g., carbon emissions, power generation, procurement costs). The authors apply the weighted sum (WS) method to get a single-objective UL model and then the KKT conditions are used to reformulate the problem as an MILP. In J. Zhao et al. (2024), the same problem is modelled with a single-objective UL problem (tax revenues maximization) and an RO approach is developed to deal with uncertainty. In Yan et al. (2024), the leader's UL problem includes the maximization of three objectives, namely social benefit, carbon subsidies equality, and carbon emissions reduction. The ϵ-constraint and KKT conditions allow to obtain a tractable single-level MILP model. In Huang and Xu (2023), the authority designs a carbon tax revenue recycling scheme as an incentive for biomass/coal co-firing with the goal of maximizing tax revenues. Power plants decide which co-firing technology to adopt and the amount of fuel to use for the coal/biomass co-firing process to minimize their costs including fuel, carbon-related, and investment costs. In this case, since the coal-fired power plants are considered to work together to influence the leader's decision on the carbon tax, an interactive heuristic method based on satisfactory functions is proposed. All these works show positive effects on carbon emissions reduction and biomass energy utilization by real-world applications in China.

Martelli et al. (2020) study the combination of incentives and carbon taxes for the introduction of renewable energy in small/medium multi-energy systems. At the UL of the bilevel problem, the government decides the incentive rates for electricity and heat from renewables and CO_2 emissions taxes to meet the desired emission reduction target while minimizing its costs. At the LL, the multi-energy system decision maker decides the energy mix and hourly operations to minimize the total annual costs including investment and operating costs. The authors propose a hybrid heuristic based on particle swarm optimization that is effectively applied to four real-world applications, namely a university campus, a hospital, an urban district, and an office building, in Italy. Results show that in all cases the optimized policies (incentives and taxes) can significantly reduce CO_2 emission and government costs.

Taha et al. (2014) and Pineda et al. (2018) study other policy makers' support schemes for renewable energy expansion. In Taha et al. (2014), the policy maker is the leader of an SLMF game, providing incentives for green energy producers in the form of variable subsidy price-based contracts called "Quasi-Feed-in-Tariff" (Quasi-FIT) to maximize the overall social welfare, measured as the consumption surplus minus subsidies and energy and environmental costs. The generation companies of the MG are the followers, who play in a non-cooperative Nash-Cornout game by deciding on green energy generation quantities to maximize their profit. The bilevel problem is reformulated as a single-level problem and solved by built-in solvers (i.e., filter, KNITRO, PATH, CONOPT) combining different algorithms on small illustrative examples showing the appropriateness of the leader's objective function under different energy lines and minimum green energy conditions.

Pineda et al. (2018) study similar problems with not only FIT, but also the feed-in premium (FIP) and tradable green certificates. FIT and FIP give rise to stochastic bilevel problems, with uncertain prices and volume risks for renewables. The policy maker decides on the incentives in the UL, and the power generation companies compete in the market under equilibrium and different risk-averse strategies. The authors solve a case study in the Danish power system under different scenarios, showing that risk-aversion of the generation companies impacts the determination of the best support scheme.

Hua et al. (2022) propose another SLMF game where the policy maker acts as the leader, deciding carbon prices and monetary compensation rates on the market, and power generation companies and consumers acts as followers with, respectively, generation and consumption decisions. The goals of the policy maker are carbon revenue neutrality and carbon emissions reduction. Generation companies maximize profits while consumers minimize energy bills. The authors adopt a tailored heuristic algorithm to solve a blockchain platform application in the British power system, showing potential positive results for all the three actors.

Contracting and trading strategies in energy markets now embed more and more renewable energy sources, which are characterized by high uncertainty. Fanzeres et al. (2015) study a hybrid bilevel model for renewable portfolio optimization of a risk-averse energy trading company. SO and RO approaches are introduced to deal with, respectively, renewable production and future prices uncertainty. In the UL, the energy trading company defines the renewable portfolio to maximize profit. In the LL, different energy production stress scenarios realize and the worst-case revenue stream is assessed. The authors propose an SLR based on the duality theory. Results on two case studies in Brazil show a significant impact of prices and renewable production uncertainty on contractor's profit.

9.5 Conclusions and future research directions

The paper analyzes bilevel optimization applications in green energy, addressing the growing need to balance economic interests with environmental sustainability in deregulated electricity markets. Through hierarchical interactions, decision makers strive to meet market demands while adhering to various constraints, including reliability and environmental concerns. Bilevel optimization enables the modeling of complex energy systems with a focus on minimizing environmental costs and promoting renewable energy solutions as well as seeking efficiency and profitability.

By analyzing bilevel players, their decisions, objectives, and solution methods, this work consolidates existing literature and highlights the importance of sustainable energy optimization. We have categorized bilevel applications in the green energy sector, covering a wide range of scenarios where environmental considerations are integral to decision-making processes. In integrated energy systems applications, the leader and the followers typically represent two decision makers of the grid, often managing different energy sources. Case studies with smart grids and microgrids show the efficacy of bilevel optimization in enhancing energy efficiency, stability, and reducing gas consumption and emissions both from the generation side and the demand side. In renewable energy markets, bilevel games typically emerge between investor and producers of renewables. In wind energy markets, the investor, as leader, optimizes the expansion of the wind farm for profit maximization. The followers, the wind energy producers, compete on the amount of energy to produce to maximize social welfare or individual profit. These applications demonstrate notable increases in wind power penetration and the anticipated generation of energy from renewables. Although less prevalent, applications involving other renewable energy sources such as biofuel energy and bioelectricity derived from waste-to-process sources also underscore the potential of governmental incentives and subsidies in promoting sustainable power generation from agricultural and industrial processes. Numerous case studies exploring emission trading systems and support schemes showcase the potential of bilevel optimization in reducing CO_2 emissions. Solution methods typically encompass single-level reformulations to be solved exactly or tailored heuristic methods.

Future research should continue to explore innovative bilevel optimization approaches to address evolving challenges in green energy. The study area is growing fast with new related topics. Technological progress and increasing pressure on governments and industrial entities to take measures against environment degradation introduce alternative sources of renewable energy, including hydropower and electricity-hydrogen hybrid power systems (see, e.g., Cai et al., 2023 and Guo et al., 2023). Governmental incentives and support schemes may be extended to cover the three economic, environmental, and social dimensions of sustainability together to pursue economic profitability, emission reduction, and social welfare together. New needs are emerging in societal energy market involving, for instance, electric vehicles charging networks (see, e.g., Zeng et al., 2020) and cloud data electricity-carbon market (see, e.g., Zeng et al., 2024).

In conclusion, this survey serves as a foundational resource for researchers and practitioners interested in understanding and advancing bilevel optimization applications in the green energy sector, contributing to the ongoing transition towards a more sustainable energy future.

References

Bard, J. F., Plummer, J., & Claude Sourie, J. (2000). A bilevel programming approach to determining tax credits for biofuel production. *European Journal of Operational Research*, *120*(1), 30–46.

Baringo, L., & Conejo, A. J. (2014). Strategic wind power investment. *IEEE Transactions on Power Systems*, *29*(3), 1250–1260.

Cai, P., Mi, Y., Ma, S., Li, H., Li, D., & Wang, P. (2023). Hierarchical game for integrated energy system and electricity-hydrogen hybrid charging station under distributionally robust optimization. *Energy*, *283*, 128471.

Caselli, G., Iori, M., & Ljubić, I. (2024). Bilevel optimization with sustainability perspective: A survey on applications (technical report).

Elkington, J. (2002). *Cannilbals with forks: The triple bottom line of the 21st century*. Oxford Press.

Fan, W., Tan, Q., Zhang, A., Ju, L., Wang, Y., Yin, Z., & Li, X. (2023). A bilevel optimization model of integrated energy system considering wind power uncertainty. *Renewable Energy*, *202*, 973–991.

Fanzeres, B., Street, A., & Barroso, L. A. (2015). Contracting strategies for renewable generators: A hybrid stochastic and robust optimization approach. *IEEE Transactions on Power Systems*, *30*(4), 1825–1837.

Feijoo, F., & Das, T. K. (2015). Emissions control via carbon policies and microgrid generation: A bilevel model and pareto analysis. *Energy*, *90*, 1545–1555.

Gabriel, S. A., Conejo, A. J., Fuller, J. D., Hobbs, B. F., & Ruiz, C. (2012). *Complementarity modeling in energy markets*. Springer Science & Business Media.

Guo, H., Gong, D., Zhang, L., Wang, F., & Du, D. (2023). Hierarchical game for low-carbon energy and transportation systems under dynamic hydrogen pricing. *IEEE Transactions on Industrial Informatics*, *19*, 2008–2018.

Hua, W., Jiang, J., Sun, H., Teng, F., & Strbac, G. (2022). Consumer-centric decarbonization framework using stackelberg game and blockchain. *Applied Energy*, *309*, 118384.

Huang, Q., & Xu, J. (2020). Bi-level multi-objective programming approach for carbon emission quota allocation towards co-combustion of coal and sewage sludge. *Energy*, *211*, 118729.

Huang, Q., & Xu, J. (2023). Carbon tax revenue recycling for biomass/coal co-firing using stackelberg game: A case study of jiangsu province, china. *Energy*, *272*, 127037.

Kainuma, M., Matsuoka, Y., Morita, T., & Hibino, G. (1999). Development of an end-use model for analyzing policy options to reduce greenhouse gas emissions. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *29*(3), 317–324.

Lampariello, L., Sagratella, S., Shikhman, V., & Stein, O. (2020). Interactions Between Bilevel Optimization and Nash Games. In S. Dempe & A. Zemkoho (Eds.), *Bilevel Optimization: Advances and next challenges* (pp. 3–26). Springer International Publishing.

Li, G., Zhang, R., Jiang, T., Chen, H., Bai, L., & Li, X. (2017). Security-constrained bilevel economic dispatch model for integrated natural gas and electricity systems considering wind power and power-to-gas process. *Applied Energy*, *194*, 696–704.

Liu, J., & Li, J. (2015). A bi-level energy-saving dispatch in smart grid considering interaction between generation and load. *IEEE Transactions on Smart Grid*, *6*(3), 1443–1452.

Martelli, E., Freschini, M., & Zatti, M. (2020). Optimization of renewable energy subsidy and carbon tax for multi energy systems using bilevel programming. *Applied Energy*, *267*, 115089.

Maurovich-Horvat, L., Boomsma, T. K., & Siddiqui, A. S. (2015). Transmission and wind investment in a deregulated electricity industry. *IEEE Transactions on Power Systems*, *30*(3), 1633–1643.

Pineda, S., Boomsma, T. K., & Wogrin, S. (2018). Renewable generation expansion under different support schemes: A stochastic equilibrium approach. *European Journal of Operational Research*, *266*(3), 1086–1099.

Shi, Y., Deng, Y., Wang, G., & Xu, J. (2020). Stackelberg equilibrium-based eco-economic approach for sustainable development of kitchen waste disposal with subsidy policy: A case study from china. *Energy*, *196*, 117071.

Song, X., Wang, Y., Zhang, Z., Shen, C., & Peña-Mora, F. (2021). Economic-environmental equilibrium-based bi-level dispatch strategy towards integrated electricity and natural gas systems. *Applied Energy*, *281*, 116142.

Taha, A. F., Hachem, N. A., & Panchal, J. H. (2014). A quasi-feed-in-tariff policy formulation in micro-grids: A bi-level multi-period approach. *Energy Policy*, *71*, 63–75.

Wang, Y., Liu, S., Wang, J., & Zeng, B. (2018). Capacity expansion of wind power in a market environment with topology control. *IEEE Transactions on Sustainable Energy*, *10*(4), 1834–1843.

Wogrin, S., Pineda, S., & Tejada-Arango, D. A. (2020). Applications of bilevel optimization in energy and electricity markets. In S. Dempe & A. Zemkoho (Eds.), *Bilevel optimization: Advances and next challenges* (pp. 139–168). Springer International Publishing.

Yan, S., Lv, C., & Yao, L. (2024). Social welfare and equality equilibrium based carbon tax subsidy incentive approach for biomass-coal co-firing towards carbon emissions. *Energy*, *291*, 130282.

Zeng, B., Dong, H., Xu, F., & Zeng, M. (2020). Bilevel programming approach for optimal planning design of ev charging station. *IEEE Transactions on Industry Applications*, *56*, 2314–2323.

Zeng, B., Zhou, Y., Xu, X., & Cai, D. (2024). Bi-level planning approach for incorporating the demand-side flexibility of cloud data centers under electricity-carbon markets. *Applied Energy*, *357*, 122406.

Zhao, J., Liu, Y., & Chen, A. (2024). Modeling robust bi-level bcc production planning problem with uncertain carbon emission mechanism. *Computers & Chemical Engineering*, *181*, 108548.

Zhao, N., & You, F. (2019). Dairy waste-to-energy incentive policy design using stackelberg-game-based modeling and optimization. *Applied Energy*, *254*, 113701.

Zhong, J., Li, Y., Wu, Y., Cao, Y., Li, Z., Peng, Y., Qiao, X., Xu, Y., Yu, Q., Yang, X., Li, Z., & Shahidehpour, M. (2023). Optimal operation of energy hub: An integrated model combined distributionally robust optimization method with stackelberg game. *IEEE Transactions on Sustainable Energy*, *14*(3), 1835–1848.

Conclusions

This thesis collects a series of self-contained works in the OR field, exploring mathematical programming and forecasting methodologies for real-world applications of a variety of decision-making problems, from single-objective to bi-objective and finally to bilevel optimization. This collection has revealed not only the richness and complexity of real-world OR applications but also the close interrelation between the field and sustainable development. The contributions of each chapter have significantly advanced our understanding of the decision-making problems and provided practical solutions to diverse economic, environmental, and social challenges. The key conclusions that can be drawn from this thesis are the following.

Advancements in single-objective optimization The first part of the thesis is devoted to single-objective optimization problems and exact approaches.

We have derived ILP formulations for real-world single-objective assignment problems in the education and healthcare sectors. Binary variables are employed for assignment decisions and linear inequalities reproduce the eligibility and problem-specific requirements. Objectives are in some sense social-related. In Chapter 2, we aim to maximize the satisfaction of preferences expressed by university tutors for courses over an entire academic period. We have designed an ILP with binary variables for tutors/workshops assignment with several complicating constraints for avoiding overlaps, respecting tutors' maximum working hours, and allowing tutors' movements between different campuses. `Gurobi` solver was able to solve large random instances of the problem, which is faced by many universities every year and is crucial for the quality of teaching since tutors tend to give high-quality workshops when they are assigned to preferred courses. `Gurobi` also solved a real-world instance of the University of Edinburgh (with almost 300 tutors and around 100 courses) to optimality, increasing satisfied preferences by 30% with respect to the hand-based assignment. The optimal solution was provided to the organization as a starting-point solution for the timetabling management of the year. We have also tested different objective functions to further investigate fair assignment, considering ordered preference lists or the minimization of the number of courses assigned to each tutor, both appearing valid alternative tutors' utility functions. In Chapter 3, we aim to minimize the overcrowding in the common waiting rooms of a hospital by redesigning the clinic and waiting rooms layout for outpatient appointments. Overcrowding, calculated as the surplus of people on the room capacity, is a measure of infection risk for hospital patients. Indeed, the problem has emerged in COVID-19 emergency times, during which hospitals, and all social spaces, were required with disruptive and immediate changes to limit virus infections as much as possible. OR came into play with several optimization solutions, including ours. Again, `Gurobi` was able to solve our ILP formulation implemented in

Python with two sets of two-index binary variables (tested to be more efficient than ILP with one set of three-index binary variables) on the realistic instance of an Italian hospital to optimality. We have shown that weekly optimal assignment could improve by up to 90% the hospital manual layout. However, a weekly assignment has appeared to require several weekly room changes, that means great managerial efforts by the hospital staff. Such insights have been a crucial source of information for the hospital decision makers. Longer time horizons make the solver struggling with computing times to reach optimality in reasonable time and reduce the improvement on the objective function, showing the difficulty to find good quality and stable solutions of the problem and the importance of trading off social distancing and managerial organization. Future studies could address different approaches to solve larger instances (e.g., with larger time horizons), including exact decomposition techniques to decompose the assignment decisions of appointments to clinic rooms and clinic rooms to waiting areas in two stages, and matheuristic approaches (i.e., heuristics making use of mathematical models or exact algorithms) that solve the optimal assignment for each sector or floor of the hospital and then combine small optimal solutions heuristically.

We have also developed exact approaches in the field of machine scheduling. In Chapter 4, we have studied a new parallel machine scheduling problem with additional consideration of human workforce, that is to assign jobs to machines and workers with eligibility and problem-specific precedence constraints to minimize the total weighted tardiness of jobs. We have introduced in the machine scheduling literature a new concept of precedence between jobs (i.e., contiguity), and we have designed method-specific MILP and CP formulations and a combinatorial Benders' decomposition to solve complex instances of the considered problem. We have built our decomposition to schedule the jobs in the master problem, and then assign jobs to machines and workers in the subproblem. This allows to break precedence and eligibility constraints in the primary and secondary problems, respectively. Benders' cuts are iteratively generated on minimal infeasible subsets of master decision variables, which are found heuristically by the procedure. We have deeply tested our algorithms on experimental instances, finding problem-specific characteristics which make each method outperform the others. In addition, we have succesfully solved a real-world instance of the problem provided by an Italian engineering test laboratory. Promising future research directions could include the development of a hyperalgorithm that combines the strengths of each method, and the study of the dynamic extension of the problem where a rolling horizon approach could help to take into account disruptive events such as late jobs release or machine breakdowns.

Forecasting in decision-making The exploration of forecasting methods has underscored the importance of accurate predictions in enhancing decision-making processes, especially in complex supply chains, such as the perishable food industry, which is characterized by products with highly variable demand trends and short storage. In Chapter 5, we have studied the forecasting problem of an Italian large company operating in the processed meat industry, in which the daily demand of several products must be predicted as accurately as possible. We have tested four machine learning methods and different training procedures on their models. The tests have shown that a combination of the methods outperforms all classical approaches of the forecasting literature and reduces the forecasting error by up to 18% on real data. Finally, we have performed a simulation test on the inventory data of the company, showing that machine learning based forecasting can significantly reduce the number of expired packages to be discarded. This number is an crucial indicator of inventory efficiency in the food supply chain, whose businesses are seriously concerned with not only the economic but also the environmental and social performance.

Balancing economic efficiency and environmental sustainability The second part of the thesis delved into the realm of bi-objective optimization, where economic efficiency and environmental sustainability are considered simultaneously. The developed models and multi-objective optimization methods have presented practical solutions for network optimization problems, offering decision makers a clear understanding of trade-offs between cost and environmental impact. In Chapter 6, we have solved a real-world bi-objective facility location problem arising in the Italian waste management industry. The goal is to minimize costs of opening facilities and collecting waste and the transport CO_2 emissions by deciding whether and where to open additional intermediate transfer facilities among a set of candidate locations and the quantities of waste on the links of the network from source points to final treatment facilities. We have provided a single-period MILP model with flow conservation and capacity constraints, and its multi-period extension that allows to keep facility decisions fixed and let flow decisions vary across the periods. Our second MILP model is more realistic and original in the waste management literature, since it considers seasonal waste fluctuations and combines multi-period and multi-objective perspectives. We have designed an approximated ϵ-constraint method which gives Pareto fronts of solutions for costs/emissions trade-off. In addition, we have also proposed an exact and a heuristic weighted-sum approach, testing different weights to see the variability of solutions with respect to different values of importance of costs and emissions minimization. These methods have been applied to two case studies of an Italian multi-utility in different Italian districts, showing around 20% reductions in costs and emissions by MILP solutions with respect to the non-optimized status quo without intermediate facilities and environmental considerations, and also good stability of solutions across the costs/emissions Pareto fronts. Such solutions give important insights to the company decision maker, who can observe the impact of extreme as well as conservative decisions on the economic and environmental performance of the process. Further computational experiments have shown the scalability of our ϵ-constraint method in building well-shaped Pareto fronts of optimal solutions for random instances with up to 400 sources, 200 locations, and five different waste types. We have also specifically described how costs and emissions parameters have been computed based on the real case, contributing to the theory on the inclusion of emissions consideration in OR models and easing the applicability of the models to similar real-world settings. Future studies could be addressed to extending the MILP models to related vehicle routing and different capacity allocation decisions.

Chapter 7 deals with a decision problem in the vehicle fleet replacement field, in which the bi-objective approach is recent. The problem is inspired by real-world case studies and consists in deciding every year whether and which vehicles of the fleet must be replaced with budget constraints to minimize total discounted costs (for buying, keeping, using, and replacing vehicles) and CO_2 emissions from fuel or energy consumption over the time horizon. The fleet manager aims at not only replacing the vehicles at the optimal point of life in the economic perspective, but also considering the electrification as well as other green replacement options for “greening” the fleet. In the preliminary phase of this research, we have explored ILP and DP approaches as promising tools for solving the problem to optimality, thus evaluating profitable and sustainable long-terms replacement plans. EE principles are employed to define the economic objective function. First, we have studied a specific problem with one family of assets of a single type, providing two network-flow-based ILP formulations and a DP model. In ILP, binary or integer variables are employed to represent the decisions on each arcs of the network for how may vehicles to buy and replace, considering constant demand over the periods. In the forward DP algorithm, a list of possible states (describing the age of each vehicles) is generated in every period from each state at the previous period. A particular effort has been addressed to the application of reduction rules to reduce the size of the DP network. Both approaches are exact and allow to obtain optimal solutions. So far, we have obtained very preliminary computational results on random instances. Then, we have formulated two alternative ILP models for the more general problem with multiple families and multiple replacement options. The research is currently addressed to the deep computational evaluations of our models, the definition of new reduction and dominance rules to improve the DP algorithm, and the exploration of alternative exact approaches, including column generation. The goal is to contribute to the asset replacement literature with outperforming algorithms. Lastly, we aim to solve two real-world case studies that involve large vehicle fleets of two important companies to obtain practical results.

Bilevel optimization for hierarchical decision-making The exploration of bilevel optimization in the third part of the thesis has shed light on the challenges and potential solutions for decision-making scenarios involving multiple actors with conflicting goals. In bilevel games, the leader decides first, and then the follower react to the leader’s decision by optimizing their decision. In Chapter 8, the survey has provided a comprehensive overview of bilevel optimization applications, emphasizing the importance of considering economic, environmental, and social dimensions in sustainability-driven decision-making. We have explored a wide range of sectors in which bilevel optimization (as well as traditional mathematical programming since longer) has been applied, including transportation and logistics, production planning and manufacturing, waste, material, and environment management, supply chains, and disaster prevention and response. We have selected more than 100 papers from relevant journals and aggregated them by subsectors or classes of notable decision-making problems. We have introduced the survey by outlining the main elements of bilevel programming problems developed for real-world applications, before summarizing all the selected papers in tables and text. We have addressed our focus on the identification of the decision makers in the UL and LL problems, their decisions, and their goals and perspectives (i.e., whether exclusively profitability/efficiency related or also addressed to environmental and social goals), but also on solution methods and practical results from case studies. Moreover, we have presented the typical bilevel formulations applied in the surveyed works. We have found many applications where the leaders in the UL are central or local authorities with rather intentional sustainability perspectives (i.e., multi-objective approach) and the followers are firms or end users with classical individual-oriented utility goals. Other cases involve two companies or two supply chains, one acting first and the other reacting after, which both take decisions on green initiatives. In the field of transportation, there is often one policy/decision maker minimizing the overall economic, environmental, or social impact (e.g., carbon emissions or hazmat risk minimization; social welfare maximization) of the overall system, and multiple individual end users, typically modelled in an equilibrium model (e.g., traffic equilibrium or market equilibrium) where each user maximizes their own utility functions. More rarely, sustainability goals are also considered in the followers' decision-making problem. The survey is still in a preliminary version, however we have been able to provide a clear picture of the current literature and draw some conclusive insights on practical implications and possible extensions in the field. In Chapter 9, we have focused our attention on bilevel problems within the energy sector, as a separate survey. We have considered green energy problems, namely in the areas of integrated energy systems, renewable energy markets, and emission trading and support schemes. Throughout our analysis, we identify prevalent bilevel models and solution methodologies. The bilevel game is typically between the investor in new power generation networks and the energy producers or consumers, or the government establishing emission trading systems and companies or users. By examining case studies, we illustrate the potential of bilevel optimization models to foster the expansion of renewable energy markets and the reduction of CO_2 emissions. In this survey, we have contributed to the analysis of a research topic of growing interest and highlighted the importance of sustainable energy optimization.

Final insights and future directions This thesis has highlighted the critical role of OR, particularly SO, in addressing global challenges such as climate crisis and achieving some UN SDGs. The integration of economic, environmental, and social perspectives in decision-making has been shown to enrich OR and contribute to improving not only efficiency and profitability of businesses, but also their impact on people and the planet. The methodologies provide a foundation for tackling decision-making problems with varying degrees of complexity. The developed models, algorithms, and survey presented in this thesis can help for future research in OR. The practical insights can stimulate further advancements in finding more sustainable and efficient solutions in complex decision-making contexts. As future research, we plan to extend and conclude the surveys on bilevel programming of Chapters 8-9 and define one or more bilevel problems with sustainable goals to be tackled by exact or heuristic algorithms. We also would like to extend the solution of vehicle fleet replacement problems faced in Chapter 7, finalizing exact algorithms and mathematical models and developing new efficient heuristics for the real-world case studies we faced.