

Methods for Combinatorial Optimization and Their Applications



a Ph.D. Thesis by A. M. BERNARDELLI

Doctoral Dissertation submitted to the Department of Mathematics of the University of Pavia and to the Faculty of Informatics of the Università della Svizzera Italiana in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computational Mathematics and Decision Sciences.

SUPERVISOR AT UNIVERSITY OF PAVIA:

Prof. Stefano Gualandi.

SUPERVISOR AT UNIVERSITÀ DELLA SVIZZERA ITALIANA:

Prof. Luca Maria Gambardella.



UNIVERSITÀ
DI PAVIA



Università
della
Svizzera
italiana

Methods for Combinatorial Optimization and Their Applications

Ambrogio Maria Bernardelli

Abstract. Combinatorial optimization is a fundamental area of mathematical optimization, dealing with problems that involve discrete structures and have a wide range of applications. As problem complexity grows, advancements in computational power, theoretical research, and innovative algorithmic frameworks have driven significant progress. This thesis examines combinatorial optimization methods across four key areas: integrality gaps, neural network training, stochastic optimization in healthcare, and energy management system optimization. First, the metric Steiner Tree problem is studied to investigate the integrality gap of linear programming relaxations. Lower bounds on the integrality gap of the bidirected cut formulation are established for small instances, and heuristic algorithms are developed to identify instances with large gaps. Additionally, structural insights and conjectures regarding the integrality gap are proposed. Second, combinatorial optimization techniques are applied to train few-bit neural networks, such as binarized and integer-valued networks. A multi-objective ensemble approach is introduced, improving robustness and sparsity and showcasing the potential of solver-based methods in low-data scenarios. This method is applied to different multiclassification problems. Next, a two-phase approach is developed for surgical scheduling, addressing uncertainties in surgery durations and emergency arrivals. Finally, distributionally robust optimization is applied to optimal power flow problems, coupled with an analysis of Jabr-like relaxations. The contribution of this thesis consists of presenting several complex problems, using their study as applications of diverse combinatorial optimization methods, highlighting the field's adaptability.

December 2024

I think our time is up.
I know. Hold my hand.
Hold your hand?
Yes. I want you to.
All right. Why?
Because that's what people do when they're waiting for the end of something.

— Cormac McCarthy, *Stella Maris*

Contents

Introduction	1
1 On the Integrality Gap of the Steiner Tree Problem	7
1.1 Introduction	7
1.2 Integrality gap of BCR for fixed n	11
1.2.1 Problem definition for computing the integrality gap of small instances	11
1.3 A novel formulation for the complete metric case	13
1.3.1 Properties of the complete metric formulation	15
1.3.2 The gap problem for the CM formulation	19
1.4 Heuristic enumeration of nontrivial vertices	21
1.4.1 Avoiding redundancy	21
1.4.2 Two heuristics procedure for vertices enumeration	25
1.5 Computational results	33
1.5.1 Lower bounds for the integrality gap for $n \leq 10$	33
1.5.2 A comparison between the two proposed heuristics	35
1.5.3 Beyond pure half-integer vertices	36
1.6 Conclusion and future works	38
2 Neural Network Training via Mixed Integer Linear Programming	41
2.1 Introduction	41
2.2 Related Works	43
2.3 Few-bit Neural Networks	45
2.3.1 Binarized Neural Networks	45
2.3.2 Integer Neural Networks	47
2.4 The <i>BeMi</i> ensemble	48
2.4.1 A multi-objective MILP model for training INNs	48
2.4.2 The <i>BeMi</i> structure	51
2.5 Empirical Study	54
2.5.1 Experiment 1	56
2.5.2 Experiment 2	58
2.5.3 Experiment 3	59
2.5.4 Experiment 4	60
2.5.5 Experiment 5	62
2.5.6 Experiment 6	63
2.6 Conclusion and Future Work	64

3	Stochastic optimization: scheduling of inpatient and outpatient surgeries	67
3.1	Introduction	67
3.2	Literature review	70
3.3	Problem statement	74
3.4	Mathematical models	75
3.4.1	Random variables	77
3.4.2	Advance scheduling: Chance Constrained Integer Programming model	78
3.4.3	Allocation scheduling: two-phase Stochastic Integer Programming model	82
3.4.4	Overall objective function	88
3.5	Methodology	88
3.5.1	Advance Scheduling: Monte Carlo sampling	89
3.5.2	Allocation Scheduling: SAA and N -fold SAA	89
3.5.3	Allocation scheduling: a genetic algorithm	90
3.6	Experimental setup	92
3.6.1	Specialties: MSS, surgical procedures, and distributions	92
3.6.2	Inpatients, outpatients, and emergencies: attributes and costs	94
3.6.3	Instance generation	97
3.6.4	Tests	97
3.7	Computational analysis	98
3.7.1	Results: advance scheduling	98
3.7.2	Results: allocation scheduling	103
3.7.3	Trade-off among objectives: parameter variation	106
3.7.4	Inpatient and outpatients: general insights	112
3.7.5	Emergency patients: simulation	115
3.8	Conclusions	116
4	On the Exactness of Jabr-like Models and Distributionally Robust Optimal Power Flow	121
4.1	Introduction	121
4.2	Distributionally robust OPF	122
4.2.1	Parameters and decision variables	124
4.2.2	Constraints definition	125
4.2.3	Inequality affine constraints	126
4.2.4	Conditional Value at Risk	128
4.2.5	Objective Function Definition	128
4.2.6	Pre-reduction model	129
4.2.7	Finite reduction	129
4.2.8	Numerical results	132
4.2.9	Conclusions	133
4.3	A multilinear approach on cycles	136
4.3.1	Standard multilinear relaxation	137

4.3.2	Handling affine trasformations	138
4.3.3	Other cuts	139
4.3.4	A possible generalization	141
4.3.5	Conclusions and future work	142
Conclusions		143
Appendices		145
A	Steiner Tree Problem: further details	145
A.a	Enumerating vertices with Polymake and weakness of the BCR formulation	145
A.b	Enumerating vertices with Polymake for the CM formu- lation	146
A.c	Pure one-quarter algorithm	146
B	Further Neural Network Experiments and Generalizations . . .	147
B.a	Complement to Neural Network Experiments	147
B.b	Ensamble Generalization	147
C	Linear formulation of the SMIP model $\mathcal{B}_{ij}^{II}(\omega)$	149
References		153

INTRODUCTION

Combinatorial optimization is a field within mathematical optimization that focuses on finding the best solution from a finite set of possible solutions, which are often discrete. Unlike continuous optimization, where the decision variables can take any value within a given range, combinatorial optimization problems involve discrete decisions, such as selecting elements from a set, ordering tasks, or partitioning a graph. The applications of combinatorial optimization are vast, covering fields such as logistics, network design, scheduling, and machine learning.

Over the years, combinatorial optimization has evolved to address a wide range of increasingly complex problems. These problems often involve large-scale instances, complicated constraints, and the need to provide high-quality solutions within feasible time limits. As the complexity of the problems has grown, so too have the methods developed to solve them. In this context, various algorithmic frameworks, from exact methods to approximate heuristics, have been proposed to tackle these challenging optimization tasks.

This evolution has been fueled by several factors: the exponential growth in computational power, breakthroughs in theoretical research, and the applicability of combinatorial optimization to emerging fields. For example, advances in artificial intelligence and machine learning have introduced new challenges, such as optimizing neural network architectures or neural network verification, which inherently involve combinatorial decision-making. Similarly, fields like healthcare and energy management systems increasingly rely on combinatorial approaches to solve highly specialized optimization problems.

One characteristic of combinatorial optimization problems is their inherent difficulty, particularly in achieving near-optimal solutions, and a key concept in characterizing this difficulty is given by the integrality gap of a linear programming formulation, defined as the supremum among all the ratios between the optimal value of an integer solution and the optimal value of its natural LP relaxation. It quantifies the difference between the optimal solutions of the integer program and its corresponding linear programming relaxation. This measure provides insight into how well the relaxation approximates the original problem and serves as a theoretical bound on the performance of approximation algorithms derived from the relaxation.

In particular, given a combinatorial problem, one can start studying the integrality gap of one of its linear programming formulations by focusing on small instances. By focusing on the behavior of such instances, one can identify

patterns and design families of instances that exhibit the largest possible integrality gap. This analysis helps understand the limitations of linear programming relaxations as approximations and constructing worst-case scenarios that test the boundaries of algorithmic performances. Furthermore, studying such instances helps develop stronger relaxations or alternative techniques for narrowing the integrality gap, contributing to improved approximation algorithm design. A standard technique for evaluating the integrality gap of small instances makes use of the complete enumeration of the fractional vertices of the polytope defined by the linear relaxation of the integer programming formulation. Such a technique reduces the study of infinite instances to the solving of a finite number of linear programs. The downside of this method is the fact that the number of vertices grows exponentially with the size of the instances, and so, while the complete enumeration of the vertices of a polytope is possible up to a certain instance dimension, heuristic approaches have to be employed to find promising vertices. These heuristic approaches are based both on theoretical results, which regards the structure of some families of fractional vertices, and computational results, which shows particular patterns in vertices with high values of integrality gap.

Chapter 1 deals with the study of the integrality gap of a linear formulation of a classical combinatorial problem: the Steiner Tree problem (STP) on graphs. In a general setting, the STP plays a central role in network design problems, design of integrated circuits, location problems, and more recently even in machine learning, systems biology, and bioinformatics. It asks for designing a network that interconnects a given set of points (referred to as terminals) at minimum cost. When the choice of non-terminal points is limited to a finite set of points, the problem is formulated on an edge-weighted graph (with terminal points representing a subset of nodes) with the goal of finding a subtree that interconnects all the terminals at minimum cost. In this chapter, we study the metric Steiner Tree problem on graphs focusing on computing lower bounds for the integrality gap of the bi-directed cut (BCR) formulation and introducing a stronger formulation, called Complete Metric (CM), specifically designed to address the weakness of the BCR formulation on metric instances. A key contribution of this work is extending the Gap problem, previously explored in the context of the Traveling Salesman problems, to the metric Steiner Tree problem. To tackle the Gap problem for Steiner Tree instances, we establish several structural properties of the CM formulation. Computationally, we exploit these structural properties to design two complementary heuristics for finding nontrivial small metric Steiner instances with a large integrality gap. We present several vertices for graphs with a number of nodes ≤ 10 , which realize the best-known lower bounds on the integrality gap for the CM and the BCR formulations. We also present two new conjectures on the integrality gap of the BCR and CM formulations for small graphs.

This chapter presents the results obtained with coauthors E. Vercesi, S.

Gualandi, M. Mastrolilli, and L. M. Gambardella [20]. The author’s contributions to this work consist in a deep analysis of the structural properties of the vertices of the CM formulation, and in the resulting designing of the algorithms for the vertex search.

The STP on graphs is part of a class of purely combinatorial problems, whose formulation can be naturally written as a combinatorial optimization problem. However, an increasing number of problems that were originally not purely combinatorial are evolving to have combinatorial formulations, allowing them to benefit from the powerful techniques and methods developed in combinatorial optimization. This shift enables the application of discrete optimization approaches to a broader range of domains, in particular, to machine learning. By recasting machine learning problems in combinatorial terms, it becomes possible to leverage the precision and efficiency of combinatorial solvers. In particular, training neural networks (NNs) using combinatorial optimization solvers has gained attention in recent years. In low-data settings, the use of state-of-the-art mixed integer linear programming solvers, for instance, has the potential to exactly train a NN while avoiding computing-intensive training and hyperparameter tuning and simultaneously training and sparsifying the network. In Chapter 2, we study the case of few-bit discrete-valued neural networks, both binarized neural networks (BNNs) whose values are restricted to ± 1 and integer-valued neural networks (INNs) whose values lie in the range $\{-P, \dots, P\}$. Few-bit NNs receive increasing recognition because of their lightweight architecture and ability to run on low-power devices: for example, being implemented using Boolean operations. This chapter proposes new methods to improve the training of BNNs and INNs. Our contribution is a multi-objective ensemble approach based on training a single NN for each possible pair of classes and applying a majority voting scheme to predict the final output. Our approach results in the training of robust sparsified networks whose output is not affected by small perturbations on the input and whose number of active weights is as small as possible. We empirically compare this BeMi approach with the current state of the art in solver-based NN training and with traditional gradient-based training, focusing on BNN learning in few-shot contexts. We compare the benefits and drawbacks of INNs versus BNNs, bringing new light to the distribution of weights over the $\{-P, \dots, P\}$ interval. Finally, we compare multiobjective versus single-objective training of INNs, showing that robustness and network simplicity can be acquired simultaneously, thus obtaining better test performances.

This chapter presents the results obtained with coauthors S. Milanese, S. Gualandi, H. C. Lau, and N. Yorke Smith [19]. A preliminary report of this work also appeared at the Learning and Intelligent Optimization 2023 conference [17]. The author’s contributions to this work consist in the designing of the One-Vs-One scheme used for the ensemble and the utilization of a multi-objective approach for the single NN.

NN training can be viewed as a specific type of stochastic optimization problem, where the training process is conducted over a sampled subset of the full dataset. Each sample represents a possible scenario from the broader data distribution, and the objective is to optimize model parameters such that the network generalizes well across all potential scenarios. Thus, we can see how incorporating stochasticity into optimization models allows for the development of solutions that are not only feasible under uncertain conditions but also robust and adaptable to a range of possible scenarios. Stochastic approaches, such as scenario-based optimization, stochastic programming, and probabilistic heuristics, enable decision-makers to balance trade-offs between optimality and reliability. By accounting for randomness, these methods can help identify strategies that minimize risk, improve performance, and enhance resilience in dynamic and uncertain environments, making stochasticity an indispensable aspect of modern combinatorial optimization.

In particular, one of the most studied classes of stochastic problems is stochastic scheduling problems, which arise frequently in healthcare management. With the advancement of surgery and anesthesiology in recent years, surgical clinical pathways have changed significantly, with an increase in outpatient surgeries. However, the surgical scheduling problem is particularly challenging when inpatients and outpatients share the same operating room blocks, due to their different characteristics in terms of variability and preferences. In Chapter 3, we present a two-phase stochastic optimization approach that takes into account such characteristics, considering multiple objectives and dealing with uncertainty in surgery duration, arrival of emergency patients, and no-shows. Chance Constrained Integer Programming and Stochastic Mixed Integer Programming are used to deal with the advance scheduling and the allocation scheduling, respectively. Since Monte Carlo sampling is inefficient for solving the allocation scheduling problem for large-size instances, a genetic algorithm is proposed for sequencing and timing procedures. Finally, a quantitative analysis is performed to analyze the trade-off between schedule robustness and average performance under the selection of different patient mixes, providing general insights for operating room scheduling when dealing with inpatients, outpatients, and emergencies.

This chapter presents the results obtained with coauthors L. Bonasera, D. Duma, and E. Vercesi [16]. A preliminary report of this work achieved second place at the 14th AIMMS-MOPTA Optimization Modeling Competition. The author's contributions to this work consist in the studying of the different probability distribution mixtures for the surgery durations and the designing of the advance scheduling model.

Another area in which stochastic optimization problems naturally arise is power management. In particular, problems in contexts such as energy distribution, battery management, and smart grids, becomes increasingly complex in the stochastic optimization framework. In these scenarios, the objective

is to efficiently allocate power resources, manage demand, and optimize energy usage while accounting for the uncertainty and variability inherent in energy production and consumption. Stochasticity arises from factors such as fluctuating renewable energy sources (e.g., solar and wind), uncertain demand patterns, and the reliability of energy storage systems. This uncertainty introduces significant challenges in decision-making, as optimal power management strategies must account for a range of potential future scenarios.

In particular, even in the deterministic framework, optimal power flow (OPF) problems are notoriously challenging due to their inherent complexity and the large solution space they involve. These problems seek to determine the optimal settings of control variables (such as generator outputs, transformer taps, and voltage levels) to minimize costs, such as generation or transmission losses, while satisfying a set of system constraints like power balance, voltage limits, and line capacities. The difficulty arises from the non-linear, non-convex nature of the power system equations, which can lead to multiple local minima and a combinatorial explosion in the number of potential configurations, especially in large-scale networks. As a result, solving OPF problems requires sophisticated optimization techniques that can navigate this vast search space efficiently, often incorporating both continuous and discrete variables. The problem becomes undoubtedly more complex in the stochastic optimization framework.

Chapter 4 deals with the analysis of a distributionally robust optimization approach to an OPF problem, integrating real-world data regarding production of energy from wind energy sources provided by CESI – Centro Elettrotecnico Sperimentale Italiano. The robust approach is performed over an ambiguity set defined as the ball in the space of probability distributions centered in an empirical distribution (the collected data) and evaluated using the Wasserstein distance. Finite reduction results are applied to obtain a model that is then tested on classical instances modified with renewable energy sources. In addition, different strengthening of the classical Jabr relaxation are studied, leveraging topology results regarding the cycles of the network. In particular, linear relaxations of multilinear constraints are discussed, presenting tailored methods and possible generalizations of classical techniques.

This chapter presents the results obtained with coauthors G. Riccardi and S. Gualandi, part of which constituted the focus of Riccardi’s Master’s Thesis in Mathematics. The author’s contributions to this work consist in application of a distributionally robust optimization framework to an OPF problem, and in the studying of different linearization methods for multilinear terms that naturally arises whenever a complex network topology is considered in OPF problems.

For more detailed introductions and discussions of the various problems in relation to the literature, we refer to Sections 1.1, 2.1, 3.1 and 4.1.

ON THE INTEGRALITY GAP OF THE STEINER TREE PROBLEM

1.1 Introduction

Given a graph $G = (V, E)$ with $n = |V| = |\{1, \dots, n\}|$ nodes and cost $c_{ij} \geq 0$ on each edge $\{i, j\} \in E$, and a subset of nodes $T \subset V$ with $t = |T| \geq 2$, the Steiner Tree Problem (STP) consists of finding the minimum-cost tree that spans T . The nodes in T are called *terminals*, and those in $V \setminus T$ are called *Steiner nodes*. Note that, by a relabelling of the vertices, we might just assume that $T = \{1, \dots, t\}$. In the following, we will often denote simply by G an instance of the STP, that is, the graph, the edge costs, and the set of terminals. The STP is NP-Hard, and the corresponding decision problem is NP-Complete [74]. The best-known polynomial-time algorithm for the STP guarantees an approximation ratio of 1.39 [33], and improving this ratio is still an open problem. The most well-known cases are when $|T| = 2$, which is the shortest path problem, and for $|T| = n$, which is minimum spanning tree.

STP is often solved via integer linear programming with several formulations that have been proposed through the years. For a catalog of them, we refer to [55], while for more recent surveys, to [79, 90].

One of the most promising formulations is the Bidirected cut formulation (BCR), introduced by Edmonds [48]. The core of the BCR formulation consists of fixing a root node $r \in T$, replacing each undirected edge $\{i, j\}$ with two oriented arcs (i, j) and (j, i) , and introducing 0–1 flow variables x_{ij} for each arc. If we denote by A the set of oriented arcs, and by $\delta^-(W) := \{(i, j) \in A \mid i \notin W, j \in W\}$ the set of arcs entering $W \subset V$, the BCR polytope is defined as follows:

$$P_{BCR}(n, t) := \{x \in \mathbb{R}_{\geq 0}^m : \quad (1.1a)$$

$$x_{ij} + x_{ji} \leq 1, \quad \{i, j\} \in E, \quad (1.1b)$$

$$x(\delta^-(W)) \geq 1, \quad W \subset V \setminus \{r\}, W \cap T \neq \emptyset \quad (1.1c)$$

where $m = |A|$, and we define the set of integer points of $P_{BCR}(n, t)$ as

$$S_{BCR}(n, t) := P_{BCR}(n, t) \cap \{0, 1\}^m. \quad (1.2)$$

Given a STP instance G with n nodes and t terminals and with c as edge costs, we will define with

$$\text{STP}(G) := \min\{c^\top x \mid x \in S_{BCR}(n, t)\} \quad (1.3)$$

the optimal value of one of its integer solutions and with

$$\text{opt}_{BCR}(G) := \min\{c^\top x \mid x \in P_{BCR}(n, t)\} \quad (1.4)$$

the optimal value of the linear relaxation of BCR. We will use the same notation for all of the formulations presented in this work. Note that we will drop the arguments (n, t) when unnecessary or clear from the context, for better readability. Note also that the value (1.3) is independent from the formulation.

Despite their exponential number, the cut constraints (1.1c) can be separated in polynomial time using any max flow or min-cut algorithm, as detailed in [79]. Note that it has been shown that the optimal value of BCR is independent of the choice of the root [55]. The state-of-the-art (parallel) implementation of a more tight version of the BCR model can be found in SCIP-Jack [52]. Other more recent approaches are reviewed in [90].

Whenever the graph G is complete, and the edge costs are metric, we have a complete metric Steiner Tree problem. The edge costs define a metric if they satisfy the following properties: (i) $c_{ij} = 0$ if and only if $i = j$; (ii) $c_{ij} \geq 0$ (*positivity*), (iii) $c_{ij} = c_{ji}$ (*symmetry*), (iv) $c_{ij} \leq c_{ik} + c_{jk}$ (*triangle inequality*). Metric STP instances are relevant in particular for VLSI circuit design [108], and efficient combinatorial algorithms exist for rectilinear costs [66] and for packing Steiner trees [59]. A review of results for metric (and rectilinear) STP is contained in [66].

In approximation algorithms, we are interested in studying the *integrality gap* of an integer formulation, defined as the supremum among all the ratios between the optimal value of an integer solution and the optimal value of its natural LP relaxation. We now define the integrality gap of the BCR formulation as the following

$$\alpha := \sup_G \frac{\text{STP}(G)}{\text{opt}_{BCR}(G)}. \quad (1.5)$$

For the BCR, the exact value of α is unknown, but it is bounded below by $\frac{6}{5}$ [138], which improved the previous bound of $\frac{36}{31} \cong 1.161$ in [33]. Until 2024, the best-known upper bound for the integrality gap was two. Recently, [34] demonstrated that this upper bound can be improved to 1.9988.

Note that proving that $\alpha < 1.39$ would lead to a better approximation algorithm with respect to the state of the art. The lower bound introduced by [33] is based on a recursive family of instances, depending on a parameter p and having an integrality gap which tends asymptotically to $\frac{36}{31}$ for $n \rightarrow \infty$.

Main Contributions This paper presents a novel formulation for the complete metric STP, called the Complete Metric (CM) formulation. This formulation exploits the metric costs to define a polytope, denoted by P_{CM} , having a smaller number of vertices compared to the polytope P_{BCR} implied by the BCR formulation. The main motivation for introducing our formulation is to enable a study on the integrality gap of small-size instances of the Steiner

Tree problem by adapting the approach designed for the Symmetric TSP and presented in [30, 31]. Without our new CM formulation, it would be nearly impossible to use the method of [31] due to the number of vertices of P_{BCR} , which includes a huge number of feasible vertices for the cut constraints but which will never be optimal for any *metric* cost vector. For instance, Table 1.1 reports the number of feasible and optimal vertices of P_{BCR} and P_{CM} , computed by complete enumeration using Polymake, for instances with 4 and 5 nodes and 3 or 4 terminals, showing the potential impact of our approach on the overall number of vertices.

The core intuition of our new CM formulation is that in a complete metric graph, any Steiner node is visited only if its outdegree is at least 2, because if the indegree and outdegree of a Steiner node are both equal to 1, then an optimal solution with a smaller cost that avoids detouring in that node exists. Note that this requirement is profoundly different from asking a Steiner node not to be a leaf, which is a much weaker condition. The existence of such a solution is guaranteed by the property that the graph is metric and complete. Indeed, such a solution may not exist in a non-complete graph. Using these relations on the degree of Steiner nodes, we introduce a new family of constraints to the BCR formulation, reflecting our main intuition.

The main contributions of this paper are as follows:

1. We prove in Theorem 1 that our new polytope P_{CM} contains among the integer vertices only those that could be optimal for metric costs. Given an integer vertex, we describe the cost vector that makes that vertex optimal.
2. We prove in Lemma 3 a connectedness property of the points of P_{CM} , and in Lemma 4 we set an upper bound on the number of edges in integer points.
3. We characterize in Lemmas 6 to 8 isomorphic vertices of polytopes of different dimensions, namely, we link vertices of the polytope corresponding to the STP with n nodes and t terminals with the vertices corresponding to the STP with $n + 1$ nodes and t terminals, and vice versa.
4. Exploiting the previous results, we introduce two new heuristic algorithms for enumerating the vertices of P_{CM} . Using these two algorithms, we compute vertices of P_{CM} and P_{BCR} with the largest known integrality gap for instances with up to 10 vertices.

To the best of our knowledge, this work is the first attempt to extend the work of [31] to the metric Steiner Tree problem. Note that an interesting STP instance with a small value of n (i.e., $n = 15$), but with a large integrality gap (i.e., equal to $\frac{8}{7}$) is the Skutella's graph shown in Figure 1.1 and reported in [82]. In our computational results, we will present other interesting instances with less than 10 nodes but having a large integrality gap, see Figure 1.4.

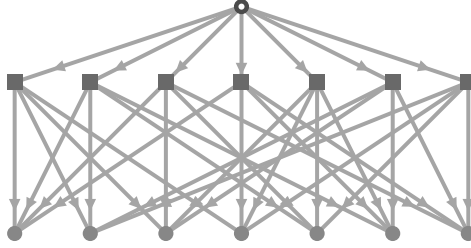


Figure 1.1: Small STP instance with a large integrality gap: Skutella’s graph with $n = 15$, $t = 8$, and $\alpha = \frac{8}{7}$ [82]. The hollow circle represents the root, the circles represent the terminals, and the squares represent the Steiner nodes. Every arc correspond to a variable x_{ij} of value equal to $\frac{1}{4}$.

Table 1.1: Number of feasible and optimal vertices for P_{BCR} and P_{CM} . While the BCR polytope has several (feasible) vertices that cannot be optimal for any metric cost, the CM polytope does not suffer this issue (and it implicitly reduces the number of isomorphic vertices).

n	t	P_{BCR}		P_{CM}	
		feasible	optimal	feasible	optimal
4	3	256	70	4	4
5	3	28 345	3 655	5	5
5	4	24 297	3 645	44	44

Outline. The outline of this paper is as follows. Section 1.2 reviews the approach introduced in [14] for computing the integrality gap of small instances ($n \leq 10$) of the Traveling Salesman Problem and presents how to apply a similar approach to the BCR formulation of the STP. The critical step is the complete enumeration of the vertices of the BCR polytope, which are several thousand already for $n = 5$, as shown in Table 1. Section 1.3 presents the CM formulation, and we prove interesting properties of the corresponding polytope, which allows us to apply the methodology of [14] to look for small instances of STP with a large integrality gap. In Section 1.4, we observe that the exhaustive enumeration of vertices is intractable for $n \geq 6$, and we present two heuristic procedures for generating vertices of P_{CM} , exploiting graph isomorphism. In Section 1.5, we present the vertices for graph $n \leq 10$, realizing the best-known lower bounds on the integrality gap for the CM and the BCR formulation. We conclude the paper with a perspective on future works.

1.2 Integrality gap of BCR for fixed n

In this section, we present our strategy to compute the integrality gap of the BCR formulation for STP, which is based on the approach introduced in [14, 50].

1.2.1 Problem definition for computing the integrality gap of small instances

Let $\mathcal{G}_{n,t} := \{G \mid \text{metric with } n \text{ nodes and } t \text{ terminals}\}$. We define

$$\alpha_G := \frac{\text{STP}(G)}{\text{opt}_{BCR}(G)} \quad (1.6)$$

$$\alpha_{n,t} := \sup_{G \in \mathcal{G}_{n,t}} \alpha_G. \quad (1.7)$$

Note that α_G is the integrality gap of a given instance of metric STP, while $\alpha_{n,t}$ is the maximum integrality gap once we fix both the cardinality of T and V . Clearly, the integrality gap (1.5) is equal to

$$\alpha = \sup_{n,t} \alpha_{n,t}. \quad (1.8)$$

As mentioned in the introduction, the two cases $t = 2$ and $t = n$ can be solved in polynomial time for every n , since for $t = 2$ the problem reduces to the shortest path (SP) problem, while for $t = n$ it reduces to the minimum spanning tree (MST) problem. In general, the polynomial time algorithms for SP (e.g., [44]) and MST (e.g., [109]) do not use the IP formulation. Hence, the existence of a polynomial time algorithm does not naturally imply an integral formulation when coming to the BCR formulation. For $t = n$, the polyhedron is proven to be integral [48], while [55] shows the same result for $t = 2$, and hence in these two cases $\alpha = 1$. However, the exact value of α for $2 < t < n$ is unknown.

In this work, we extend the approach presented in [14, 50] for TSP for computing the exact value of α for the BCR formulation. Similarly to [14], if we divide the costs $c_{ij}, i, j \in V$ of an instance G for the optimal value $\text{STP}(G)$, we obtain another instance G' , having an optimal value $\text{STP}(G) = 1$ but the same set of optimal solutions. Hence, defining $\mathcal{G}_{n,t}^1 := \{G \mid \text{metric with } n \text{ nodes and } t \text{ terminals, } \text{STP}(G) = 1\}$, we obtain

$$\alpha_{n,t} := \sup_{G \in \mathcal{G}_{n,t}^1} \frac{1}{\text{opt}_{BCR}(G)},$$

and hence

$$\frac{1}{\alpha_{n,t}} := \inf_{G \in \mathcal{G}_{n,t}^1} \text{opt}_{BCR}(G). \quad (1.9)$$

Note that (1.9) leads to an optimization problem having linear constraints but a quadratic objective function (note that c_e and x_{ij} are both decision

variables) that can be formulated as follows:

$$\min \sum_{\{i,j\} \in E} c_e(x_{ij} + x_{ji}) \quad (1.10a)$$

$$\text{s.t. } x_{ij} + x_{ji} \leq 1, \quad e = \{i, j\} \in E, \quad (1.10b)$$

$$x(\delta^-(W)) \geq 1, \quad W \subset V \setminus \{r\}, W \cap T \neq \emptyset, \quad (1.10c)$$

$$0 \leq x_{ij} \leq 1, \quad \forall i, j \in V, i \neq j, \quad (1.10d)$$

$$c_{ij} \geq 0, \quad \forall \{i, j\} \in E, \quad (1.10e)$$

$$c_{ij} \leq c_{ik} + c_{jk}, \quad \forall \{i, j\}, \{i, k\}, \{j, k\} \in E, \quad (1.10f)$$

$$\sum_{\{i,j\} \in E} c_e(\bar{\tau}_{ij} + \bar{\tau}_{ji}) \geq 1, \quad \forall \bar{\tau} \in S_{BCR}(n, t). \quad (1.10g)$$

Constraints (1.10b)–(1.10d) ensures the feasibility of vector x , while constraints (1.10e)–(1.10f) ensure the property of c being metric. Constraint (1.10g) ensures $\text{STP}(G) = 1$, with the costs of G defined by c . Our preliminary computational results show that (1.10a)–(1.10g) are intractable even for $n = 5$ and $t = 3, 4$. Therefore, we proceed as done in [14, 31, 50], exploiting the vertex representation of P_{BCR} . Similarly to [14, 31, 50], and by recalling that for each cost vector c , there exists an optimal solution attained at a vertex, we can re-write (1.10a)–(1.10g) as a *linear program* for each vertex \bar{x} of $P_{BCR}(n, t)$.

$$\min \sum_{\{i,j\} \in E} c_e(\bar{x}_{ij} + \bar{x}_{ji}) \quad (1.11a)$$

$$\text{s.t. } c_{ij} \geq 0, \quad \forall \{i, j\} \in E, \quad (1.11b)$$

$$c_{ij} \leq c_{ik} + c_{jk}, \quad \forall \{i, j\}, \{i, k\}, \{j, k\} \in E, \quad (1.11c)$$

$$\text{the optimal solution of } c \text{ is attained at } \bar{x}, \quad (1.11d)$$

$$\sum_{\{i,j\} \in E} c_e(\bar{\tau}_{ij} + \bar{\tau}_{ji}) \geq 1, \quad \forall \bar{\tau} \in S_{BCR}(n, t). \quad (1.11e)$$

and define $\text{Gap}(\bar{x})$ as the optimal value of (1.11).

As in [14, 50], we observe that constraint (1.11d) can be formulated using the complementary slackness conditions. Such conditions ensure that \bar{x} belongs to the point minimizing the STP at cost c . A detailed formulation is presented below

$$\min \sum_{\{i,j\} \in E} c_e(\bar{x}_{ij} + \bar{x}_{ji}) \quad (1.12a)$$

$$\text{s.t. } c_{ij} \leq c_{ik} + c_{jk}, \quad \forall \{i, j\}, \{i, k\}, \{j, k\} \in E, \quad (1.12b)$$

$$y_e + \sum_{(i,j) \in \delta^-(W)} z_W + d_{ij} \leq c_e, \quad \forall (i, j) \in A, \quad (1.12c)$$

$$y_e = 0, \quad \forall (i, j) \text{ s.t. } \bar{x}_{ij} + \bar{x}_{ji} < 1, e = \{i, j\} \in E, \quad (1.12d)$$

$$z_W = 0, \quad \forall W \subset V \setminus \{r\}, W \cap T \neq \emptyset \text{ s.t.} \quad \sum_{(i,j) \in \delta^-(W)} \bar{x}_{ij} > 1, \quad (1.12e)$$

$$d_{ij} = 0, \quad \forall (i,j) \in A \text{ s.t. } \bar{x}_{ij} = 1, \quad (1.12f)$$

$$y_e + \sum_{(i,j) \in \delta^-(W)} z_W + d_{ij} - c_e = 0, \quad \forall (i,j) \in A \text{ s.t. } \bar{x}_{ij} > 0, \quad (1.12g)$$

$$y_e, d_{ij}, d_{ji} \leq 0, \quad \forall e = \{i,j\} \in E, \quad (1.12h)$$

$$z_W \geq 0, \quad \forall W \subset V \setminus \{r\}, W \cap T \neq \emptyset, \quad (1.12i)$$

$$c_{ij} \geq 0, \quad \forall \{i,j\} \in E, \quad (1.12j)$$

$$\sum_{\{i,j\} \in E} c_e(\bar{\tau}_{ij} + \bar{\tau}_{ji}) \geq 1, \quad \forall \bar{\tau} \in S_{BCR}(n, t). \quad (1.12k)$$

1.3 A novel formulation for the complete metric case

Scip-Jack [52], the state-of-art solver for the (Graphic) STP, relies on a modified version of the BCR formulation, firstly introduced in [79]. The formulation used by Scip-Jack (SJ) is presented through its associated polytope:

$$P_{SJ}(n, t) := \{x \in [0, 1]^m : \quad (1.13a)$$

$$x(\delta^-(W)) \geq 1 \quad W \subset V \setminus \{r\}, W \cap T \neq \emptyset, \quad (1.13b)$$

$$x(\delta^-(r)) = 0, \quad (1.13c)$$

$$x(\delta^-(v)) = 1, \quad v \in T \setminus \{r\}, \quad (1.13d)$$

$$x(\delta^-(v)) \leq 1, \quad v \in S, \quad (1.13e)$$

$$x(\delta^-(v)) \leq x(\delta^+(v)), \quad \forall v \in S, \quad (1.13f)$$

$$x(\delta^-(v)) \geq x_a, \quad \forall a \in \delta^+(v), v \in S \quad (1.13g)$$

where $\delta^+(W) := \{(i, j) \mid i \in W, j \notin W\}$.

Constraints (1.13c)–(1.13e) describe the inflow of every node: the first equation ensures that no inflow is present in the root, the second equation ensures that the inflow of terminal nodes is exactly equal to 1, since every terminal must be reached, and the third equation ensures that the inflow of non-terminal nodes is smaller or equal than 1 since a non-terminal node may or may not be part of an optimal solution. Note that both terminal and non-terminal nodes have an inflow of at most 1 so that at most one path exists from the root to any node. Constraint (1.13f) ensures that non-terminal nodes cannot be leaves of the solution. Constraint (1.13g) ensures that no flow is generated from non-terminal nodes. Notice that this formulation is not specific to the metric we want to attack, as illustrated by the example below.

Example 1. Let $G = (V, E)$ be a complete metric graph with $V =$

$\{1, 2, 3, 4, 5\}$ and let $T = \{1, 2\}$. Define x as the following

$$x_{ij} = \begin{cases} 1, & \text{if } (i, j) \in \{(1, 2), (3, 4), (4, 5), (5, 3)\}, \\ 0, & \text{else.} \end{cases} \quad (1.14)$$

We have that x is feasible for the SJ formulation with $r = 1$, but it is never optimal for any metric cost since by setting $x_{3,4} = x_{4,5} = x_{5,3} = 0$ we obtain a feasible solution with a strictly smaller cost. Note that, in particular, x is not connected.

To prevent this issue, we introduce a stronger formulation tailored to the Complete Metric (CM) case, which is presented below through its associated polytope

$$P_{CM}(n, t) := \{x \in [0, 1]^m : \quad (1.15a)$$

$$x(\delta^-(W)) \geq 1, \quad W \subset V \setminus \{r\}, W \cap T \neq \emptyset, \quad (1.15b)$$

$$x(\delta^-(r)) = 0, \quad (1.15c)$$

$$x(\delta^-(v)) \leq 1, \quad v \in V \setminus \{r\}, \quad (1.15d)$$

$$2x(\delta^-(v)) \leq x(\delta^+(v)), \quad \forall v \in S. \quad (1.15e)$$

In particular, in our new formulation, the left-hand side of Constraint (1.13f) is multiplied by 2. This ensures that a non-terminal node is visited only if its outflow is at least 2. The idea is that, in a complete metric graph, if the inflow and the outflow of a non-terminal node are both equal to 1, then there exists an optimal solution with a smaller cost that avoids detouring in that node. The existence of such a solution is guaranteed by the property that the graph is metric and complete. Note that such a solution may not exist in a non-complete graph, for example, when $G = (V, E)$ with $V = \{1, 2, 3\}$, $E = \{\{1, 3\}, \{2, 3\}\}$ and $T = \{1, 2\}$. We also avoid adding the equivalent of Constraint (1.13g) because of the following lemma.

Lemma 1. *When dealing with positive costs, Constraint (1.13g) is redundant even for the simpler BCR formulation.*

Before proving Lemma 1, we recall the *Multi Commodity Flow* (MCF) formulation [41]:

$$\min \sum_{\{i,j\} \in E} c_e(x_{ij} + x_{ji}) \quad (1.16a)$$

$$\text{s.t. } x_{ij} + x_{ji} \leq 1, \quad e = \{i, j\} \in E, \quad (1.16b)$$

$$f^t(\delta^-(v)) - f^t(\delta^+(v)) = \begin{cases} -1, & v = r \\ 1, & v = t \\ 0, & v \neq t, r \end{cases} \quad v \in V, t \in T \setminus \{r\} \quad (1.16c)$$

$$f_{ij}^t \leq x_{ij} \quad \forall (i, j) \in A, \quad (1.16d)$$

$$f_{ij}^t, x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A. \quad (1.16e)$$

This formulation has a few interesting properties that we use in the following proof of Lemma 1.

Proof of Lemma 1. Let x_{ij} be an optimum vertex for the BCR formulation with a positive cost c . By [41, Theorem 3.2], in particular, because of the equivalence

$$\min\{c^\top x \mid x \in P_{MCF}(n, t)_{|x}\} = \min\{c^\top x \mid x \in P_{BCR}(n, t)\},$$

there exists a configuration of variables f_{ij}^t such that $f_{ij}^t \leq x_{ij}$ for every $t \in T$, $i, j \in V$ and $\sum_i f_{ij}^t - \sum_i f_{ji}^t = 0$ for every $t \in T$, $j \in S$. Because x_{ij} is optimal for strictly positive costs, we have that $x_{ij} = \max_{t \in T} f_{ij}^t$ and so there exists $t_{ij} \in T$ such that $x_{ij} = f_{ij}^{t_{ij}}$. Now, let $k \in S$. For every $a \in \delta^+(k)$, that is, for every $l \in V \setminus \{k\}$ we have that

$$\begin{aligned} x_a &= x_{kl} && \text{by definition} \\ &= f_{kl}^{t_{kl}} && \text{by maximization} \\ &\leq \sum_i f_{ki}^{t_{kl}} && \text{by nonnegativity} \\ &= \sum_i f_{ik}^{t_{kl}} && \text{by (1.16c)} \\ &\leq \sum_i x_{ik} && \text{by (1.16d)} \\ &= x(\delta^-(k)) && \text{by definition} \end{aligned}$$

which is equivalent to Constraint (1.13g). \square

Before discussing how we use this formulation to retrieve information regarding the integrality gap of the BCR formulation, we list some properties of the CM formulation that we consider interesting.

1.3.1 Properties of the complete metric formulation

We first show that for a particular configuration of complete metric graphs, namely, graphs with no triples of collinear points, the set of integer solutions of the SJ formulation coincides with the set of integer solutions of the CM formulation.

Lemma 2. *Let G be a complete metric graph with $c \in \mathbb{R}^{(n-1) \times n}$ defining the edge weights and let $x \in S_{SJ}$ be optimal for the cost vector c . If*

$$c_{ij} < c_{ik} + c_{kj} \quad \forall \{i, j\}, \{i, k\}, \{j, k\} \in E, \quad (1.17)$$

then $x \in S_{CM}$ and it is optimal for the same cost vector. Moreover, if $y \in S_{CM}$ is optimal for G , then $y \in S_{SJ}$ and it is optimal for G .

Proof. We clearly have that $S_{CM} \subset S_{SJ}$. Suppose then by contradiction that there exists $x \in S_{SJ}$, optimal for G such that $x \notin S_{CM}$. Because of the constraints that describe the two models, this solution x must verify

$$\sum_{i \neq j} x_{ij} \leq \sum_{k \neq j} x_{jk}, \quad 2 \cdot \sum_{i \neq j} x_{ij} > \sum_{k \neq j} x_{jk},$$

for a certain $j \in V \setminus T$. It follows that there exists $i, k \in V$ such that $x_{ij} = x_{jk} = 1$. Since we are in a complete graph, setting these two variables to zero and setting $x_{ik} = 1$ gives us a feasible solution, which is also of smaller cost because of hypothesis (1.17), which is in contradiction with the optimality of x .

Let $y \in S_{CM}$ optimal for G . Clearly, $y \in S_{SJ}$ is feasible for SJ. Suppose by contradiction that there exists $z \in S_{SJ}$ such that $c^\top z < c^\top y$. For the first part of the proof, we have that $z \in S_{CM}$ optimal for CM, which contradicts the optimality of y . \square

Observation 1. Note that, without hypothesis (1.17), we can say that given a complete metric graph G and $x \in S_{SJ}$ optimal for G , there exists $x' \in S_{SJ}$ optimal for G such that $x' \in S_{CM}$ optimal for G . In particular, x' is chosen as one of the optimal integer solutions of SJ that avoids detouring into non-terminal nodes, where detouring into a node means entering with one edge and exiting with one edge. The same reasoning can be applied to the BCR formulation.

Observation 2. Note that Lemma 2 does not hold when removing the integrality hypothesis. Take, for example, as graph G the metric completion of the instance **se03** of the SteinLib [80]. We have that

$$\text{opt}_{SJ}(G) = 11 < 12 = \text{opt}_{CM}(G) = \text{STP}(G).$$

We then have that $\text{opt}_{SJ}(\cdot) \leq \text{opt}_{CM}(\cdot)$, and so the integrality gap of the CM formulation is a lower bound for the integrality gap of the SJ formulation on complete metric graphs. Moreover, the bound is not always tight. The same holds for the BCR formulation.

An interesting property of the CM formulation is connectedness. Constraints (1.13b) enforce that in an SJ solution, all the terminal nodes belong to the same connected component, but this is not guaranteed for non-terminal nodes. For the CM formulation instead, the following lemma holds.

Lemma 3. *The support graph of any feasible point of P_{CM} is connected.*

Proof. It suffices to prove that no connected components without terminals exist since every terminal belongs to the same connected component because of Constraint (1.15b). Let $x \in P_{CM}$ and let $H \subset V$ be a connected component of x containing no terminals. Constraint (1.15e) implies

$$\sum_{i,j \in H} x_{ij} = \sum_{j \in H} \sum_{i \in H} x_{ij} \geq \sum_{j \in H} 2 \sum_{i \in H} x_{ji} = 2 \sum_{i,j \in H} x_{ji} = 2 \sum_{i,j \in H} x_{ij}.$$

The only possibility is that $\sum_{i,j \in H} x_{ij} = 0$, so no connected component without terminals can be part of a feasible solution for CM. \square

Note that Lemma 3 does not hold for the SJ formulation, as Example 1 shows.

Another interesting property of the CM formulation deals with constraint reduction. In this case, we can prove theoretical results on the number of edges in an integer CM solution and, consequently, on the number of Steiner nodes.

Lemma 4. *Let $x \in S_{CM}(n, t)$. Then, x verifies*

$$\sum_{i,j} x_{ij} \leq \min(n-1, 2t-3). \quad (1.18)$$

Proof. Given x , let $G_x = (V_x, E_x)$ denote the corresponding support graph, that is $V_x = \{i \in V : x(\delta^+(i)) + x(\delta^-(i)) > 0\}$ and $E_x = \{e = \{i, j\} \in E : x_{ij} + x_{ji} > 0\}$. We know that G_x is acyclic because of Constraint (1.15d) and we also know that G_x is connected because of Lemma 3, so G_x is a tree. Since $|V_x| \leq n$, we have that $\sum_{i,j} x_{ij} \leq n-1$. Now, we only need to prove that $\sum_{i,j} x_{ij} \leq 2t-3$. We have that

$$\begin{aligned} \sum_{i,j} x_{ij} &= \sum_j \sum_{i \neq j} x_{ij} = \sum_{j \in T} \sum_{i \neq j} x_{ij} + \sum_{j \in V \setminus T} \sum_{i \neq j} x_{ij} = \\ &= \sum_{i \neq r} x_{ir} + \sum_{j \in T \setminus \{r\}} \sum_{i \neq j} x_{ij} + \sum_{j \in V \setminus T} \sum_{i \neq j} x_{ij} \leq \\ &\leq 0 + (t-1) + \frac{1}{2} \sum_{j \in V \setminus T} \sum_{k \neq j} x_{jk}, \end{aligned}$$

where the last inequality holds because of Constraint (1.15c), Constraint (1.15d) combined with Constraint (1.15b), and Constraint (1.15e), respectively. Note that only the last one gives us inequality since the others hold with equality. We can now rewrite

$$\sum_{j \in V \setminus T} \sum_{k \neq j} x_{jk} = \sum_{i,j} x_{ij} - \sum_{j \in T} \sum_{k \neq j} x_{jk}.$$

Combining this with the previous equation, we get that

$$\sum_{i,j} x_{ij} \leq t-1 + \frac{1}{2} \sum_{i,j} x_{ij} - \frac{1}{2} \sum_{j \in T} \sum_{k \neq j} x_{jk}.$$

Rearranging the terms, we obtain

$$\frac{1}{2} \sum_{i,j} x_{ij} \leq t-1 - \frac{1}{2} \sum_{j \in T} \sum_{k \neq j} x_{jk}$$

and hence, multiplying by 2

$$\begin{aligned} \sum_{i,j} x_{ij} &\leq 2t - 2 - \sum_{j \in T} \sum_{k \neq j} x_{jk} = \\ &= 2t - 2 - \sum_{k \neq r} x_{rk} - \sum_{j \in T \setminus \{r\}} \sum_{k \neq j} x_{jk} \leq 2t - 2 - 1 - 0 = 2t - 3, \end{aligned}$$

where the last inequality holds because $\sum_{k \neq r} x_{rk} \geq 1$ by taking $W = V \setminus \{r\}$ in Constraint (1.15b), and because $x_{jk} \geq 0$, respectively. \square

Observation 3. Let $t \leq \frac{n}{2} + 1$ and so $\min(n - 1, 2t - 3) = 2t - 3$. Then, if we consider the CM, our solution is a tree with at most $2t - 3$ edges, so it has $2t - 3 + 1 = 2t - 2$ nodes, t of which are terminals, leaving us with $t - 2$ Steiner vertices. Thus, it suffices to write Constraints (1.15b) only for

$$W = W_1 \sqcup W_2, \quad W_1 \subset T \setminus r, \quad |W_1| \geq 1, \quad W_2 \subset V \setminus T, \quad |W_2| \leq t - 2, \quad (1.19)$$

instead of writing it for any $W = W_1 \sqcup W_2, W_2 \subset V \setminus T$. For instance, in the case $(n, t) = (20, 5)$ we go from $(2^4 - 1) \times 2^{15} = 491520$ possible choices of W to just $(2^4 - 1) \times \sum_{i=0}^3 \binom{15}{i} = 8640$, which is around 1.8% of the total.

After discussing the properties that make the CM formulation interesting by itself, we now focus on commenting on its advantages in deducing information on the lower bounds of the BCR.

First, we discuss why studying the complete metric case is not restrictive. In particular, we use the *metric closure* of a graph, defined below.

Definition 1 (Metric Closure of a Graph). Let $G = (V, E)$ be an edge-weighted connected graph. We define the *metric closure* of G the complete metric graph $\bar{G} = (V, \bar{E})$ such that the weight of the edge $\{i, j\}$ in \bar{G} is equal to the value of the shortest paths from i to j in G .

We now link the integrality gap of the BCR formulation of a graph to the corresponding integrality gap of its metric closure.

Lemma 5. Let $G = (V, E)$, $T \subset V$ be a Steiner instance, and let \bar{G} be the Steiner instance corresponding to the metric closure of G . Then we have that

$$STP(G) = STP(\bar{G}), \quad opt_{BCR}(G) = opt_{BCR}(\bar{G}). \quad (1.20)$$

Proof. Let x be an integer feasible solution for G for the BCR formulation. Then, it is also a feasible solution for \bar{G} , and because of the definition of metric closure, it is a feasible solution with a smaller cost. We have then that $STP(\bar{G}) \leq STP(G)$. Let now \bar{x} be a feasible solution for \bar{G} for the BCR formulation. Reasoning in a non-oriented way, if we take every edge of \bar{x} and substitute it with the corresponding shortest path in G , we obtain a subgraph of G that can be oriented as a feasible solution x of G , with a smaller cost. The cost is (non-strictly) smaller because we may take the same

edge in different shortest paths. We then have that $\text{STP}(\tilde{G}) \geq \text{STP}(G)$ and so $\text{STP}(\tilde{G}) = \text{STP}(G)$.

For the same reasoning, we have that $\text{opt}_{BCR}(G) = \text{opt}_{BCR}(\tilde{G})$, with the exception that, when substituting an edge of \tilde{G} with the corresponding shortest path in G , since we are dealing with fractional solutions, if we have to take the same edge multiple times because it appears in multiple shortest paths, we have to take the minimum between 1 and the sum of all the values with which that edge appears. This choice preserves feasibility and does not produce a solution with a larger cost. \square

From this lemma, it follows that the integrality gap calculated with respect to the CM formulation only on metric graphs is a lower bound for the integrality gap of the BCR formulation across all graphs.

1.3.2 The gap problem for the CM formulation

With this in mind, one can proceed as in Section 1.2 and define a Gap problem for the CM formulation. Given \bar{x} vertex of $P_{\text{CM}}(n, t)$, we define its gap as the linear problem of finding the cost vector that maximizes the integrality gap of a vertex \bar{x} , among those for which \bar{x} is optimal. As the structure is nontrivial, we first write the constraints of the dual formulation.

Let $W(i, j)$ defined as

$$W(i, j) := \{W \mid W \subset V \setminus \{r\}, W \cap T \neq \emptyset, (i, j) \in \delta^-(W)\}$$

Then, using the theory of LP duality, we can write the following.

$$y_{ri} + v_i + \sum_{w \in W(r, i)} z_w \leq c_{ri}, \quad i \in T \setminus \{r\}, \quad (1.21a)$$

$$y_{rj} + v_j + 2u_j + \sum_{w \in W(r, j)} z_w \leq c_{rj}, \quad j \in S, \quad (1.21b)$$

$$y_{ij} + v_j + \sum_{w \in W(i, j)} z_w \leq c_{ij}, \quad i, j \in T \setminus \{r\}, \quad (1.21c)$$

$$y_{ij} + v_j + 2u_j + \sum_{w \in W(i, j)} z_w \leq c_{ij}, \quad i \in T \setminus \{r\}, j \in S, \quad (1.21d)$$

$$q + y_{ir} \leq c_{ir}, \quad \forall i \in T \setminus \{r\}, \quad (1.21e)$$

$$q + y_{jr} \leq c_{jr}, \quad \forall j \in S, \quad (1.21f)$$

$$y_{ji} + v_i - u_j + \sum_{w \in W(j, i)} z_w \leq c_{ji}, \quad i \in T \setminus \{r\}, j \in S, \quad (1.21g)$$

$$y_{ij} + v_j + 2u_j - u_i + \sum_{w \in W(i, j)} z_w \leq c_{ij}, \quad i, j \in S, \quad (1.21h)$$

$$q \text{ free}, z \geq 0, v_j, u_j, y_{ij} \leq 0. \quad (1.21i)$$

Note that we can merge some constraints, in particular, (1.21a) and (1.21c) are the same constraint where $i \in T$ and $j \in T \setminus \{r\}$. The same holds for (1.21b) and (1.21d) if $i \in T$ and $j \in S$. Lastly, we can drop constraint (1.21e) and (1.21f) as variable q is free, and it only appears in these constraints. Note that, referring to the primal formulation, this would imply deleting the variables x_{ir} , $i \in V \setminus \{r\}$. Hence, the dual polytope can be rewritten as

$$y_{ij} + v_j + \sum_{w \in W(ij)} z_w \leq c_{ij}, \quad i \in T, j \in T \setminus \{r\}, \quad (1.22a)$$

$$y_{ij} + v_j + 2u_j + \sum_{w \in W(i,j)} z_w \leq c_{ij}, \quad i \in T, j \in S, \quad (1.22b)$$

$$y_{ji} + v_i - u_j + \sum_{w \in W(j,i)} z_w \leq c_{ji}, \quad i \in T \setminus \{r\}, j \in S, \quad (1.22c)$$

$$y_{ij} + v_j + 2u_j - u_i + \sum_{w \in W(i,j)} z_w \leq c_{ij}, \quad i, j \in S, \quad (1.22d)$$

$$z \geq 0, v_j, u_j, y_{ij} \leq 0. \quad (1.22e)$$

Given a vertex $x \in P_{\text{CM}}(n, t)$, we introduce variables c_{ij} , $\{i, j\} \in E$ that satisfy the triangle inequality and non-negativity constraints. Adding them to the slackness compatibility conditions we obtain the following linear program with exponentially many variables and constraints:

$$\min \sum_{\{i,j\} \in E} c_e(\bar{x}_{ij} + \bar{x}_{ji}) \quad (1.23a)$$

$$\text{s.t. } c_{ij} \leq c_{ik} + c_{jk} \quad \forall \{i, j\}, \{i, k\}, \{j, k\} \in E, \quad (1.23b)$$

$$z_W = 0, \quad \forall W \in \mathcal{W}, \quad (1.23c)$$

$$v_j = 0, \quad \forall j \in V \setminus \{r\}, \bar{x}(\delta^-(j)) < 1, \quad (1.23d)$$

$$u_j = 0, \quad \forall j \in S, 2x(\delta^-(v)) < \bar{x}(\delta^+(v)), \quad (1.23e)$$

$$y_{ij} = 0, \quad \forall (i, j) \in A, \bar{x}_{ij} < 1, \quad (1.23f)$$

$$c_{ij} \geq 0, \quad \forall \{i, j\} \in E, \quad (1.23g)$$

$$\sum_{\{i,j\} \in E} c_e(\bar{z}_{ij} + \bar{z}_{ji}) \geq 1, \quad \forall \bar{z} \in S_{\text{CM}}(n, t) \quad (1.23h)$$

$$\forall \bar{x}_{ij} > 0 :$$

$$y_{ij} + v_j + \sum_{w \in W(ij)} z_w - c_{ij} = 0, \quad \forall i \in T, j \in T \setminus \{r\}, \quad (1.23i)$$

$$y_{ij} + v_j + 2u_j + \sum_{w \in W(i,j)} z_w - c_{ij} = 0, \quad \forall i \in T, j \in S, \quad (1.23j)$$

$$y_{ji} + v_i - u_j + \sum_{w \in W(j,i)} z_w - c_{ji} = 0, \quad \forall i \in T \setminus \{r\}, j \in S, \quad (1.23k)$$

$$y_{ij} + v_j + 2u_j - u_i + \sum_{w \in W(i,j)} z_w - c_{ij} = 0, \quad \forall i, j \in S. \quad (1.231)$$

1.4 Heuristic enumeration of nontrivial vertices

In this section, we present the theoretical results and algorithms used to enumerate vertices of the polytope $P_{CM}(n, t)$. We first introduce results linking polytopes of different dimensions, and then, relying upon these and other structural results, we present two different algorithms for vertices enumeration.

1.4.1 Avoiding redundancy

First, let us define a particular class of vertices that will be of interest for our results.

Definition 2. Let x be a vertex of $P_{CM}(n, t)$. We will call x a *spanning vertex* if all of the nodes are part of the solution x , that is, $x(\delta^-(i)) + x(\delta^+(i)) > 0$ for all $i \in V$.

Note that Lemma 3 implies that every spanning vertex is connected. In an STP, Steiner nodes may or may not be part of an optimal solution. This holds for vertices of $P_{CM}(n, t)$, both integer and non-integer, that is, not all of the vertices are spanning vertices. Hence, we can consider whether a non-spanning vertex of $P_{CM}(n, t)$ can be seen as a spanning vertex of a polytope of a smaller dimension, and vice versa, that is, if a spanning vertex of $P_{CM}(n, t)$ can be seen as a vertex of a polytope of a larger dimension. The following results link vertices of $P_{CM}(n+1, t)$ with vertices of $P_{CM}(n, t)$ and vice versa. These results will be used in the enumeration of vertices to reduce the dimension of our search space by avoiding redundancy.

Lemma 6. Let $x \in \mathbb{R}^{(n-1) \times n}$. Define $y \in \mathbb{R}^{n \times (n+1)}$ as

$$y_{ij} = \begin{cases} x_{ij}, & 1 \leq i, j < n+1, \\ 0, & \text{otherwise.} \end{cases} \quad (1.24)$$

Then, $x \in P_{CM}(n, t)$ if and only if $y \in P_{CM}(n+1, t)$.

Proof. Let $x \in P_{CM}(n, t)$. Note that y satisfies the domain constraints. Regarding Constraint (1.15b), we distinguish two cases. Let W be a set as described in (1.15b) for y . (a) If $n+1 \in W$, going from x to y adds the variables $x_{i, n+1}$ which are all zero so since x satisfies the constraint y satisfies it too. (b) If $n+1 \notin W$, going from x to y adds the variables $x_{n+1, j}$ which are all zero so since x satisfies the constraint y satisfies it too. Constraints (1.15c)–(1.15d) are satisfied by y since x satisfies them, and we are only adding variables that take the value zero. Regarding Constraint (1.15e), if $j = n+1$, the constraint holds trivially since all the variables are zero. If $j \neq n+1$, going from x to

y adds the variables $x_{i,n+1}, x_{n+1,j}$ which are all zero, so since x satisfies the constraint, y also satisfies it.

Let $y \in P_{CM}(n+1, t)$ of the form (1.24). Note that x satisfies the domain constraints. Let W be a set as described in (1.15b) for x . Let $\hat{W} := W \cup \{n+1\}$. \hat{W} is a set for which y satisfies the correspondent constraint (1.15b). In the \hat{W} constraint, the only variables that appear are the ones appearing in the W constraint plus the variables $x_{i,n+1}$ which are all zero. Since the \hat{W} constraint is satisfied by y , the W constraint is satisfied by x . Constraints (1.15c)–(1.15d) are clearly satisfied by x since y satisfies them. Regarding Constraint (1.15e), passing from y to x removes the variables $x_{i,n+1}, x_{n+1,j}$ which are all zero, so since y satisfies the constraint, x also satisfies it. \square

The following lemmas show how to identify vertices of $P_{CM}(n+1, t)$ with the ones of $P_{CM}(n, t)$ and vice versa.

Lemma 7. *Let x be a vertex of $P_{CM}(n, t)$. Then*

$$y_{ij} = \begin{cases} x_{ij}, & \text{if } i, j \neq n+1, \\ 0, & \text{otherwise} \end{cases} \quad (1.25)$$

is a vertex of $P_{CM}(n+1, t)$.

Proof. The idea of the proof is to show by contradiction that if y is not a vertex, that x cannot be a vertex as well. In detail, we have that $y \in P_{CM}(n+1, t)$ because of Lemma 6. Let $P_{CM}(n+1, t)_0$ be the subpolytope of $P_{CM}(n+1, t)$ defined as

$$P_{CM}(n+1, t)_0 := \{z \in P_{CM}(n+1, t) : z_{i,n+1} = z_{n+1,j} = 0, 1 \leq i, j \leq n\}.$$

Let

$$\begin{aligned} \pi : P_{CM}(n+1, t)_0 &\rightarrow P_{CM}(n, t) \\ (z_{ij})_{i,j} &\mapsto (z_{ij})_{i,j \neq n+1} \end{aligned}$$

be the projection considering the first n nodes. Note that $\pi(y) = x$ and that π is an injective map. Note also that $\text{Im}(\pi) \subset P_{CM}(n, t)$ because of Lemma 6. By contradiction, suppose that there exist $a, b \in P_{CM}(n+1, t)$ such that $a \neq b$, $y = \frac{1}{2}a + \frac{1}{2}b$. We have that

$$y_{i,n+1} = y_{n+1,j} = 0 = \frac{1}{2}(a_{i,n+1} + b_{i,n+1}) = \frac{1}{2}(a_{n+1,j} + b_{n+1,j}).$$

Since $a, b \in P_{CM}(n+1, t)$, we have that $a_{i,n+1}, b_{i,n+1}, a_{n+1,j}, b_{n+1,j} \geq 0$ and so $a_{i,n+1}, b_{i,n+1}, a_{n+1,j}, b_{n+1,j} = 0$. Thus, $a, b \in P_{CM}(n+1, t)_0$ and we can define $c := \pi(a)$, and $d := \pi(b)$, and we have that $c, d \in P_{CM}(n, t)$, $c \neq d$, $x = \frac{1}{2}c + \frac{1}{2}d$, a contradiction. \square

Lemma 8. *Let $t < n$ and let y be a vertex of $P_{CM}(n, t)$ of the form*

$$y_{ij} = \begin{cases} x_{ij}, & \text{if } i \neq n \neq j, \\ 0, & \text{else,} \end{cases} \quad (1.26)$$

for $n \in V \setminus T$. Then x is a vertex of $P_{CM}(n-1, t)$.

Proof. We have that $x \in P_{CM}(n-1, t)$ because of Lemma 6. Let

$$\begin{aligned} i: P_{CM}(n-1, t) &\hookrightarrow P_{CM}(n, t) \\ (z_{i,j})_{i \neq n \neq j} &\mapsto ((z_{i,j})_{i \neq n \neq j}, 0, \dots, 0) \end{aligned}$$

be the trivial immersion and note that $i(x) = y$. Note also that $\text{Im}(i) \subset P_{CM}(n, t)$ because of Lemma 6. By contradiction, suppose there exist $c, d \in P_{CM}(n-1, t)$ such that $c \neq d$, $x = \frac{1}{2}c + \frac{1}{2}d$. If we define $a := i(c)$, and $b := i(d)$, we have that $a, b \in P_{CM}(n, t)$, $a \neq b$, $y = \frac{1}{2}a + \frac{1}{2}b$, and so we have a contradiction. \square

Note that the result above still holds if we replace the node n with any node $k \in V \setminus T$.

Observation 4. Let π and i be the maps introduced in the proofs of Lemma 7 and Lemma 8, respectively. Note that π is an injective map and $i(P_{CM}(n, t)) \subset P_{CM}(n+1, t)_0$, thus we have that π is also a surjective map and so it is bijective. Moreover, π is linear and sends vertices in vertices. In particular $P_{CM}(n+1, t)_0 \cong P_{CM}(n, t)$, where the isomorphism is given by the map π . Note that π is a surjective map because given an element $x \in P_{CM}(n, t)$, we have that $\pi(i(x)) = x$, and we can map $i(x)$ through π because $i(P_{CM}(n, t)) \subset P_{CM}(n+1, t)_0$. This implies that, in the aim of evaluating vertices of our polytopes, it is sufficient to evaluate vertices of $P_{CM}(n, t)$ to get all of the vertices of $P_{CM}(m, t)$, for every $m = t, t+1, \dots, n$. Alternatively, we can evaluate only the spanning vertices of $P_{CM}(n, t)$ for every n, t , since every non-spanning vertex can be seen as a spanning vertex of a polytope of a smaller dimension, applying the lemmas above iteratively. Note that we are only interested in non-isomorphic vertices because isomorphic vertices have the same integrality gap. Note also that the results presented above hold for the BCR and the SJ formulations. The proof can be done in almost the same way.

Observation 5. As we have seen, the trivial way to go from a vertex of $P_{CM}(n+1, t)$ to a vertex of $P_{CM}(n, t)$ is removing zeros, and the trivial way to go from a vertex of $P_{CM}(n, t)$ to a vertex of $P_{CM}(n+1, t)$ is adding zeros. As one would expect, the trivial way to go from a vertex of $P_{CM}(n+1, t+1)$ to a vertex of $P_{CM}(n, t)$ and vice versa is the “dual” procedure of the previous one, that is, adding or removing one 1. Note that this can be done in different ways. More precisely, the following procedures start with a vertex of $P_{CM}(n, t)$ and return a vertex of $P_{CM}(n+1, t+1)$.

- (a) Add an edge of weight 1 between a node v of indegree 1 and the new added terminal, see for example Figure 1.2b \rightarrow Figure 1.3a and Figure 1.2a \rightarrow Figure 1.3b.
- (b) Same as (a), but substituting the outflow of v with the outflow of the newly added terminal, see for example Figure 1.2b \rightarrow Figure 1.3c.
- (c) Add an edge of weight 1 between the newly added terminal and the root, then swap the role of these two nodes, see Figure 1.2a \rightarrow Figure 1.3d.

Reversing these procedures, when possible, allows us to go from a $P_{CM}(n+1, t+1)$ to a vertex of $P_{CM}(n, t)$. The proofs are similar to the ones presented above. For all of the procedures above, it is clear that the generated vertices are not isomorphic to the ones we start from.

Note that the above procedures do not change the integrality gap, as shown by the lemma below.

Lemma 9. *Let $\epsilon \geq 0$, $c \in \mathbb{R}_{\geq \epsilon}^n$ a cost vector of a minimization ILP instance, and let $x \in \{0, 1\}^n$ be the variables of the LP. Denote by \bar{x}, \hat{x} an optimal integer solution and an optimal solution of the LP relaxation, respectively. Suppose an index k exists such that $\hat{x}_k = 1$. Then, the instance \tilde{c} defined as*

$$\tilde{c}_j = \begin{cases} c_j & j \neq k, \\ \epsilon & j = k, \end{cases} \quad (1.27)$$

has a greater or equal integrality gap than the instance c . Moreover, \hat{x} is an optimum for the LP relaxation of the instance \tilde{c} .

Proof. Let $P \subset [0, 1]^n$ be the polytope defined by the LP relaxation. For the second part, suppose by contradiction that there exists $y \in P$ such that $\tilde{c}^\top y < \tilde{c}^\top \hat{x}$. This can be rewritten as

$$\tilde{c}^\top y = \sum_{i=1}^n \tilde{c}_i y_i = \epsilon y_k + \sum_{i \neq k}^n c_i y_i < \epsilon \hat{x}_k + \sum_{i \neq k}^n c_i \hat{x}_i.$$

Thus,

$$\begin{aligned} c^\top y &= c_k y_k + \sum_{i \neq k}^n c_i y_i < c_k y_k + \epsilon(\hat{x}_k - y_k) + \sum_{i \neq k}^n c_i \hat{x}_i = \\ &= \epsilon(\hat{x}_k - y_k) + c_k y_k - c_k \hat{x}_k + \sum_{i=1}^n c_i \hat{x}_i = \\ &= (\epsilon - c_k)(\hat{x}_k - y_k) + c^\top \hat{x} = (\epsilon - c_k)(1 - y_k) + c^\top \hat{x} \leq c^\top \hat{x}, \end{aligned}$$

which contradicts the optimality of \hat{x} . For the first part, we have

$$\text{IG}(\tilde{c}) = \frac{\text{ILP}(\tilde{c})}{\text{LP}(\tilde{c})} = \frac{\text{ILP}(\tilde{c})}{\tilde{c}^\top \hat{x}} = \frac{\text{ILP}(\tilde{c})}{\text{LP}(c) - c_k + \epsilon}.$$

We have that $\text{ILP}(\tilde{c}) \leq \tilde{c}^\top \bar{x} \leq c^\top \bar{x} = \text{ILP}(c)$. We now want to prove that $\text{ILP}(\tilde{c}) \geq \text{ILP}(c) - c_k + \epsilon$. Suppose by contradiction that there exists $\bar{y} \in P$ integer such that $\tilde{c}^\top \bar{y} < c^\top \bar{x} - c_k + \epsilon$. But then

$$\begin{aligned} c^\top \bar{y} &= c_k \bar{y}_k + \sum_{i \neq k}^n c_i \bar{y}_i = (c_k - \epsilon) \bar{y}_k + \tilde{c}^\top \bar{y} \\ &< (c_k - \epsilon) \bar{y}_k + c^\top \bar{x} - c_k + \epsilon = (\bar{y}_k - 1)(c_k - \epsilon) + c^\top \bar{x} \leq c^\top \bar{x}, \end{aligned}$$

which contradicts the optimality of \bar{x} . We then have that $\text{ILP}(c) \geq \text{ILP}(\tilde{c}) \geq \text{ILP}(c) - c_k + \epsilon$, which implies

$$\frac{\text{ILP}(c)}{\text{LP}(c) - c_k + \epsilon} \geq \text{IG}(\tilde{c}) \geq \frac{\text{ILP}(c) - c_k + \epsilon}{\text{LP}(c) - c_k + \epsilon} \geq \frac{\text{ILP}(c)}{\text{LP}(c)} = \text{IG}(c)$$

where the last inequality holds because $\text{ILP}(c) \geq \text{LP}(c) \geq 0$, $\text{LP}(c) - c_k + \epsilon \geq 0$, $c_k \geq \epsilon$. \square

Observation 6. Let \hat{x} be a fractional vertex of $P_{CM}(n, t)$ such that $\hat{x}_{ij} = 1$. Because of Lemma 9, the maximum integrality gap is reached by minimizing the value of c_{ij} , making node i and node j collapse onto each other. This can be done by choosing a sequence of values of ϵ such that $\epsilon \rightarrow 0$. The study of the gap of this vertex is then equivalent to the study of the gap of a vertex of a smaller dimension. Note that, if we restrict the study to the metric case, even if (1.27) might not define a metric cost, the result still holds because of Lemma 5.

1.4.2 Two heuristics procedure for vertices enumeration

In the following, we state more properties of the CM formulation that permit the design of two different heuristic procedures, in particular, one general search and one dedicated to a specific class of vertices. We are only interested in spanning vertices (Observation 4).

The $\{1, 2\}$ -costs heuristic

The first procedure is based on the observation that, when looking for integer solutions of the CM formulation, it is enough to study only metric graphs with edge weights in the set $\{1, 2\}$.

Theorem 1. *Let $x \in S_{CM}(n, t)$. Then, x is an integer optimal solution for the CM formulation with the metric cost $c_{ij} = 2 - (x_{ij} + x_{ji}) \in \{1, 2\}$.*

Proof. Consider x and the STP instance given by the vector c defined in the statement. We want to prove that x is optimal. Let x' be the integer optimal solution for c and let s, s' be the number of Steiner nodes of x and x' , respectively. Let us write $x_e = x_{ij} + x_{ji}$ and the same for x' . We will divide the proof into two cases. First, we will prove (i) that if $s' \geq s$, then

necessarily $s' = s$ and $x = x'$. Then, we will prove (ii) that if $s' < s$ we get a contradiction.

- (i) Since the optimal solution is a tree with t terminals and s and s' Steiner node, respectively, we have that

$$\sum_{e \in E} x_e = t + s - 1, \quad \sum_{e \in E} x'_e = t + s' - 1.$$

Note that the definition of c implies that the cost is equal to 1 on the edges of the support graph of the solution and equal to 2 otherwise. Because of the definition of c , we have that

$$\sum_{e \in E} c_e x_e = t + s - 1.$$

Now let $I_0 = \{e : x_e = 0, x'_e = 1\}$, $I_1 = \{e : x_e = 1, x'_e = 0\}$, $I = \{e : x_e = x'_e\}$. We then have that

$$\begin{aligned} \sum_{e \in E} c_e x'_e &= \sum_{e \in I_0} c_e x'_e + \sum_{e \in I_1} c_e x'_e + \sum_{e \in I} c_e x'_e = 2 \times |I_0| + |I| \geq |I_0| + |I| = \\ &= t + s' - 1 \geq t + s - 1 = \sum_{e \in E} c_e x_e, \end{aligned}$$

and they are equal if and only if $s' = s$ and $I_0 = \emptyset$, and since these two conditions imply $I_1 = \emptyset$, we have that $x = x'$.

- (ii) Let $\mathcal{S}(x)$ and $\mathcal{S}(x')$ be the set of Steiner nodes of the solution x and x' , respectively. Let $\mathcal{S} = \{s_1, \dots, s_k\} = \mathcal{S}(x) \setminus \mathcal{S}(x')$ and let z be the number of edges of the form $s_i s_j$. Note that $\mathcal{S} \neq \emptyset$, otherwise we would have $s' \geq s$. Now, x' is a tree with $s' + t$. Thanks to the hypothesis $s' < s$, we have $s' = s - k$, and hence x' is a tree with $s + t - k$ nodes, so $s + t - k - 1$ edges. On the other side, x has $s + t - 1$ edges, all of them of cost 1, while all of the other edges have cost 2. We now have to evaluate how many edges of cost 2 x' must have, given that it does not contain any node of the set \mathcal{S} . We have that c contains exactly $s + t - 1$ edges of cost 1, and the number of those edges that contain a node of \mathcal{S} is $\left(\sum_{i=1}^k \deg(s_i)\right) - z$.

Since we said that x' must contain $s + t - k - 1$ edges, its cost is $E_1 + 2 \times E_2$, where E_1 is the number of edges of cost 1 and E_2 is the number of edges of cost 2, and we have that

$$E_1 \leq s + t - 1 - \left(\left(\sum_{i=1}^k \deg(s_i) \right) - z \right), \quad E_2 = s + t - k - 1 - E_1,$$

and the minimum of $E_1 + 2 \times E_2$ is attained when E_1 is exactly equal to the rhs. The difference between the cost of x' and the cost of x , which is exactly $s + t - 1$, is then at least

$$\begin{aligned} & 2 \times \left(\left(\sum_{i=1}^k \deg(s_i) \right) - k - z \right) - \left(\left(\sum_{i=1}^k \deg(s_i) \right) - z \right) = \\ & = \left(\sum_{i=1}^k \deg(s_i) \right) - 2k - z \geq 3k - 2k - z \geq k - z \geq 1, \end{aligned}$$

So, x' is not optimal, and we have a contradiction. Note that $\deg(s_i) \geq 3$ because of Constraints (1.15d) and (1.15e), and $k - z \geq 1$ because the support graph associated to \mathcal{S} as a subgraph of x is a forest since it is a subgraph of a CM solution, which is a tree. □

Observation 7. Note that the generalization of Theorem 1 does not hold in general for the non-integer case, that is, if x is a non-integer point of $P_{CM}(n, t)$, then x is not necessarily an optimal solution for the CM formulation with the metric cost

$$c_{ij} = 2 - \mathbb{1}_{E_x}(\{i, j\}), \quad E_x = \{e = \{i, j\} \in E : x_{ij} + x_{ji} > 0\}, \quad (1.28)$$

see, for example, the vertex depicted in Figure 1.4d. In this case, with the cost assignation (1.28), we have that the fractional vertex has a value of $11/2$ (multiply the number of edges by $1/2$), while the optimal value of the CM formulation for this instance is 5. Thus, the vertex shown in Figure 1.4d cannot be optimal for this instance. It still holds that the vertex mentioned above is an optimum of a metric graph where every edge weight is in the set $\{1, 2\}$, namely setting the cost as in (1.28) but changing the cost of the two edges outflowing the root, setting them to 2 instead of 1. Note that in this case, the subgraph linked to edges with cost 1 is not connected, as the root represents a connected component.

The observation above, together with Theorem 1, lead us to formulate a heuristic search based on the generation of metric graphs with edge weights in the set $\{1, 2\}$ and then solve the STP on those instances. The detailed procedure called $OTC(n, t)$ as in One-Two-Costs is described in Algorithm 1.

Note that for computational reasons, we restricted our search to the generation of only connected graphs, and so to graphs with costs $\{1, 2\}$ in which the subgraph regarding the edges of cost 1 spans all the nodes and is connected. We know this is a strong restriction, making the procedure unable to find some vertices, see Observation 7. Note also that we restrict our search to graphs $G = (V, E)$ with $n \leq |E| \leq n \cdot t - t^2$: the lower bound is given by the fact that we are only interested in non-integer vertices, and the upper bound was derived after a first set of computational experiments.

Algorithm 1 $\{1, 2\}$ -costs vertices heuristic

```

procedure OTC( $n, t$ )
 $\mathbb{G} = \{G = (V, E) \mid G \text{ connected, } |V| = n, n \leq |E| \leq n \cdot t - t^2\}$ 
 $\mathbb{T} = \{T \mid T \subset \{1, \dots, n\}, |T| = t\}$ 
for  $G \in \mathbb{G}$  do
  for  $T \in \mathbb{T}$  do
    for  $r \in T$  do
       $G_{T,r}$  = node-colored graph with
        ·  $G$  as its support graph
        ·  $r$  colored as root
        ·  $i$  colored as terminal  $\forall i \in T \setminus \{r\}$ 
        ·  $j$  colored as steiner  $\forall i \notin T$ 
      if  $G_{T,r} \not\cong H \forall H \in \mathfrak{G}$  then
        add  $G_{T,r}$  to  $\mathfrak{G}$ 
 $\mathcal{V} = \emptyset$ 
for  $G_{T,r} \in \mathfrak{G}$  do
  obtain STP instance  $(G, T, r)$  from  $G_{T,r}$  with  $c_{ij} = \begin{cases} 1, & \{i, j\} \in G_{T,r} \\ 2, & \text{otherwise} \end{cases}$ 
  solve the relaxation of the CM formulation for the instance above
  if a solution  $x$  is found and it is a non-integer vertex of  $P_{CM}(n, t)$  then
    add  $x$  to  $\mathcal{V}$ 
return  $\mathcal{V}$ 

```

Pure half-integer vertices

Guided by the literature, we narrow our research to a specific class of vertices, which is conjectured to exhibit the maximum integrality gap in other NP-Hard problems (see, e.g., the Symmetric Traveling Salesman Problem [14, 31] and the Asymmetric Traveling Salesman Problem [50]). In particular, we restrict our attention to vertices having their values in $\{0, \frac{1}{2}, 1\}$. Given a non-integer vertex x of $P_{CM}(n, t)$, we say that x is *half-integer* (HI) if $x_{ij} \in \{0, 1/2, 1\} \forall i, j \in V$ and we say that x is *pure half-integer* (PHI) if $x_{ij} \in \{0, 1/2\} \forall i, j \in V$. In the following section, we state and prove some properties of PHI vertices. The choice of focusing only on PHI vertices instead of HI vertices is motivated by Lemma 9 and Observation 6.

Lemma 10. *Let x be a pure half-integer solution of $P_{CM}(n, t)$ which is also a vertex of $P_{BCR}(n, t)$ optimum for a metric cost. Then we have that $x_{ij} > 0 \Rightarrow x_{ji} = 0$.*

Proof. Since x is pure half-integer, we have $x_{ij} = 1/2$. Suppose by contradiction that $x_{ji} \neq 0$, and so by the same reasoning, $x_{ji} = 1/2$. Because of Lemma 3, we have that the set $\{i, j\}$ is not a connected component of x , namely, is not an isolated 2-cycle, and neither of the two nodes can be the

root, as the root has inflow equal to 0 because of Constraint (1.15c). Thus, there must exist a path from the root to the two nodes, and so there must exist an active arc going from a third node to one of the two nodes we are considering. Without loss of generality, let $x_{ki} > 0$, that implies $x_{ki} = 1/2$. Suppose $x_{ik} = 0$ since, otherwise, we can do the same reasoning for nodes $\{i, j, k\}$ and repeat it until we return to the root, which has no inflow. Now, we distinguish between two cases.

- (a) No other inflow is present in j , that is, $x_{aj} = 0 \forall a \neq j$. Note that this implies that j is not a terminal since it has an inflow of $1/2$. Then x is not optimum. Consider x' that is equal to x on all the arcs but the arc (j, i) , and set $x'_{ji} = 0$. For any nonnegative c , $c^\top x' < c^\top x$. Note that x' is feasible for the BCR. Constraint (1.1b) is clearly satisfied. Constraint (1.1c) could not be verified by x' only for a set W for which $I \in W$, $j \notin W$, because then it appears the only variables that differ from x . Let us take one of these sets and define $\bar{W} = W \cup \{j\}$. We can write

$$\begin{aligned}
\sum_{(a,b) \in \delta^-(W)} x'_{ab} &= \sum_{\substack{(a,b) \in \delta^-(W) \\ (a,b) \neq (j,i)}} x'_{ab} + x'_{ji} = \\
&= \sum_{(a,b) \in \delta^-(\bar{W})} x'_{ab} - \sum_{a \in V \setminus W} x'_{aj} + x'_{ji} = \\
&= \sum_{(a,b) \in \delta^-(\bar{W})} x_{ab} - \sum_{a \in V \setminus W} x_{aj} + x'_{ji} = \\
&= \sum_{(a,b) \in \delta^-(\bar{W})} x_{ab} + 0 + x'_{ji} \geq 1 + 0 = 1,
\end{aligned}$$

where the inequality holds because x is feasible and W is a valid set. So we have that x' is feasible even for the constraints regarding the sets W for which $i \in W$, $j \notin W$ and so it is feasible for the BCR. If x' is feasible for the CM, the proof is concluded. If x' is not feasible for the CM formulation, it is because of Constraint (1.15e) because x' satisfied all of the other constraints since x is feasible for CM. Regarding Constraint (1.15e), if x' is not feasible for the CM anymore, it is because the outdegree of j in x was exactly two, namely $x_{ji} = \frac{1}{2}$ and there exist d such that $x_{jd} = \frac{1}{2}$. Hence, we can build x'' from x' by removing arc x'_{ij} and x'_{jd} from x' and by adding the arc x''_{id} , avoiding the detour in j . This solution is feasible for the CM, and it holds $c^\top x'' \leq c^\top x'$, for the non-negativity and the triangle inequality. Hence, $c^\top x'' < c^\top x$, from the relation between x and x' already proved. Hence, we can conclude that if the only inflow of the (Steiner) node j is x_{ij} , x is neither optimal for the CM nor for the BCR.

- (b) The total inflow of j is 1, and so there exists l such that $x_{lj} = 1/2$. Suppose $x_{jl} = 0$ and suppose that both k and l have an inflow of 1. This will ensure

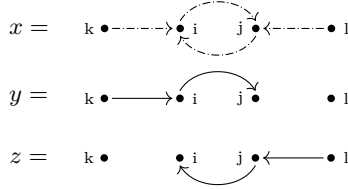
the feasibility of the two points we are about to construct. Then x is not a vertex of $P_{BCR}(n, t)$, because by setting

$$y_{ab} = \begin{cases} 0, & \text{if } a = l, b = j, \text{ or } a = j, b = i, \\ 1, & \text{if } a = i, b = j, \text{ or } a = k, b = i, \\ x_{ab}, & \text{else,} \end{cases}$$

$$z_{ab} = \begin{cases} 1, & \text{if } a = l, b = j, \text{ or } a = j, b = i, \\ 0, & \text{if } a = i, b = j, \text{ or } a = k, b = i, \\ x_{ab}, & \text{else,} \end{cases}$$

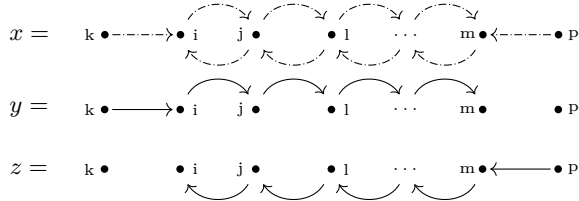
we have $y \neq z$, $x = \frac{1}{2}y + \frac{1}{2}z$, and $y, z \in P_{BCR}(n, t)$ by an argument similar from the one above.

Visually, we can represent the three points as the following



where we draw only the interesting arcs. Note that dashed arcs represent a value of $1/2$ while full arcs represent a value of 1 . If $x_{jl} \neq 0$, that is, $x_{jl} = 1/2$, then we can go backward until we find one node m such that there exists p for which $x_{pm} = 1/2$, $x_{mp} = 0$, and such a p exists because we can go back to the root with the same reasoning as above. Suppose that both k and p have an inflow of 1 . We now do the same reasoning with y and z but consider the whole paths from p to i and from k to m instead of the paths from l to i and from k to j .

Visually, we can represent the three points as the following



where we draw only the interesting arcs. Note that dashed arcs represent a value of $1/2$ while full arcs represent a value of 1 . If k or p do not have an inflow of 1 , we can just go backward until we find a point with this

property. If we do not find it, we go backward to the root. At this point, we can do the same reasoning with the paths as we did above. \square

We now focus on a particular type of PHI vertices, namely spanning vertices such that every Steiner node has indegree exactly one. We conjecture that every PHI spanning vertex has this property based on the following reasoning: first of all, because of Lemma 10, there are no loops of length 2, and so every edge can be oriented in only one way. Suppose there exists a Steiner node k such that $\text{indeg}(k) > 1$, and since the maximum inflow is 1 because of Constraint (1.15d) and we are dealing with pure half-integer solutions, we have that $\text{indeg}(k) = 2$. Then, regarding the MCF formulation, there exist $T_1, T_2 \subset T$, $T_1, T_2 \neq \emptyset$, and $i, j \in V$ such that $f_{ik}^{t_1} = f_{jk}^{t_2} = 1/2$, $\forall t_1 \in T_1, t_2 \in T_2$. We conjecture that is always possible to construct $y, z \in P_{BCR}(n, t)$ such that $y \neq z$ and $x = \frac{1}{2}y + \frac{1}{2}z$, leading to a contradiction. In particular, y is derived by x by setting $f_{ik}^{t_1} = 1$, $f_{jk}^{t_2} = 0$, $\forall t_1 \in T_1, t_2 \in T_2$, and all the other variables are set accordingly to (1.16c), while z is derived by x by setting $f_{ik}^{t_1} = 0$, $f_{jk}^{t_2} = 1$, $\forall t_1 \in T_1, t_2 \in T_2$, and all the other variables are set accordingly to (1.16c).

We now derive some properties of these vertices that will be exploited in our heuristic search.

Lemma 11. *Let x be a pure half-integer solution of $P_{CM}(n, t)$, $t \geq 3$ that is also a vertex of $P_{BCR}(n, t)$ and an optimum for a metric cost. Let x be a spanning vertex such that every Steiner node has indegree 1. Then, it holds that*

- $|\{(i, j) \in A \mid x_{ij} > 0\}| = n + t - 2$,
- $3t - n - 4 \geq 0$.

Proof. For the first point, it suffices to count the incoming edges of each node. We have one incoming edge for each Steiner node and exactly two incoming edges for every terminal that is not the root since every terminal has an inflow exactly equal to one, and our edges have weights $1/2$. The total number of edges is $n - t + 2(t - 1) = n + t - 2$.

For the second point, because of Constraint (1.15b), we have that at least two edges exit from the root and at least two edges enter in every other terminal. Moreover, since in every Steiner node enters exactly one edge, at least two edges must come out. We then have that $2(n + t - 2) \geq 2t + 3(n - t)$ and so $3t - n - 4 \geq 0$. \square

The properties stated above represent the core of the heuristic we now present. We generate all of the non-isomorphic connected undirected graphs such that every node is of degree at least 2 and with exactly $n + t - 2$ edges with the command `geng` of nauty [97]. For every generated graph, we generate all the non-isomorphic orientation of the edges, that can only be oriented in one

Algorithm 2 Pure half-integer vertices search

```

1: procedure PHI( $n, t$ )
2:  $\mathbb{G} = \{G = (V, E) \mid G \text{ connected, } \deg(i) \geq 2 \forall i \in V, |V| = n, |E| = n + t - 2\}$ 
3:  $\text{di}\mathbb{G} = \emptyset$ 
4: for  $G = (V, E) \in \mathbb{G}$  do
5:   if  $|\{i \in V \mid \deg(i) = 2\}| \leq t$  then
6:     add to  $\text{di}\mathbb{G}$  every non-isomorphic orientation of  $G$  s.t.
7:       · every edge can be oriented in only one way
8:       · every node has a maximum indegree of 2
9:    $\mathcal{V} = \emptyset$ 
10:  for  $\text{di}G = (V, A) \in \text{di}\mathbb{G}$  do
11:    if  $|\{i \in V \mid \text{indeg}(i) = 0\}| = 1$  then
12:      if  $|\{i \in V \mid \text{indeg}(i) = 1\}| = n - t$  then
13:        if  $|\{i \in V \mid \text{indeg}(i) = 2\}| = t - 1$  then
14:           $x_{ij} = 1/2$  iff  $(i, j) \in A$  is a solution of  $P_{CM}(n, t)$  with
15:            ·  $\{r\} = \{i \in V \mid \text{indeg}(i) = 0\}$ 
16:            ·  $V \setminus T = \{i \in V \mid \text{indeg}(i) = 1\}$ 
17:            ·  $T \setminus \{r\} = \{i \in V \mid \text{indeg}(i) = 2\}$ 
18:          if  $x$  is a feasible vertex of  $P_{CM}(n, t)$  then
19:            add  $x$  to  $\mathcal{V}$ 
20: return  $\mathcal{V}$ 

```

way because of Lemma 10, and such that every node has a maximum indegree of 2 since we have Constraint (1.15d) and we are dealing with PHI solutions. This generation of digraphs can be done with the command `watercluster2` of `nauty`. The obtained digraph can be mapped into a spanning PHI vertex of $P_{CM}(n, t)$ for every feasible case. In particular, we have to check that: (i) There exist exactly $n - t$ nodes with in-degree 1 (Steiner nodes); (ii) There exists one node with in-degree 0 (root); (iii) There exist exactly $t - 1$ nodes of in-degree 2 (terminals). We filter all the generated graphs for these properties and then check if the remaining ones are vertices of $P_{CM}(n, t)$. This procedure called $\text{PHI}(n, t)$, is illustrated in Algorithm Algorithm 2.

Observation 8. Note how the $\text{PHI}(n, t)$ can be generalized to vertex attaining values in the set $\{0, 1/m\}$ just by changing some values: the indegree of the terminal nodes must now be m , as well as the outdegree of the root, while the indegree of the Steiner nodes is again 1. This gives us a total number of edges of $n + (m - 1) \times t - m$. In addition, every node has degree at least $\min(3, m)$; if $m > 3$ the number of nodes with degree 3 is at most $n - t$; there must exist one node of indegree 0, $n - t$ nodes of indegree 1, and $t - 1$ nodes of indegree m . In Section 1.5.3, the case of $m = 4$ is discussed in more details, with the algorithm for this particular case being presented in Appendix A.c.

1.5 Computational results

In this section, we aim to generate vertices of the P_{CM} polytope and, for any vertex, evaluate the maximum integrality gap that can be attained at that vertex. Recall that vertices of the P_{BCR} polytope that are feasible for the CM formulation are also vertices of the P_{CM} polytope.

Specifically, our approach involves: (i) running the two proposed heuristics for small values of n to produce vertices of P_{CM} , (ii) computing the maximum integrality gap associated with these vertices by solving the Gap problem, and (iii) extending our analysis beyond purely half-integer vertices.

Implementation details All the tests are executed on a desktop computer with a CPU 13th Gen Intel(R) Core(TM) i5-13600 and 16 GB of RAM. All the functions have been implemented in Python. For the optimization tasks, we use the commercial solver Gurobi 11.0 [63].

1.5.1 Lower bounds for the integrality gap for $n \leq 10$

Our first set of algorithmic experiments aims at generating nontrivial vertices of P_{CM} having a large integrality gap. Table 1.2 and Table 1.3 present the lower bounds on the integrality gap and the number of nonintegral vertices we can compute with our two heuristics presented in Section 1.4. Recall that the vertices computed by the OTC procedures are filtered by isomorphism in post-processing. In the following paragraphs, we discuss the results for specific values of the number of vertices, starting from $n = 6$. Recall that results for $n \leq 5$ are presented in Table 1.1: we computed every vertex with Polymake, finding a maximum value of integrality gap equal to 1. Further details can be found in Appendix A.a.

($n = 6$) For $n = 6$, for any value of $t \leq 3 \leq n - 1$, the best lower bound we compute is always equal to 1. We conjecture that for $n = 6$, the CM and the BCR formulation have a gap equal to 1.

($n = 7$) For $n = 7$, we compute four vertices attaining the gap of $\frac{10}{9}$ with the heuristic OTC; three of them belong to the same class of isomorphism. Figure 1.2 shows those four vertices, where Figures Figure 1.2b, Figure 1.2c, and Figure 1.2d show the three isomorphic graphs. Moreover, these vertices are *pure half-integer*, e.g., $x_a \in \{0, \frac{1}{2}\}$. Although the directed support graph is the same for the four vertices, the arc orientation changes, and, in particular, the node labeled as “root” is different. Note that the PHI heuristic can only find two of the four vertices since it can find only one vertex for every class of isomorphism of node-colored edge-weighted directed graphs. On the contrary, the OTC heuristic may find more than one representative for the same class of isomorphism.

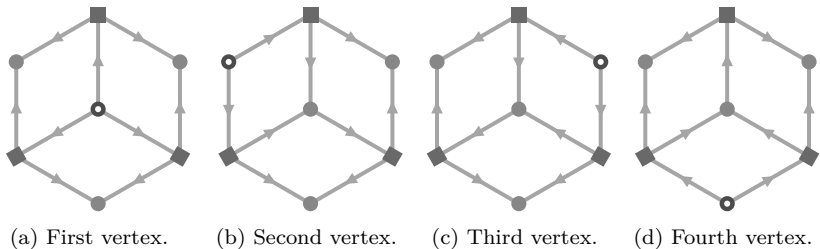


Figure 1.2: Fractional vertices of $(7, 4)$, all with integrality gap $10/9$. Hollow circle: root; Circles: Terminals; Square: Steiner node. Note that the second, third, and fourth vertex belong to the same class of isomorphism, while the first one belongs to another class.

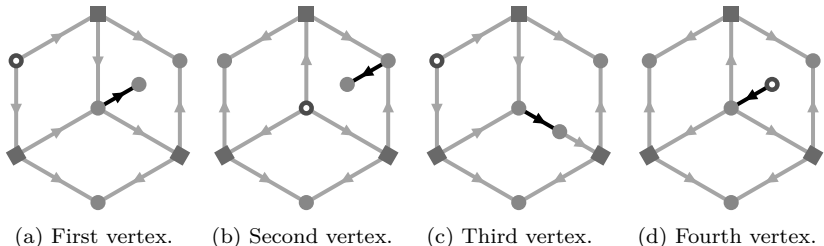


Figure 1.3: Fractional vertices of $(8, 5)$, all with integrality gap $10/9$. Hollow circle: root. Circles: terminals. Squares: Steiner nodes.

($n = 8$) The case $n = 8$ is more involved. The PHI does not find fractional vertices for the cases $t = 3, 4$. While the OTC does not find fractional vertices for $t = 3$, it finds again the vertices of $(7, 4)$ since it does not only find spanning vertices, as we already discussed. Both heuristics only find fractional vertices of integrality gap 1 for the case $t = 6$. For $t = 7$, only the PHI heuristic finds fractional vertices, all of them with an integrality gap of 1. The most interesting case is $t = 5$: the maximum integrality gap is depicted in Figure 1.4a while different values of integrality gap are depicted in Figure 1.4d and Figure 1.4e. Note that the maximum integrality gap of this case for the PHI heuristic is $12/11$, while the maximum integrality gap for the OTC heuristic is again $10/9$: some of the vertices attaining this value are depicted in Figure 1.3. Note also how these vertices can be obtained from vertices of $(7, 4)$, as explained in Observation 5.

($n = 9$) For $n = 9$, we can only run the PHI heuristics, as the OTC heuristic runs out of memory. Note how no vertices for the case $t = 3, 4$ are found, accordingly to the second point of Lemma 11, while for the cases $t = 7, 8$ only

Table 1.2: PHI versus OTC heuristic. The third and sixth columns report the number of non-isomorphic vertices; the fourth and seventh columns report the maximum value of the gap obtained for these vertices; the fifth and eighth columns report the number of vertices attaining the maximum gap.

n	t	PHI			OTC		
		# vert.	max gap	# vert. max. gap	# vert.	max gap	# vert. max. gap
6	4	1	1	1	0	-	-
	5	7	1	7	0	-	-
7	4	2	10/9	2	11	10/9	2
	5	46	1	46	19	1	19
	6	71	1	71	8	1	8
8	4	0	-	-	19	10/9	2
	5	89	12/11	15	195	10/9	14
	6	1 070	1	1 070	239	1	239
	7	758	1	758	0	-	-

vertices of integrality gap 1 are found. The maximum values of the integrality gap for $t = 5, 6$ are $10/9$ and $14/13$, as shown in Figure 1.4b and Figure 1.4c, respectively. Different values of integrality gap are depicted in Figure 1.4. Notice how all the non-trivial values of integrality gaps found for $n \leq 9$ are of the form $\frac{2m}{2m-1}$, with $m = 5, 6, 7, 8, 9, 10, 11, 12$.

($n \geq 10$) For $n \geq 10$, even the PHI heuristic shows its limits. For $t \in \{8, 9\}$, the heuristic did not terminate within a timelimit of 80 hours. The most interesting case we face is $t = 6$, where we found a vertex with an integrality gap of $19/18$. In this case, the value is of the form $\frac{2m+1}{2m}$, in contrast to what we found for $n \leq 9$.

For $n = 11$ and $n = 12$, we show that the cases $t \leq 5$ did not lead to any feasible PHI vertex. Tests with larger values of t were computationally currently untractable.

Note that no vertices with an integrality gap greater than 1 were found for the case $t = 3$. The exactness for this case is discussed in [138, Section 2.7.1].

1.5.2 A comparison between the two proposed heuristics

In this subsection, we discuss an in-depth comparison between the PHI and OTC heuristics. First, notice how neither of the two are exhaustive algorithms: at least one vertex can be found by the OTC heuristic but not by the PHI heuristic, and vice versa (see Figures Figure 1.3a and Observation 7). While the PHI heuristic is tailored for vertices with particular values, and so with a particular structure, the OTC is general enough to find different types of

Table 1.3: Partial performances of the PHI heuristic for $n \geq 9$.

n	t	# vert.	max gap	# vert. max. gap
9	5	64	10/9	12
	6	4 389	14/13	200
	7	21 121	1	21 121
	8	8 987	1	8 987
10	5	15	10/9	7
	6	7 386	10/9	73
	7	155 120	16/15	2 653

vertices; moreover, it remains an open question whether the heuristic becomes an exhaustive search by dropping the connectivity constraint.

Computationally, the OTC heuristic is highly demanding, even when limited to generating only connected graphs. For each generated graph, all possible assignments of the root, terminal nodes, and Steiner nodes must be considered, and an LP must be solved for every assignment. Moreover, there is no guarantee that the solution to the LP will be fractional; in fact, it may correspond to an equivalent integer solution. In addition, the OTC heuristic does not ensure that the generated solutions are non-isomorphic, necessitating a post-processing step to filter out isomorphic graphs based on node-colored edge-weighted graph isomorphism. The algorithm does not even guarantee finding spanning vertices. The PHI heuristic doesn't generate isomorphic graphs, and hence, every vertex generated belongs to a unique class of isomorphism. In addition, no LP needs to be solved since, given the orientation of the arcs, the role of every node is uniquely determined. Lastly, note that in the OTC heuristic, we have applied the extra bounds on the number of edges $n \cdot t - t^2$ derived after a first set of computational experiments. Without this hypothesis, OTC is untractable for $n \geq 8$. Table 1.2 results are obtained with this extra constraint. Note also that, even with the aforementioned bound on the number of edges, OTC is untractable for $n \geq 9$.

1.5.3 Beyond pure half-integer vertices

The computational results of the previous subsection show that the PHI heuristic is better than OTC in finding interesting vertices of the CM polytope. In addition, the PHI heuristic can be extended to enumerate all the vertices of the type $\{0, 1/m\}$, $m \in \mathbb{N}_{\geq 3}$. For example, an interesting case is $m = 4$, namely, when the vertices take value only in the set $\{0, 1/4\}$. Let us call these vertices *pure one-quarter* (POQ) vertices. In this case, our heuristic would work for each pair (n, t) as follows.

1. Generate all the non-isomorphic graphs having (i) every node of degree

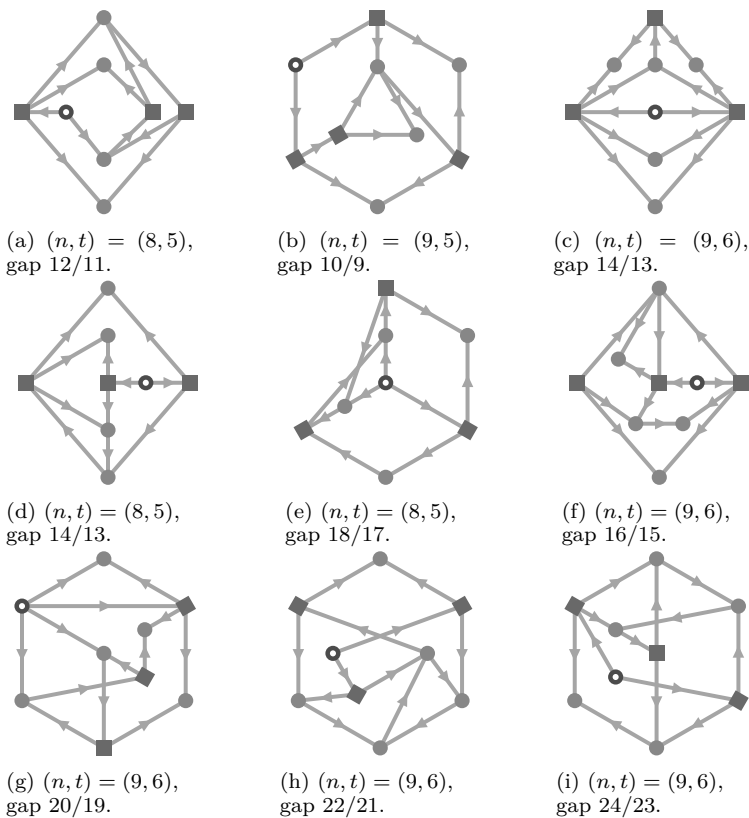


Figure 1.4: Fractional vertices of different gaps for different values of (n, t) . The first three vertices attain the maximum gap for their respective value of (n, t) .

- at least equal to 3, and (ii) exactly $n + 3t - 4$ arcs.
- 2. Filter the list of vertices computed at Step 1 by excluding all graphs with more than $n - t - 1$ nodes with degree 3. In POQ vertices, there are $n - t$ Steiner nodes that must have a minimum degree of 3, and the terminals have a minimum degree of 4.
- 3. For each oriented graph, we use **watercluster2** to get all possible orientations of edges, assuming that the maximum indegree must be equal to 4.
- 4. We filter out the list obtained at Step 3 and keep only the directed graphs

having (i) exactly one node with in-degree 0 (the root), (ii) exactly $t - 1$ nodes with in-degree 4, (iii) exactly $n - t$ nodes with in-degree 1.

In [82], the authors show that the integrality gap of the BCR formulation is at least $\frac{8}{7}$ by exposing an instance leading to such a gap. The instance has 15 nodes and 8 terminals. The optimal vertex is of POQ type, and it originated from a personal communication between Martin Skutella and the authors of [82]. This makes POQ vertices particularly relevant for our study. Figure 1.1 shows Skutella’s graph. Note that solving the Gap function for the CM formulation leads to a gap equal to $\frac{8}{7}$. Hence, the maximum integrality gap for Skutella’s graph is exactly $\frac{8}{7}$.

We defined a modified version of the PHI algorithm to find POQ vertices (for further details, see Appendix A.c), but we were not able to find any vertex with the above properties before the computation became intractable, that is $n \geq 8$.

1.6 Conclusion and future works

In this paper, we have studied the metric STP on graphs, focusing on computing lower bounds for the integrality gap for the BCR and the CM formulations. We introduced a novel ILP formulation, the Complete Metric (CM) model, tailored for the complete metric Steiner tree problem. This formulation overcomes the limitations of the BCR formulation in the metric case. For the CM formulation, we prove several structural properties of the polytope associated with its natural LP relaxation.

The core of our contribution presented in this paper is extending the Gap problem introduced in [30] for the symmetric TSP, to the metric Steiner tree problem. To speed up the search for suitable vertices, we designed two heuristics. Our heuristics outperform the exact method obtained as a natural extension of [30] to the Steiner problem and can generate nontrivial vertices for n up to 10. Note that exact methods got stuck already for $n = 6$.

We compare the performances of the two heuristics and their impact on providing insights into the exact value of the integrality gap. Although we cannot improve the bound of $\frac{10}{9}$ with $n \leq 10$, we find different structures of vertices leading to non-trivial gaps. By directly exploring vertices similar to those yielding the highest gaps for $n > 10$, we observed that these structures cannot be present for small values of n . Hence, we conjecture that with $n \leq 10$, the highest gap is $\frac{10}{9}$.

We retain that our study raises several interesting research questions. First, can an ad-hoc branching procedure be designed for the CM formulation? Second, can we improve the OTC heuristic by reducing the number of combinations we have to analyze without losing any of the outputs? Third, can we prove any further characterization of the vertices that reduce the effort for Polymake, similarly to what has been done in [14]? Lastly, can we enhance the design and implementation of the POQ heuristic to explore whether new

lower bounds for the integrality gap are achievable for this type of vertices in higher dimensions?

We conclude this paper with two conjectures: First, we conjecture that our OTC procedure, without the restriction on connectedness and the bound on the number of edges, is exhaustive and, hence, every vertex of $P_{CM}(n, t)$ can be obtained as an optimal solution of a $\{1, 2\}$ -cost instance. Second, we conjecture that every spanning vertex x of $P_{CM}(n, t)$ with $x_{ij} \in \{0, 1/m\}$, $m \geq 2$, has an in-degree of 1 in every Steiner node, and hence our PHI is exhaustive for every pure half-integer spanning vertex.

NEURAL NETWORK TRAINING VIA MIXED INTEGER LINEAR PROGRAMMING

2.1 Introduction

State-of-the-art deep neural networks (NNs) contain a huge number of neurons organized in many layers, and they require an immense amount of data for training [85]. The training process is computationally demanding and is typically performed by stochastic gradient descent algorithms running on large GPU- or even TPU-based clusters. Whenever the trained (deep) neural network contains many neurons, also the network deployment is computationally demanding. However, in some real-life applications, extensive GPU-based training might be infeasible, training data might be scarce with only a few data points per class, or the hardware using the NN at inference time might have a limited computational power, as for instance, whenever the NN is executed on an industrial embedded system [27].

Binarized Neural Networks (BNNs) were introduced in [68] as a response to the challenge of running NNs on low-power devices. BNNs contain only binary weights and binary activation functions, and hence they can be implemented using only efficient bit-wise operations. However, the training of BNNs raises interesting challenges for gradient-based approaches due to their combinatorial structure. In previous works [132], it is shown that the training of a BNN can be performed by using combinatorial optimization solvers: a hybrid constraint programming (CP) and mixed integer programming (MIP) approach outperformed the stochastic gradient approach proposed by [68] if restricted to a few-shot-learning context ([136]). The work [132] has been furthered by a number of authors more recently, as we survey in the next section.

Indeed, combinatorial approaches (principally MIP) for training neural networks, both discrete and continuous, have been employed in the literature, as demonstrated in subsequent works [107, 83]. MIP optimization has been explored in the machine learning community, as in [88, 67, 146] for instance. Various architectures and activation functions have been utilized in these studies. Solver-based training has the advantage that, in principle, the optimal NN weights can be found for the training data and that network optimization (e.g. pruning) or adversarial hardening can be performed.

The main challenge in training a NN by an exact MIP-based approach is the limited amount of training data that can be used since, otherwise, the

size of the optimization model explodes. In the recent work of [129], the combinatorial training idea was further extended to Integer-valued Neural Networks (INN). Exploiting the flexibility of MIP solvers, the authors were able to (i) minimize the number of neurons during training and (ii) increase the number of data points used during training by introducing a MIP batch training method.

We remark that training a NN with a MIP-based approach is more challenging than solving a verification problem, as in [51, 3], even if the structure of the nonlinear constraints modelling the activation functions is similar. In NNs verification [77], the weights are given as input, while in MIP-based training, the weights are the *decision variables* that must be computed.

We note several lines of work aiming at producing compact and simple NNs that maintain acceptable accuracy, e.g. in terms of parameter pruning [119, 148, 35, 49], loss function improvement [127], gradient approximation [116] and network topology structure [89].

In the context of MIP-based training and optimization of NNs, this work proposes new methods to improve the training of BNNs and INNs. In summary, our contributions are (i) the formulation of a MILP model with a multi-objective target that consists of already existing single-objective steps in a lexicographic order, (ii) the implementation of an ensemble of few-bits NNs in which each of them is specialized in a specific classification task, (iii) the proposal of a voting scheme inspired by One-Versus-One (OVO) strategy, tailored specifically for the constructed ensemble of NNs. Our computational results using the MNIST and the Fashion-MNIST dataset show that the *BeMi* ensemble permits to use for training up to 40 data points per class, thanks to the fact that the OVO strategy results in smaller MILPs, reaching an average accuracy of 81.8% for MNIST and 70.7% for Fashion-MNIST. In addition, thanks to the multi-objective function that minimizes the number of links, i.e. the connections between different neurons, up to 75% of weights are set to zero for MNIST, and up to 48% for Fashion-MNIST. We also perform additional experiments on the Heart Disease dataset, reaching an average accuracy of 78.5%. A preliminary report of this work appeared at the LION'23 conference [17]. This work better motivates and situates our approach in the state of the art, develops our ensemble approach for INNs (not only BNNs), presents more extensive and improved empirical results, and analyses the distribution of INN weights.

Outline. The remainder of this chapter is as follows. Section 2.2 situates our work in the literature. Section 2.3 introduces the notation and defines the problem of training a single INN with the existing MIP-based methods. Section 2.4 presents the *BeMi* ensemble, the majority voting scheme, and the improved MILP model to train a single INN. Section 2.5 presents the computational results on the MNIST, Fashion-MNIST and Heart Disease datasets. Finally, Section 2.6 ends with a perspective on future work.

2.2 Related Works

Recently, there has been a growing research interest in studying the impact of machine learning on improving traditional operations research methods (e.g. see [13] and [36]), in designing integrated predictive and modelling frameworks, as in [15], or in embedding pre-trained machine learning model into an optimization problem (e.g. see [91, 98, 133]). In this work, we take a different perspective, and we study how an exact MILP solver can be used for training Machine Learning models, more specifically, to train (Binary or Integer) Neural Networks. In the following paragraphs, we first review two recent applications to MILP solvers in the context of neural networks (i.e. weight pruning and NN verification), and later, we review the few works that tackled the challenge of training an NN using an exact solver.

MIP-based Neural Networks Pruning. Recent works have shown interesting results on *pruning* a trained neural network using an optimization approach based on the use of MIP solvers [119, 148, 35, 49, 58]. While pruning a trained neural network, the weights are fixed, and the optimization variables represent the decision to keep or remove an existing weight different from zero.

MIP-based Neural Networks Verification. Another successful application of exact solvers in the context of Neural Networks is for tackling the verification problem, that is, to verify under which conditions the accuracy of a given trained neural network does not deteriorate. In other words, in NN verification, the optimization problem consist in finding adversarial examples using a minimal distortion of the input data. The use of MIP solvers for this application was pioneered in [131], and later studied in several papers, as for instance, [131, 3, 28, 51, 130]. For a broader discussion of the use of polyhedral approach to verification, see Section 4 in [69]. In NN verification, the weights of the NN are given as input, and the optimization problem consists of assessing how much the input can change without compromising the output of the network. Indeed, also verification is a different optimization problem than the exact training the we discuss in the following sections.

Exact training of Neural Networks. The utilization of MIP approaches for training neural networks has already been explored in the literature, primarily in the context of few-shot learning and NNs with low-bit parameters [132, 129]. One of the main advantages of these approaches is the ability to simultaneously train and optimize the network architecture. While the modelling that defines the structure of the neural network is rigid and heavily relies on the discrete nature of the parameters, the choice of the objective function provides more flexibility and allows for the optimization of various network characteristics. For instance, it enables minimizing the number of connections in a fixed architecture network, thereby promoting lightweight

architectures.

From now on, by INN we indicate a general NN whose weights take value in the set $\{-P, \dots, P\}$, where integer $P \geq 1$. Notice this choice include the special case of BNNs ($P = 1$). When referring to an INN with $P > 1$, we will write *non-trivial* INN.

By incorporating the power of both CP and MIP, [132]’s study showcased the effectiveness of leveraging combinatorial optimization methods for training BNNs in a few-shot learning context. Two types of objective functions were selected to drive the optimization process. The first objective function aimed at promoting a lightweight architecture by minimizing the number of non-zero weights. This objective sought to reduce the overall complexity of the neural network, allowing for efficient computation and resource utilization. Note that with this choice, pruning is not necessary: the possibility of setting a weight to zero is equivalent to removing the corresponding weight (i.e. connection). In contrast, the second objective function focused on enhancing the robustness of the network, particularly in the face of potential noise in the input data. Robustness here refers to the ability of the network to maintain stable and reliable performance even when the input exhibits variations or disturbances. [129]’s research extends [132]’s single-objective approach to the broader class of Integer Neural Networks (INNs). In addition to leveraging the objectives of architectural lightness and robustness, a novel objective aimed at maximizing the number of correctly classified training data instances is introduced. It is worth noting that this type of objective shares some similarities with the goals pursued by gradient descent methods, albeit with a distinct formulation. An interesting observation is that this objective formulation remains feasible in practice, ensuring that a valid solution can be obtained. We remark that both papers propose single-objective models that involve training a single neural network to approximate a multi-classification function.

Ensembles of neural networks are well-known to yield more stable predictions and demonstrate superior generalizability compared to single neural network models [140]. In this work, we aim to redefine the concept of structured ensemble by composing our ensemble of several networks, each specialized in a distinct task. Given a classification task over k classes, the main idea is to train $\frac{k(k-1)}{2}$ INNs, where every single network learns to discriminate only between a given pair of classes. When a new data point (e.g. a new image) must be classified, it is first fed into the $\frac{k(k-1)}{2}$ trained INNs, and later, using a Condorcet-inspired majority voting scheme [147], the most frequent class is predicted as output. This method is similar to and generalizes the Support Vector Machine – One-Versus-One (SVM-OVO) approach [24], while it has not yet been applied within the context of neural networks, to the best of our knowledge.

For training every single INN, our approach extends the methods introduced in [132] and [129], described above.

2.3 Few-bit Neural Networks

In this section, we formally define a Binarized Neural Network (BNN) and an Integer Neural Network (INN) using the notation as in [132] and [129], while, in the next section, we show how to create a structured ensemble of INNs.

2.3.1 Binarized Neural Networks

The architecture of a BNN is defined by a set of layers $\mathcal{N} = \{N_0, N_1, \dots, N_L\}$, where $N_l = \{1, \dots, n_l\}$, and n_l is the number of neurons in the l -th layer. Let the training set be $\mathcal{X} := \{(\mathbf{x}^1, y^1), \dots, (\mathbf{x}^t, y^t)\}$, such that $\mathbf{x}^i \in \mathbb{R}^{n_0}$ and $y^i \in \{-1, +1\}^{n_L}$ for every $i \in T = \{1, 2, \dots, t\}$. The first layer N_0 corresponds to the size of the input data points \mathbf{x}^k . Regarding n_L , we make the following consideration. For a classification problem with $|\mathcal{I}|$ classes, we set $n_L := \lceil \log_2 |\mathcal{I}| \rceil$. For the case $|\mathcal{I}| = 2$, therefore, n_L will be equal to 1. This is consistent with binary classification problems, as the two classes can be represented as $+1$ and -1 . When $|\mathcal{I}| = 4$, then n_L will be equal to 2, and the four classes will be represented by $(+1, +1)$, $(+1, -1)$, $(-1, +1)$, and $(-1, -1)$. When $|\mathcal{I}|$ is a power of 2, the procedure generalizes in an obvious manner. However, when the number of classes is not a power of 2, we still choose n_L as the nearest integer greater than or equal to the base-2 logarithm of that number, with the caveat that a single network may opt not to classify. For example, if $|\mathcal{I}| = 3$, then $n_L = 2$, and $(+1, +1)$ will be associated with the first class, $(+1, -1)$ will be associated with the second class, $(-1, +1)$ will be associated with the third class, while $(-1, -1)$ will be interpreted as ‘unclassified’.

The link between neuron i in layer N_{l-1} and neuron j in layer N_l is modelled by weight $w_{ilj} \in \{-1, 0, +1\}$. Note that the binarized nature is encoded in the ± 1 weights, while when a weight is set to zero, the corresponding link is removed from the network. Hence, during training, we are also optimizing the architecture of the BNN.

The activation function is the binary function

$$\rho(x) := 2 \cdot \mathbb{1}(x \geq 0) - 1, \quad (2.1)$$

that is, a sign function reshaped such that it takes ± 1 values. Here the indicator function $\mathbb{1}(p)$ outputs $+1$ if proposition p is verified, and 0 otherwise. This choice for the activation function has been made in line with the literature [68].

In this work, we aim to build different MILP models for the simultaneous training and optimization of a network architecture. To model the activation function (2.1) of the j -th neuron of layer N_l for data point \mathbf{x}^k , we introduce a binary variable $u_{l,j}^k \in \{0, 1\}$ for the indicator function $\mathbb{1}(p)$. To re-scale the value of $u_{l,j}^k$ in $\{-1, +1\}$ and model the activation function value, we introduce the auxiliary variable $z_{l,j}^k = (2u_{l,j}^k - 1)$. For the first input layer, we set $z_{0,j}^k = x_j^k$; for the last layer, we account in the loss function whether $z_{L,h}^k$ is different from

y_h^k . The definition of the activation function becomes

$$z_{lj}^k = \rho \left(\sum_{i \in N_{l-1}} z_{(l-1)i}^k w_{ilj} \right) = 2 \cdot \mathbb{1} \left(\sum_{i \in N_{l-1}} z_{(l-1)i}^k w_{ilj} \geq 0 \right) - 1 = 2u_{lj}^k - 1.$$

Notice that the activation function at layer N_l gives a nonlinear combination of the output of the neurons in the previous layer N_{l-1} and the weights w_{ilj} between the two layers. Section 2.4.1 shows how to formulate this activation function in terms of mixed integer linear constraints. We remark that the modelling we proposed has already been presented in the literature by [132].

The choice of a family of parameters $W := \{w_{ilj}\}_{l \in \{1, \dots, L\}, i \in N_{l-1}, j \in N_l}$ determines the function

$$f_W : \mathbb{R}^{n_0} \rightarrow \{\pm 1\}^{n_L}.$$

The training of a neural network is the process of computing the family W such that f_W classifies correctly both the given training data, that is, $f_W(\mathbf{x}^i) = y^i$ for $i = 1, \dots, t$, and new unlabelled testing data.

In the training of a BNN, we follow two machine learning principles for generalization: robustness and simplicity [132]. In doing so, we target two objectives: (i) the resulting function f_W should generalize from the input data and be *robust* to noise in the input data; (ii) the resulting network should be *simple*, that is, with the smallest number of non-zero weights that permit to achieve the best accuracy.

Regarding the robustness objective, there is argument that deep neural networks have inherent robustness because mini-batch stochastic gradient-based methods implicitly guide toward robust solutions [75, 76, 104]. However, as shown in [132], this is false for BNNs in a few-shot learning regime. On the contrary, MIP-based training with an appropriate objective function can generalize very well [132, 129], but it does not apply to large training datasets, because the size of the MIP training model is proportional to the size of the training dataset.

One possible way to impose robustness in the context of few-shot learning is to maximize the margins of the neurons, that is, fixing one neuron, we aim at finding an ingoing weights configuration such that for every training input, the entry of the activation function evaluated at that neuron is confidently far away from the discontinuity point. Intuitively, neurons with larger margins require larger changes to their inputs and weights before changing their activation values. This choice is also motivated by recent works showing that margins are good predictors for the generalization of deep convolutional NNs [73].

Regarding the simplicity objective, a significant parameter is the number of connections [100]. The training algorithm should look for a NN fitting the training data while minimizing the number of non-zero weights. This approach can be interpreted as a simultaneous compression during training, and it has been already explored in recent works [113, 120].

MIP-based BNN training. In [132], two different MIP models are introduced: the **Max-Margin**, which aims to train robust BNNs, and the **Min-Weight**, which aims to train simple BNNs. These two models are combined with a CP model into two hybrid methods **HW** and **HA** in order to obtain a feasible solution within a fixed time limit. [132] employ CP because their MIP models do not scale as the number of training data increases. We remark that in that work, the two objectives, robustness and simplicity, are never optimized simultaneously.

Gradient-based BNN training. In [68], a gradient descent-based method is proposed, consisting of a local search that changes the weights to minimize a square hinge loss function. Note that a BNN trained with this approach only learns ± 1 weights. An extension of this method that exploits the same loss function but admits zero-value weights, called **GD_t**, is proposed in [132], to facilitate the comparison with the other approaches.

2.3.2 Integer Neural Networks

A more general discrete NN can be obtained when the weights of the network lie in the set $\{-P, -P + 1, \dots, -1, 0, 1, \dots, P - 1, P\}$, where P is a positive integer. The resulting network is called **INN**, and by letting $P = 1$, we obtain the BNN presented in the previous subsection. The activation function ρ and the binary variables u_{ij}^k are defined as above. The principles leading the training are again simplicity and robustness.

An apparent advantage of using more general integer neural networks lies in the fact that the parameters have increased flexibility while still maintaining their discrete nature. Additionally, by appropriately selecting the parameter P , one can determine the number of bits used for each parameter. For instance, $P = 1$ corresponds to 1-bit, $P = 3$ corresponds to 2-bit, $P = 8$ corresponds to 3-bit, and in general, $P = 2^{n-1}$ corresponds to n -bit.

MIP-based INN training. In [129], three MIP models are proposed in order to train INNs. The first model, **Max-Correct**, is based on the idea of maximizing the number of corrected predicted images; the second model, **Min-Hinge**, is inspired by the squared hinge loss (compare [68]); the last model, **Sat-Margin**, combines aspects of both the first two models. These three models always produce a feasible solution but use the margins only on the neurons of the last level, obtaining, hence, less robust NNs.

Relations to Quantized Neural Networks. By using an exact MIP solver for training Integer NNs, we are dealing directly with the problem of training a quantized neural network, where all the weights are restricted to take values over a small domain, as discussed above. For instance, as reviewed in [54], there is a growing trend in training NNs using floating point numbers in low

precision, that is, using only as few as 8 bits per weight (see for example [10]). However, most of the work in the ML literature either focuses on the impact of low-precision arithmetic on the computation of the (stochastic) gradient and in the backpropagation algorithm or focuses on how to *quantize* a trained NN by minimizing the deterioration in the accuracy. In our work, we take a different perspective on quantization methods, since we do not rely on a gradient-based method to train our INN, but we model and directly solve the problem of training the NN using only a restricted number of integer weights, which is called *Integer-only Quantization* in [54]. Moreover, by using an exact MIP solver, we can directly find the optimal weights of our (small) INN without running the risks to be trapped into a local minima as stochastic gradient-based methods.

2.4 The *BeMi* ensemble

This section first introduces a multi-objective model that allows a simultaneous training and optimization for an INN (Section 2.4.1), and then proposes a method for combining a set of neural networks for classification purposes (Section 2.4.2).

2.4.1 A multi-objective MILP model for training INNs

For ease of notation, we denote with $\mathcal{L} := \{1, \dots, L\}$ the set of layers and with $\mathcal{L}_2 := \{2, \dots, L\}$, $\mathcal{L}^{L-1} := \{1, \dots, L-1\}$ two of its subsets. We also denote with $\mathbf{b} := \max_{k \in T, j \in N_0} \{|x_j^k|\}$ a bound on the values of the training data.

The multi-objective target. A few MIP models are proposed in the literature to train INNs efficiently. In this work, to train a single INN, we use a lexicographic multi-objective function that results in the sequential solution of three different state-of-the-art MIP models: the **Sat-Margin (SM)** described in [129], the **Max-Margin (MM)**, and the **Min-Weight (MW)**, both described in [132]. The first model **SM** maximizes the number of confidently correctly predicted data. The other two models, **MM** and **MW**, aim to train a INN following two principles: robustness and simplicity. Our model is based on a lexicographic multi-objective function: first, we train a INN with the model **SM**, which is fast to solve and always gives a feasible solution. Second, we use this solution as a warm start for the **MM** model, training the INN only with the images that **SM** correctly classified. Third, we fix the margins found with **MM**, and minimize the number of active weights with **MW**, finding the simplest INN with the robustness found by **MM**.

Problem variables. The critical part of our model is the formulation of the nonlinear activation function (2.1). We use an integer variable $w_{ilj} \in \{-P, -P+1, \dots, P\}$ to represent the weight of the connection between neuron $i \in N_{l-1}$ and neuron $j \in N_l$. Variable u_{lj}^k models the result of the indicator

function $\mathbb{1}(p)$ that appears in the activation function $\rho(\cdot)$ for the training instance \mathbf{x}^k . The neuron activation is actually defined as $2u_{lj}^k - 1$. We introduce auxiliary variables c_{ilj}^k to represent the products $c_{ilj}^k = (2u_{lj}^k - 1)w_{ilj}$. Note that, while in the first layer, these variables share the same domain of the inputs, from the second layer on, they take values in $\{-P, -P+1, \dots, P\}$. Finally, the auxiliary variables \hat{y}_j^k represent a predicted label for the input \mathbf{x}^k , and variable q_j^k are used to take into account the data points correctly classified.

The procedure is designed such that the parameter configuration obtained in the first step is used as a warm start for the (MM). Similarly, the solution of the second step is used as a warm start for the solver to solve (MW). In this case, the margins lose their nature as decision variables and become deterministic constants derived from the solution of the previous step.

Sat-Margin (SM) model. We first train our INN using the following SM model.

$$\max \sum_{k \in T} \sum_{j \in N_L} q_j^k \quad (2.2a)$$

$$\text{s.t. } q_j^k = 1 \implies \hat{y}_j^k \cdot y_j^k \geq \frac{1}{2} \quad \forall j \in N_L, k \in T, \quad (2.2b)$$

$$q_j^k = 0 \implies \hat{y}_j^k \cdot y_j^k \leq \frac{1}{2} - \hat{\epsilon} \quad \forall j \in N_L, k \in T, \quad (2.2c)$$

$$\hat{y}_j^k = \frac{2}{P \cdot (n_{L-1} + 1)} \sum_{i \in N_{L-1}} c_{ilj}^k \quad \forall j \in N_L, k \in T, \quad (2.2d)$$

$$u_{lj}^k = 1 \implies \sum_{i \in N_{l-1}} c_{ilj}^k \geq 0 \quad \forall l \in \mathcal{L}^{L-1}, j \in N_l, k \in T, \quad (2.2e)$$

$$u_{lj}^k = 0 \implies \sum_{i \in N_{l-1}} c_{ilj}^k \leq -\epsilon \quad \forall l \in \mathcal{L}^{L-1}, j \in N_l, k \in T, \quad (2.2f)$$

$$c_{i1j}^k = x_i^k \cdot w_{i1j} \quad \forall i \in N_0, j \in N_1, k \in T, \quad (2.2g)$$

$$c_{ilj}^k = (2u_{(l-1)j}^k - 1)w_{ilj} \quad \forall l \in \mathcal{L}_2, i \in N_{l-1}, j \in N_l, k \in T, \quad (2.2h)$$

$$q_j^k \in \{0, 1\} \quad \forall j \in N_L, k \in T, \quad (2.2i)$$

$$w_{ilj} \in \{-P, -P+1, \dots, P\} \quad \forall l \in \mathcal{L}, i \in N_{l-1}, j \in N_l, \quad (2.2j)$$

$$u_{lj}^k \in \{0, 1\} \quad \forall l \in \mathcal{L}^{L-1}, j \in N_l, k \in T, \quad (2.2k)$$

$$c_{i1j}^k \in [-P \cdot \mathfrak{b}, P \cdot \mathfrak{b}] \quad \forall i \in N_0, j \in N_1, k \in T, \quad (2.2l)$$

$$c_{ilj}^k \in \{-P, -P+1, \dots, P\} \quad \forall l \in \mathcal{L}_2, i \in N_{l-1}, j \in N_l, k \in T, \quad (2.2m)$$

with $\hat{\epsilon} := \frac{\epsilon}{2P \cdot (n_{L-1} + 1)}$. The objective function (2.2a) maximizes the number of data points that are correctly classified. Note that ϵ is a small quantity standardly used to model strict inequalities. The implication constraints (2.2b) and (2.2c) and constraints (2.2d) are used to link the output \hat{y}_j^k with the corresponding variable q_j^k appearing in the objective function. The implication constraints (2.2e) and (2.2f) model the result of the indicator function for the

k -th input data. The constraints (2.2g) and the bilinear constraints (2.2h) propagate the results of the activation functions within the neural network. We linearize all these constraints with standard big-M techniques [143].

The solution of model (2.2a)–(2.2m) gives us the solution vectors $\mathbf{c}_{\text{SM}}, \mathbf{u}_{\text{SM}}, \mathbf{w}_{\text{SM}}, \hat{\mathbf{y}}_{\text{SM}}, \mathbf{q}_{\text{SM}}$. We then define the set

$$\hat{T} = \{k \in T \mid q_{j_{\text{SM}}}^k = 1, \forall j \in N_L\}, \quad (2.3)$$

of confidently correctly predicted images. We use these images as input for the next Max-Margin MM, and we use the vector of variables $\mathbf{c}_{\text{SM}}, \mathbf{u}_{\text{SM}}, \mathbf{w}_{\text{SM}}$ to warm start the solution of MM.

Max-Margin (MM) model. The second level of our lexicographic multi-objective model maximizes the overall margins of every single neuron activation, with the ultimate goal of training a robust INN. Starting from the model SM, we introduce the margin variables m_{lj} , and we introduce the following Max-Margin model.

$$\max \quad \sum_{l \in \mathcal{L}} \sum_{j \in N_l} m_{lj} \quad (2.4a)$$

$$\begin{aligned} \text{s.t.} \quad & (2.2g) \text{--}(2.2m) & \forall k \in \hat{T}, \\ & \sum_{i \in N_{L-1}} y_j^k c_{iLj}^k \geq m_{Lj} & \forall j \in N_L, k \in \hat{T}, \end{aligned} \quad (2.4b)$$

$$u_{lj}^k = 1 \implies \sum_{i \in N_{l-1}} c_{ilj}^k \geq m_{lj} \quad \forall l \in \mathcal{L}^{L-1}, j \in N_l, k \in \hat{T}, \quad (2.4c)$$

$$u_{lj}^k = 0 \implies \sum_{i \in N_{l-1}} c_{ilj}^k \leq -m_{lj} \quad \forall l \in \mathcal{L}^{L-1}, j \in N_l, k \in \hat{T}, \quad (2.4d)$$

$$m_{lj} \geq \epsilon \quad \forall l \in \mathcal{L}, j \in N_l. \quad (2.4e)$$

Again, we can linearize constraints (2.4c) and (2.4d) with standard big-M constraints. This model gives us the solution vectors $\mathbf{c}_{\text{MM}}, \mathbf{u}_{\text{MM}}, \mathbf{w}_{\text{MM}}, \mathbf{m}_{\text{MM}}$. We then evaluate \mathbf{v}_{MM} as

$$v_{ilj_{\text{MM}}} = \begin{cases} 0 & \text{when } w_{ilj_{\text{MM}}} = 0, \\ 1 & \text{otherwise,} \end{cases} \quad \forall l \in \mathcal{L}, i \in N_{l-1}, j \in N_l. \quad (2.5)$$

Min-Weight (MW) model. The third level of our multi-objective function minimizes the overall number of non-zero weights, that is, the connections of the trained INN. We introduce the new auxiliary binary variable v_{ilj} to model the presence or absence of the link w_{ilj} . Starting from the solution of model MM, we fix $\hat{\mathbf{m}} = \mathbf{m}_{\text{MM}}$, and we pass the solution $\mathbf{c}_{\text{MM}}, \mathbf{u}_{\text{MM}}, \mathbf{w}_{\text{MM}}, \mathbf{v}_{\text{MM}}$ as a warm start to the following MW model:

$$\min \quad \sum_{l \in \mathcal{L}} \sum_{i \in N_{l-1}} \sum_{j \in N_l} v_{ilj} \quad (2.6a)$$

$$\text{s.t.} \quad (2.2g) \text{--}(2.2m) \quad \forall k \in \hat{T},$$

$$\sum_{i \in N_{L-1}} y_j^k c_{iLj}^k \geq \hat{m}_{Lj} \quad \forall j \in N_L, k \in \hat{T}, \quad (2.6b)$$

$$u_{lj}^k = 1 \implies \sum_{i \in N_{l-1}} c_{ilj}^k \geq \hat{m}_{lj} \quad \forall l \in \mathcal{L}^{L-1}, j \in N_l, k \in \hat{T}, \quad (2.6c)$$

$$u_{lj}^k = 0 \implies \sum_{i \in N_{l-1}} c_{ilj}^k \leq -\hat{m}_{lj} \quad \forall l \in \mathcal{L}^{L-1}, j \in N_l, k \in \hat{T}, \quad (2.6d)$$

$$-v_{ilj} \cdot P \leq w_{ilj} \leq v_{ilj} \cdot P \quad \forall l \in \mathcal{L}, i \in N_{l-1}, j \in N_l, \quad (2.6e)$$

$$v_{ilj} \in \{0, 1\} \quad \forall l \in \mathcal{L}, i \in N_{l-1}, j \in N_l. \quad (2.6f)$$

Note that whenever v_{ilj} is equal to zero, the corresponding weight w_{ilj} is set to zero due to constraint (2.6e), and, hence, the corresponding link can be removed from the network.

Lexicographic multi-objective. By solving the three models **SM**, **MM**, and **MW**, sequentially, we first maximize the number of input data that is correctly classified, then we maximize the margin of every activation function, and finally, we minimize the number of non-zero weights. The solution of the decision variables w_{ilj} of the last model **MW** defines our classification function $f_W : \mathbb{R}^{n_0} \rightarrow \{\pm 1\}^{n_L}$.

2.4.2 The BeMi structure

Having explained the various MIP models of INNs, we next introduce our ensemble approach for MIP-based training of INNs.

Ensemble. Define $\mathcal{P} := \{\{i, j\} \text{ s.t. } i \neq j, i, j \in \mathcal{I}\}$ as the set of all the subsets of the set \mathcal{I} that have cardinality 2, where \mathcal{I} is the set of the classes of the classification problem. Then our structured ensemble is constructed in the following way.

1. We train a INN denoted by \mathcal{N}_{ij} for every $\{i, j\} \in \mathcal{P}$, i.e. for each possible pair of elements of \mathcal{I} .
2. When testing a data point, we feed it to our list of trained INNs obtaining a list of predicted labels, namely we obtain the predicted label ϵ_{ij} from the network \mathcal{N}_{ij} .
3. We then apply a majority voting system.

The idea behind this structured ensemble is that, given an input \mathbf{x}^k labelled l ($= y^k$), the input is fed into $\binom{n}{2}$ networks where $n - 1$ of them are trained to recognize an input with label l . If all of the networks correctly classify the input \mathbf{x}^k as l , then at most $n - 2$ other networks can classify the input with a different label $l' \neq l$, and so the input is correctly labelled with the most occurring label l . With this approach, if we plan to use $r \in \mathbb{N}$ inputs for each label, we are feeding each of our INNs a total of $2 \cdot r$ inputs instead of feeding

$n \cdot r$ inputs to a single large INN. Clearly, when training the networks \mathcal{N}_{ij} and the network \mathcal{N}_{ik} , the inputs of the class i are the same, so we only need a total of r inputs for each class. When $n \gg 2$, it is much easier to train our structured ensemble of INNs rather than training one large INN because of the fact that the MILP model size depends linearly on the number of input data.

Majority voting system. After the training, we feed one input \mathbf{x}^k to our list of INNs, and we need to elaborate on the set of outputs.

Definition 3 (Dominant label). For every $b \in \mathcal{I}$, we define

$$C_b = \{\{i, j\} \in \mathcal{P} \mid \mathbf{c}_{ij} = b\},$$

and we say that a label b is a *dominant label* if $|C_b| \geq |C_l|$ for every $l \in \mathcal{I}$. We then define the set of dominant labels

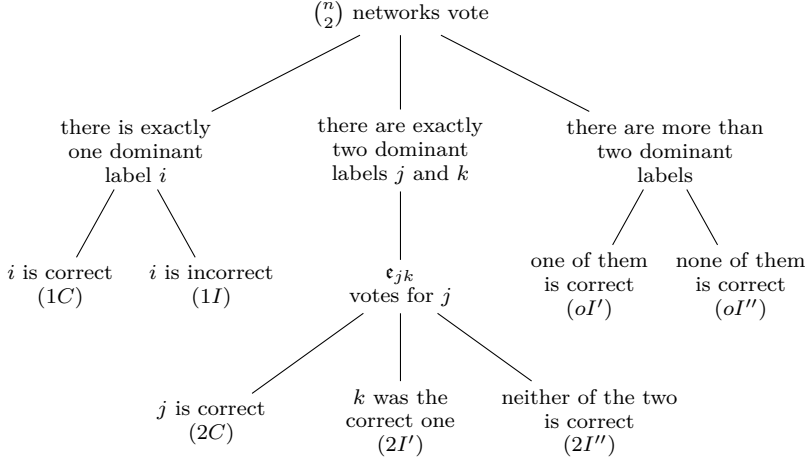
$$\mathcal{D} := \{b \in \mathcal{I} \mid b \text{ is a dominant label}\}.$$

Using this definition, we can have three possible outcomes.

- (a) There exists a label $i \in \mathcal{I}$ such that $\mathcal{D} = \{i\} \implies$ our input is labelled as i .
- (b) There exist $j, k \in \mathcal{I}$ such that $\mathcal{D} = \{j, k\} \implies$ our input is labelled as \mathbf{c}_{jk} .
- (c) There exist more than two dominant labels \implies our input is not classified.

While case (a) is straightforward, we can label our input even when we do not have a clear winner, that is, when we have trained a INN on the set of labels that are the most frequent (i.e. case (b)). Note that the proposed structured ensemble alongside its voting scheme can also be exploited for regular NNs.

Definition 4 (Label statuses). In our labelling system, when testing an input, seven different cases (herein called *label statuses*) can arise. The statuses names are of the form ‘number of the dominant labels + fairness of the prediction’. The first parameter can be 1, 2, or o , where o means ‘other cases’. The fairness of the prediction is C when it is correct, or I when it is incorrect. The superscripts related to I' and I'' only distinguish between different cases. These cases are described through the following tree diagram



where the status name is also reported.

The cases in which the classification algorithm classifies correctly are therefore only (1C) and (2C). Note that every input test will fall into exactly one label status.

Example 2. Let us take $\mathcal{I} = \{bird, cat, dog, frog\}$. Note that, in this case, we have to train $\binom{4}{2} = 6$ networks:

$$\mathcal{N}_{\{bird, cat\}}, \mathcal{N}_{\{bird, dog\}}, \mathcal{N}_{\{bird, frog\}}, \mathcal{N}_{\{cat, dog\}}, \mathcal{N}_{\{cat, frog\}}, \mathcal{N}_{\{dog, frog\}},$$

the first one distinguishes between *bird* and *cat*, the second one between *bird* and *dog*, and so on. A first input could have the following predicted labels:

$$\begin{aligned} \mathfrak{e}_{\{bird, cat\}} &= bird, & \mathfrak{e}_{\{bird, dog\}} &= bird, & \mathfrak{e}_{\{bird, frog\}} &= frog, \\ \mathfrak{e}_{\{cat, dog\}} &= cat, & \mathfrak{e}_{\{cat, frog\}} &= cat, & \mathfrak{e}_{\{dog, frog\}} &= dog. \end{aligned}$$

We would then have

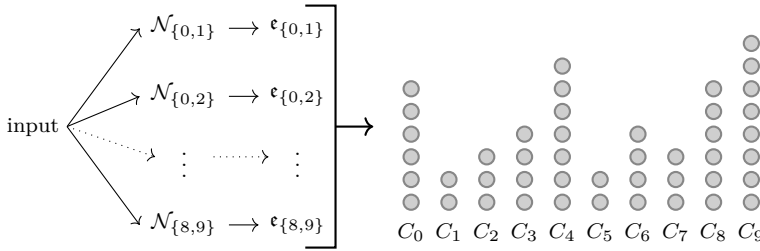
$$\begin{aligned} C_{bird} &= \{\{bird, cat\}, \{bird, dog\}\}, & C_{cat} &= \{\{cat, dog\}, \{cat, frog\}\}, \\ C_{dog} &= \{\{dog, frog\}\}, & C_{frog} &= \{\{bird, frog\}\}. \end{aligned}$$

In this case $\mathcal{D} = \{bird, cat\}$ because $|C_{bird}| = |C_{cat}| = 2 > 1 = |C_{dog}| = |C_{frog}|$ and we do not have a clear winner, but since $|\mathcal{D}| = 2$, we have trained a network that distinguishes between the two most voted labels, and so we use its output as our final predicted label, labelling our input as $\mathfrak{e}_{\{bird, cat\}} = bird$. If *bird* is the right label we are in label status (2C), if the correct label is *cat*, we are in label status (2I'). Else we are in label status (2I'').

Example 3. Let us take $\mathcal{I} = \{0, 1, \dots, 9\}$. Note that, in this case, we have to train $\binom{10}{2} = 45$ networks and that $|C_b| \leq 9$ for all $b \in \mathcal{I}$. Hence, an input could be labelled as follows:

$$\begin{aligned} C_0 &= (\{0, i\})_{i=1,2,3,5,7,8}, & C_1 &= (\{1, i\})_{i=5,6}, & C_2 &= (\{2, i\})_{i=1,5,8}, \\ C_3 &= (\{3, i\})_{i=1,2,4,5}, & C_4 &= (\{4, i\})_{i=0,1,2,5,6,7,9}, & C_5 &= (\{5, i\})_{i=6,7}, \\ C_6 &= (\{6, i\})_{i=0,2,3,7}, & C_7 &= (\{7, i\})_{i=1,2,3}, \\ C_8 &= (\{8, i\})_{i=1,3,4,5,6,7}, & C_9 &= (\{9, i\})_{i=0,1,2,3,5,6,7,8}. \end{aligned}$$

Visually, we can represent an input being labelled as above with the following scheme:



where we have omitted the name of each element of the set C_i for simplicity: for example, the dots above C_1 represent the sets $\{1, 5\}, \{1, 6\}$. Since $\mathcal{D} = \{9\}$, our input is labelled as 9. If 9 is the right label, we are in label status $(1C)$, if it is the wrong one, we are in label status $(1I)$. If instead $\hat{C}_j = C_j$, $j = 0, \dots, 7$, and

$$\hat{C}_8 = (\{8, i\})_{i=1,3,4,5,6,7,9}, \quad \hat{C}_9 = (\{9, i\})_{i=0,1,2,3,5,6,7},$$

then $|\hat{\mathcal{D}}| = |\{4, 8, 9\}| = 3$, so that our input was labelled as -1 . If the correct label is 4, 8 or 9, we are in label status (mI') , else we are in label status (mI'') . Lastly, if $\bar{C}_j = C_j$, $j \in \{0, 1, 2, 4, 5, 6, 7, 8\}$, and

$$\bar{C}_3 = (\{3, i\})_{i=1,2,4,5,9}, \quad \bar{C}_9 = (\{9, i\})_{i=0,1,2,3,5,6,7,8},$$

then $|\bar{\mathcal{D}}| = |\{4, 9\}| = 2$ and since $\{4, 9\} \in \bar{C}_4$ our input is labelled as 4. If 4 is the correct label, we are in label status $(2C)$, if 9 is the correct label, we are in label status $(2I')$, else we are in label status $(2I'')$. Note that, for brevity, in this example we used the notation $(\{j, i\})_{i=i_1, \dots, i_n} = \{\{j, i_1\}, \dots, \{j, i_n\}\}$, $j, i_1, \dots, i_n \in \{0, \dots, 9\}$.

2.5 Empirical Study

Having introduced the *BeMi* approach, we now undertake a series of six experiments in order to explore the following questions:

Dataset	classes	Input Dimension	Values	Training Set	Test Set
MNIST	10	28×28	Integers	60 000	10 000
Fashion-MNIST	10	28×28	Integers	60 000	10 000
Heart Disease	2	13	Continuous	$920 - x$	x

Table 2.1: Details of the different datasets exploited in the experiments.

- **Experiment 1:** What is the impact of a three-fold multi-objective model compared to a two-fold or single objective model? (Recall Section 2.4.)
- **Experiment 2:** How does the *BeMi* ensemble compare with the previous state-of-the-art MIP models for training BNNs in the context of few-shot learning?
- **Experiment 3:** How does the *BeMi* ensemble scale with the number of training images, considering two different types of BNNs?
- **Experiment 4:** How does the *BeMi* ensemble perform on various datasets, comparing the running time, the average gap to the optimal training MILP model, and the percentage of links removed?
- **Experiment 5:** What are the performance differences between a non-trivial INN and a BNN? Do INN exhibit particular weights distribution characteristics? A state-of-the-art comparison is also provided.
- **Experiment 6:** How does the *BeMi* ensemble perform on a continuous, low-dimension dataset, comparing BNNs and non-trivial INNs? Do INN exhibit the same weights distribution characteristics found in Experiment 5?

Datasets. Three datasets are adopted for the experiments. We use first the standard MNIST dataset [86] for a fair comparison with the literature, and second the larger Fashion-MNIST dataset [144]. For these two MNIST datasets, we test our results on 800 images for each class in order to have the same amount of test data for every class. Note that the MNIST dataset has 10 000 test data but they are not uniformly distributed over the 10 classes. For each experiment, we report the average over five different samples of images, i.e. we perform five different trainings and we report the average over them, while testing the same images. The images are sampled uniformly at random in order to avoid overlapping between different experiments. Beyond MNIST, we use the Heart Disease dataset [71] from the UCI repository. Table 2.1 summarizes the datasets.

Implementation details. As for the solver, we use Gurobi 10.0.1 [63] to solve our MILP models. The solver parameters are left to the default values if not specified otherwise. Apart from the first experiment, where we chose to consider every model equally, the fraction of time given to each step of the multi-objective model has been chosen accordingly to the importance of finding a feasible and robust solution. All the MILP experiments were run on an HPC cluster running CentOS, using a single node per experiment. Each node has an Intel CPU with 8 physical cores working at 2.1 GHz, and 16 GB of RAM. In all of our experiments concerning integer-value datasets, we fix the value $\epsilon = 0.1$. Notice that, because of the integer nature of the weights, of the image of the activation function, and of the data, setting ϵ equal to any number smaller than 1 is equivalent. When using continuous-value datasets, we fix the value $\epsilon = 1 \cdot 10^{-6}$ in accordance with the default variable precision tolerance of the Gurobi MILP solver we will use. The source code is available on GitHub [18].

Time limit management. Concerning the time limits for the different optimization models, the following choices have been made. In Experiment 1 and 6, the time limit is equally distributed between the three models to have a fair comparison. In Experiment 2, 3, 4, and 5, the majority of the imposed time limit was given to the first two models. The first model ensures feasibility of the whole pipeline and maximises the number of correctly classified images in the training phase, and it was considered important in the context of few-shot learning, since we do not have lots of images as training inputs. The second model was given a bigger time limit too because preliminary results, also shown in previous works, highlight the fact that the maximization of the margin ensures a better test accuracy with respect to the minimization of the links. In addition to this, the overall time limits have been chosen based on two criteria. Where comparisons with the literature are made, the selection ensures a fair comparison. In the remaining cases, the choice of time limit has been empirical, aiming to highlight the algorithm’s quality.

2.5.1 *Experiment 1*

The goal of the first experiment is to study the impact of the multi-objective model composed by **SM**, **MM** and **MW** with respect to the models composed by **SM** and **MM**, the one composed by **SM** and **MW**, and only **SM**, respectively.

The results refer to a BNN specialised in distinguishing between digits 4 and 9 of the MNIST dataset. These two digits were chosen because their written form can be quite similar. Indeed, among all ten digits, 4 and 9 are very often mistaken for each other, as it is shown in the confusion matrix in Appendix B.

The NN architecture consists of two hidden layers and has [784, 4, 4, 1] neurons. The architecture is chosen to mimic the one used in [132], that is [784, 16, 16, 10], but with fewer neurons. The number of training images varies

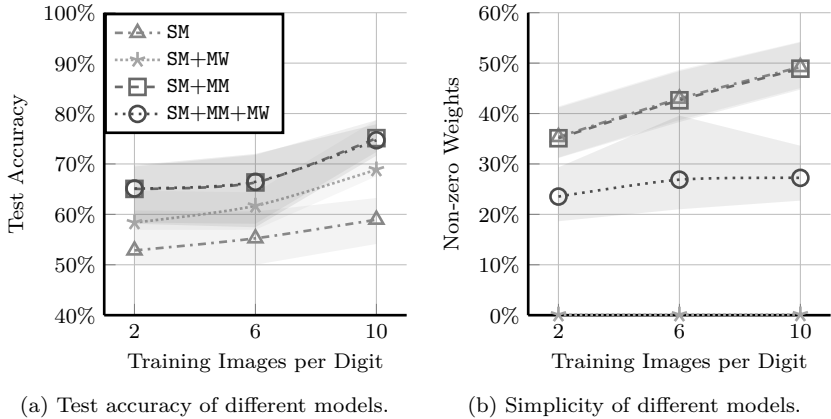


Figure 2.1: The **SM+MM+MW** achieve the same accuracy of the **SM+MM** model, outperforming the **SM** model, while having the smallest percentage of non-zero weights, a part from the **SM+MW** model, which has an almost-zero percentage of non-zero weights, but also a lower accuracy of the models that maximize the margins. The dotted lines represent the average accuracy obtained over 5 instances, while the shaded areas highlight the range between the minimum and maximum values.

between 2, 6 and 10, while the test images are 800 per digit, so 1600 in total. The imposed time limit is 30 minutes, equally distributed in the steps of each model: 30 minutes for the **SM**, 15 + 15 for the **SM+MW**, 15 + 15 for the **SM+MM**, and 10 + 10 + 10 for **SM+MM+MW**.

Figure 2.1a compares the test accuracy of the four hierarchical models, showing how the **MM** model ensures an increase in accuracy, while the **MW** allows the network to be pruned without performance being affected. Figure 2.1b displays the percentages of non-zero weights of the three trained models. Note that in this case the training accuracy is always 100% and so we did not add it to the plot. Also, dotted lines represent the average accuracy obtained over 5 instances, while the shaded areas highlight the range between the minimum and maximum values. This will be the case for every other plot if not specified otherwise. The **MW** step allows the number of non-zero weights to drop without accuracy being affected, hence resulting in an effective pruning. This behaviour is also observed for other couples of digits, even the ones that are easier to distinguish, namely, 1 and 8. Results of this experiment are reported in Appendix B.

Based on these reasons, for the remaining experiments, we will exclusively employ the model that incorporates all three steps.

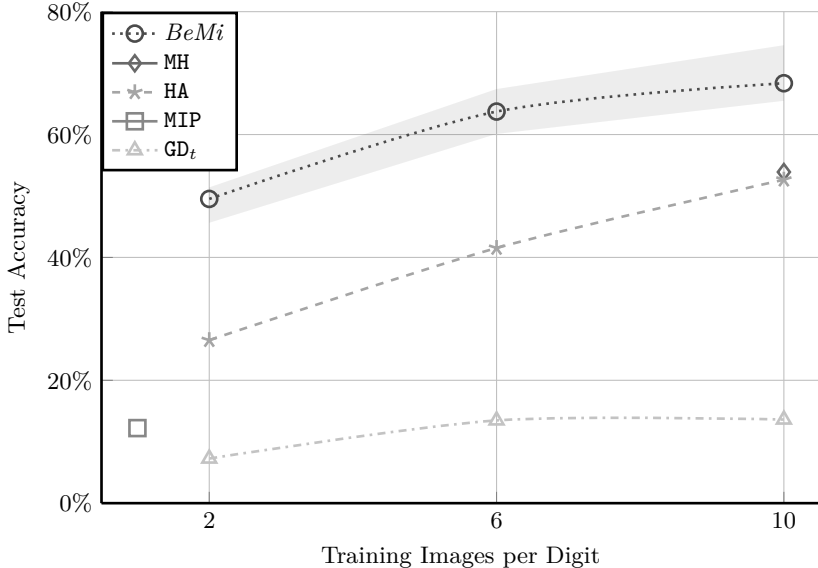


Figure 2.2: Comparison of published approaches vs *BeMi*, in terms of accuracy over the MNIST dataset using few-shot learning with 2, 6, and 10 images per digit.

2.5.2 Experiment 2

The goal of the second experiment is to compare the *BeMi* ensemble with the following state-of-the-art methods: the pure MIP model in [132]; the hybrid CP-MIP model based on **Max-Margin** optimization (HA) [132]; the gradient-based method GD_t introduced in [68] and adapted in [132] to deal with link removal; and the **Min-hinge** (MH) model proposed in [129]. For the comparison, we fix the setting of [132], which takes from the MNIST up to 10 images for each class, for a total of 100 training data points, and which uses a time limit of 7 200 seconds to solve their MIP training models.

In our experiments, we train the *BeMi* ensemble with 2, 6 and 10 samples for each digit. Since our ensemble has 45 BNNs, we leave for the training of each single BNN a maximum of 160 seconds (since $160 \cdot 45 = 7\,200$). In particular, we give a 75 seconds time limit to the solution of **SM**, 75 seconds to **MM**, and 10 seconds to **MW**. In all of our experiments, whenever the optimum is reached within the time limit, the remaining time is added to the time limit of the subsequent model. We remark that our networks could be trained in parallel, which would highly reduce the wall-clock runtime. For the sake the completeness, we note that we are using $45 \cdot (784 \cdot 4 + 4 \cdot 4 + 4 \cdot 1) = 142\,020$

parameters (all the weights of all the 45 BNNs) instead of the $784 \cdot 16 + 16 \cdot 16 + 16 \cdot 10 = 12\,960$ parameters used in [132] for a single large BNN. Note that, in this case, the dimension of the parameter space is $3^{12\,960} (\cong 10^{6\,183})$, while, in our case, it is $45 \cdot 3^{3\,156} (\cong 10^{1\,507})$. In the first case, the solver has to find an optimal solution between all the $10^{6\,183}$ different parameter configurations, while with the *BeMi* ensemble, the solver has to find 45 optimal solutions, each of which lives in a set of cardinality $10^{1\,507}$. This significantly improves the solver performances. We remark that a parameter configuration is given by a weight assignment $\hat{W} = (\hat{w}_{ijl})_{ijl}$ since every other variable is uniquely determined by \hat{W} .

Figure 2.2 compares the results of our *BeMi* ensemble with the four other methods presented above. Note that the pure MIP model in [132] can handle a single image per class in the given time limit, and so only one point is reported, and note also that for the minimum hinge model *MH* presented in [129] only the experiment with 10 digits per class was performed. We report the best results reported in the original papers for these four methods.

The *BeMi* ensemble obtains an average accuracy of 68%, outperforms all other approaches when 2, 6 or 10 digits per class are used. Note that our method attains 100% accuracy on the training set, that is, the *SM* model correctly classifies all the images. In this case, the first model is not needed to ensure feasibility, but it serves mainly as a warm start for the *MM* model.

2.5.3 Experiment 3

The goal of the third experiment is to study how our approach scales with the number of data points (i.e. images) per class, and how it is affected by the architecture of the small BNNs within the *BeMi* ensemble. For the number of data points per class, we use 10, 20, 30, 40 training images per digit. We use the layers $\mathcal{N}_a = [784, 4, 4, 1]$ and $\mathcal{N}_b = [784, 10, 3, 1]$ for the two architectures. While the first architecture is chosen as to be consistent with the previous experiments, the second one can be described as an integer approximation of $[N, \log_2 N, \log_2(\log_2 N), \log_2(\log_2(\log_2 N))]$. Herein, we refer to Experiments 3a and 3b as the two subsets of experiments related to the architectures \mathcal{N}_a and \mathcal{N}_b . In both cases, we train each of our 45 BNNs with a time limit of 290s for model *SM*, 290s for *MM*, and 20s for *MW*, for a total of 600s (i.e. 10 minutes for each BNN).

Figure 2.3a shows the results for Experiments 3a and 3b: the dotted and dashed lines refer to the two average accuracies of the two architectures, while the coloured areas include all the accuracy values obtained as the training instances vary. The two architectures behave similarly and the best average accuracy exceeds 81%.

Table 2.2 reports the results for the *BeMi* ensemble where we distinguish among images classified as correct, wrong, or unclassified. These three conditions refer to different label statuses specified in Definition 4: the correct labels are the sum of the statuses (1C) and (2C); the wrong labels of statuses

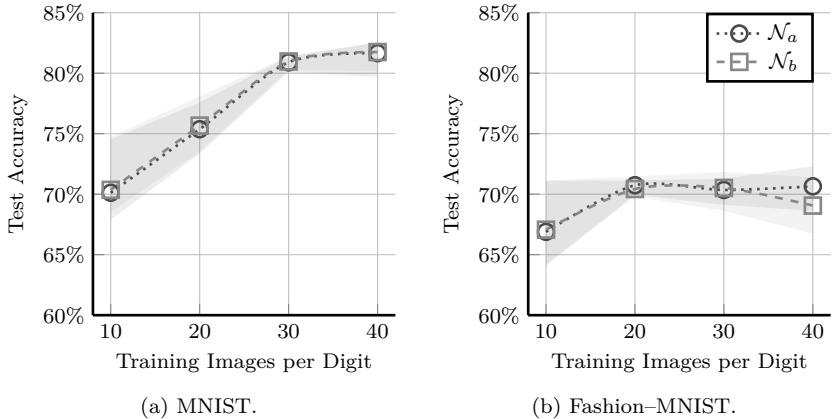


Figure 2.3: Average accuracy for the *BeMi* ensemble tested on two architectures, namely $\mathcal{N}_a = [784, 4, 4, 1]$ and $\mathcal{N}_b = [784, 10, 3, 1]$, using 10, 20, 30, 40 images per class.

Table 2.2: Percentages of MNIST images classified as correct, wrong, or unclassified (*n.l.*) for the architecture $\mathcal{N}_a = [784, 4, 4, 1]$. The unclassified images are less than 2.31%.

Images per class	Classification (%)		
	correct	wrong	<i>n.l.</i>
10	70.12	27.65	2.23
20	75.37	22.32	2.31
30	80.90	17.46	1.64
40	81.66	16.68	1.66

($2I'$), ($2I''$), and ($1I$); the unclassified labels (*n.l.*) of (oI') and (oI''). Notice that the vast majority of the test images have only one dominant label, and so falls into statuses ($1C$) or ($1I$). The unclassified images are less than 2.31%. These results are reported in Table 2.3.

2.5.4 Experiment 4

The goal of the fourth experiment is to revisit the questions of Experiments 3a and 3b with the two architectures \mathcal{N}_a and \mathcal{N}_b , using the more challenging Fashion-MNIST dataset.

Figure 2.3b shows the results of Experiments 3a and 3b. As in Figure 2.2, the dotted and dashed lines represent the average percentages of correctly classified images, while the coloured areas include all accuracy values obtained as the instances vary. For the Fashion-MNIST, the best average accuracy

Table 2.3: Percentages label statuses of MNIST images for the architecture $\mathcal{N}_a = [784, 4, 4, 1]$. The vast majority of the test images have only one dominant label, and so falls into statuses $(1C')$ or $(1I')$.

Images per class	Label status (%)						
	$(1C')$	$(1I')$	$(2C')$	$(2I')$	$(2I'')$	(oI')	(oI'')
10	68.43	24.53	1.69	1.21	1.91	1.88	0.35
20	73.79	19.33	1.58	1.39	1.60	2.02	0.29
30	79.64	15.01	1.26	1.25	1.20	1.46	0.18
40	80.34	14.36	1.32	1.09	1.23	1.45	0.21

Table 2.4: Aggregate results for Experiments 2 and 3: the 4-th column reports the runtime to solve the first model **SM**; *Gap (%)* refers to the mean and maximum percentage gap at the second MILP model **MM**; *Links (%)* is the percentage of non-zero weights after the solution of models **MM** and **MW**.

Dataset	Layers	Images per class	SM time (s)	Gap (%)		Links (%)	
				mean	max	(MM)	(MW)
MNIST	784,4,4,1	10	3.00	12.06	20.70	49.13	29.21
		20	6.47	14.81	22.18	54.70	28.41
		30	10.60	16.04	24.08	56.44	30.46
		40	15.04	15.98	26.22	57.92	29.27
	784,10,3,1	10	6.04	4.52	7.37	49.28	24.72
		20	15.01	5.46	8.40	54.97	24.40
		30	22.68	5.92	11.00	56.73	26.96
		40	33.97	6.21	20.87	58.29	24.66
F-MNIST	784,4,4,1	10	4.83	13.34	26.48	87.75	58.76
		20	9.77	14.05	28.91	90.73	59.97
		30	36.10	19.95	136.33	92.41	58.12
		40	72.15	30.36	333.70	93.57	59.46
	784,10,3,1	10	11.42	4.87	9.14	87.90	51.57
		20	21.46	5.11	9.86	91.05	52.29
		30	35.12	6.30	40.07	92.78	52.62
		40	52.37	8.99	56.14	94.01	53.38

exceeds 70%.

Table 2.4 reports detailed results for all Experiments 2 and 3. The first two columns give the dataset and the architecture, the third column specifies the number of images per digit used during training. The 4-th column reports the runtime for solving model **SM**. Note that the time limit is 290 seconds; hence, we solve exactly the first model, consistently achieving a training accuracy of 100%. The remaining four columns give: *Gap (%)* refers to the mean and maximum percentage gap at the second MILP model (**MM**) of our lexicographic multi-objective model, as reported by the Gurobi **MIPgap** attribute; *Links (%)* is the percentage of non-zero weights after the solution of the second model **MM**, and after the solution of the last model **MW**. The results show that the

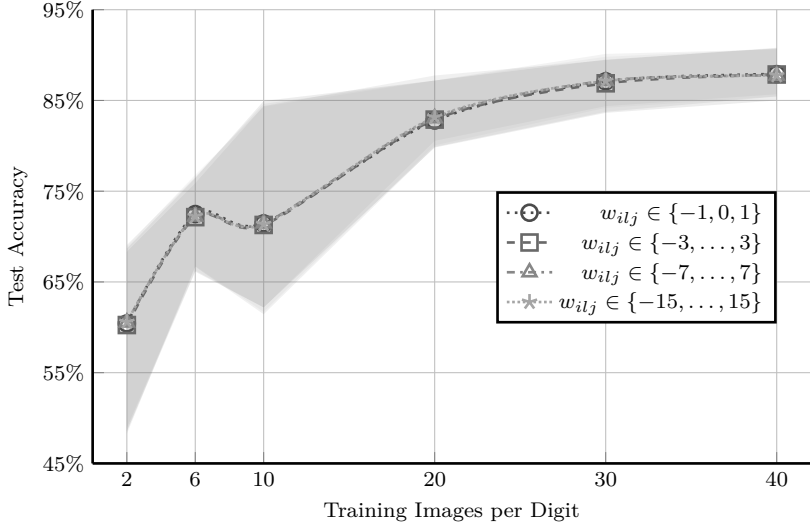


Figure 2.4: Comparison of accuracy for different values of P . Note how using an exponentially larger research space, namely, using values of P greater than 1, does not improve the accuracy.

runtime and the gap increase with the size of the input set. However, for the percentage of removed links, there is a significant difference between the two datasets: for MNIST, our third model **MW** removes around 70% of the links, while for the Fashion-MNIST, it removes around 50% of the links. Note that in both cases, these significant reductions show how our model is also optimizing the BNN architecture. Furthermore, note that even if the accuracy of the two architectures is comparable, the total number of non-zero weights of \mathcal{N}_a is about half the number of non-zero weights of \mathcal{N}_b .

2.5.5 Experiment 5

The goal of the fifth experiment is to compare the performances of BNNs and non-trivial INNs. The results refer to five different runs of an INN of architecture $[784, 4, 4, 1]$ specialised in distinguishing between digits 4 and 9 of the MNIST dataset. The number of training images varies between 2, 6, 10, 20, 30, and 40, while the test images are 800 per digit, so 1600 in total. The imposed time limit is 290s + 290s + 20s.

As Figure 2.4 shows, different INNs are comparable not only in the average accuracy, represented by the dotted lines, but also in the maximum and minimum accuracy, reported by the shaded areas.

In order to study why different values of P lead to comparable accuracy,

Table 2.5: Weights distributions of the INNs whose accuracy is depicted in Figure 2.4. The column $w = -P$ indicates the percentage of weights that are equal to $-P$, and so on. The networks have different values but similar extremal weights distributions, where less than 2% of the weights attain a value that is not P , $-P$, or zero, indicated as *others*.

P	Images per class	$w = -P$	$w = 0$	$w = P$	<i>others</i>
1	2	4.06	73.66	22.28	-
	6	4.75	73.70	21.55	-
	10	5.15	74.02	20.83	-
	20	7.12	61.31	31.57	-
	30	6.21	74.89	18.90	-
	40	13.16	64.98	21.86	-
3	2	3.86	75.91	20.15	0.08
	6	4.60	73.46	21.63	0.31
	10	6.25	73.50	19.72	0.53
	20	12.48	69.58	17.19	0.75
	30	13.06	74.00	11.89	1.05
7	40	13.99	65.51	19.31	1.19
	2	3.90	75.89	20.10	0.11
	6	4.46	73.40	21.76	0.38
	10	8.82	68.00	22.60	0.58
	20	12.46	65.56	21.06	0.92
15	30	14.99	73.82	9.95	1.24
	40	18.05	70.90	9.40	1.65
	2	3.69	75.89	20.30	0.12
	6	4.68	73.31	21.64	0.37
	10	5.53	73.28	20.61	0.58
	20	7.43	69.34	22.19	1.04
	30	13.76	67.64	17.23	1.37
	40	17.34	70.86	10.13	1.67

we report the weights distributions of the INNs whose accuracy is depicted in Figure 2.4. Table 2.5 highlights not only the role of the MW model but also the extreme-valued nature of the distributions. In fact, it can be seen that apart from the percentage of weights set to zero, the vast majority of the remaining weights have a value of either P or $-P$, with less than 1.67% of the weights attaining one of the other intermediate values. We remark that this type of phenomenon is recurrent in the literature under the name of magnitude pruning, see [64, 101, 25]. In our setting, such a phenomenon occurs spontaneously.

2.5.6 Experiment 6

The goal of the sixth experiment is to study the impact of the multi-objective model and the weight distributions over a different dataset, namely the Heart

Table 2.6: Average accuracy and weight distribution for the Heart Disease dataset. The column Average accuracy depicts the average percentage of correctly classified test data. The column $w = -P$ indicates the percentage of weights that are equal to $-P$, and so on. For each value of P , the best result in terms of accuracy with respect to the model is written in bold. Notice that in the majority of the cases, the multi-objective function performs better than the single-objective one. Notice also that even if the percentage of the non-zero and non-extremal weights, indicated as *others*, is higher than the one obtained with the MNIST dataset, the distribution is still not uniform.

Models	P	Average accuracy	$w = -P$	$w = 0$	$w = P$	<i>others</i>
SM	1	74.00	35.43	45.14	19.43	-
	3	75.00	21.43	27.43	16.57	34.57
	7	77.00	27.43	14.86	16.86	40.85
	15	77.00	18.86	11.71	12.86	56.57
SM + MM	1	75.00	14.57	18.29	67.14	-
	3	75.50	10.57	13.43	63.14	12.86
	7	73.50	12.29	9.43	52.57	25.71
	15	78.50	11.71	5.71	46.86	35.72
SM + MM + MW	1	75.50	9.43	27.43	63.14	-
	3	76.00	5.43	25.43	57.43	11.71
	7	73.50	7.43	18.57	49.71	24.29
	15	78.50	8.57	13.43	45.71	32.29

Disease Dataset by [71].

Table 2.6 reports the average accuracy and weights distributions of this experiment. The average was performed over 5 instances, and for each instance 200 data samples were used, where 80% was used for the training and 20% was used for the test. All the networks have the same architecture, namely [13, 5, 1], and each network’s imposed time limit is 60 minutes, equally distributed in the steps of each model: 60 minutes for the SM, 30 + 30 for the SM+MM, and 20 + 20 + 20 for SM+MM+MW.

2.6 Conclusion and Future Work

This work introduced the *BeMi* ensemble, a structured architecture of INNs for classification tasks. Each network specializes in distinguishing between pairs of classes and combines different approaches already existing in the literature to preserve feasibility while being robust and simple. These features and the nature of the parameters are critical to enabling neural networks to run on low-power devices. In particular, binarized NNs can be implemented using Boolean operations and do not need GPUs to run.

The output of the *BeMi* ensemble is chosen by a majority voting system that generalizes the one-versus-one scheme. Notice that the *BeMi* ensemble

is a general architecture that could be employed using other types of neural networks.

An interesting conclusion from our computational experiments is a counterintuitive result: that the greater flexibility in the search space of INNs does not necessarily result in better classification accuracy compared to BNNs. We find it noteworthy that our computational evidence supports the idea that simpler BNNs are either superior or equal to INNs in terms of accuracy.

A current limitation of our approach is the strong dependence on the randomly sampled data used for training. In future work, we plan to improve the training data selection by using a k -medoids approach, dividing all images of the same class into disjoint non-empty subsets and consider their centroids as training data. This approach should mitigate the dependency on the sampled training data points.

Second, we also plan to better investigate the scalability of our method with respect to the number of classes of the classification problem training fewer BNNs, namely, one for every $\mathcal{J} \in \mathcal{Q} \subset \mathcal{P}$, with $|\mathcal{Q}| \ll |\mathcal{P}|$. Besides the datasets we exploited, in future, we intend to investigate datasets more appropriate for the task of few-shot learning [32].

Third, another possible future research direction is to exploit the generalisation of our ensemble, allowing to have networks which distinguishes between m classes instead of 2, where the total number of classes of the problem is $n \gg m$. Note that the definitions of this generalization are presented in Appendix B.

Fourth, an interesting future research direction regards stochastic/robust optimization and scenario generation, in the following sense. In the case of the MNIST/FashionMNIST for example, the images can be seen as samples of many unknown probability distributions, one for each class: if one would like to train a neural network with a MIP, using few images, the selection of these samples with which the training is performed is crucial, and so the study of this problem from a stochastic point of view could lead to interesting results.

STOCHASTIC OPTIMIZATION: SCHEDULING OF INPATIENT AND OUTPATIENT SURGERIES

3.1 Introduction

With the advancement of surgery and anesthesiology in recent years, surgical clinical pathways have changed significantly, with an increase in outpatient surgeries [110]. While inpatients are admitted to the hospital by planning an overnight and a certain Length-of-Stay (LoS) in the ward, outpatients require low-complexity surgery and have a state of health that allows them to be discharged within a few hours, except for complications. Although the main difference between inpatient and outpatient care lies in the duration of the stay after the end of the surgery, these two classes of patients have also different characteristics in terms of variability, resources, and needs. For instance, while inpatients have a higher uncertainty in the surgery duration due to the average more complex procedures, outpatients register a higher rate of late cancellations or no-shows. Consequently, in operational contexts dealing with both types of patients by sharing the same Operating Rooms (ORs), these differences should be considered to ensure an adequate outcome and throughput when the surgical schedule is determined.

The surgical scheduling problem is defined as the ensemble of decisions about surgical procedures to be executed in the operating theater, the resource allocation for their execution, and their sequencing within a certain planning horizon [96]. This problem lies at the lowest of three decision levels that can be identified in the whole decision process concerning OR planning and scheduling. At the strategic level, a case mix (i.e. a set of specialties) is defined and a pool of ORs is assigned to it. Then, the tactical level concerns the planning of the OR blocks, which is the decision about which ORs have to be opened in the days of the planning horizon, and the definition of a Master Surgery Schedule (MSS) that assigns OR blocks to specialties. Finally, the schedule of the surgeries and their actual execution are managed at the operational level. A recent literature review of the OR Planning & Scheduling [65] highlighted the high number of aspects to be considered at the three decision levels, showing an increasing complexity of the modeling in terms of considered decision problems in the studies published in the last years.

From an elective patient perspective [122, 128], a MSS is given as input and the surgical scheduling is solved by addressing two main subproblems, namely the advance scheduling and the allocation scheduling. Under the adoption of

the common block-scheduling strategy, which is the most common setting of the real-world operating theaters [11], the advance scheduling consists of an *assignment procedure*, in which patients are selected from the waiting list and assigned to the OR blocks of the planning horizon. In dealing with the assignment procedure, several aspects should be taken into consideration, such as the urgency of the patient from a clinical point of view, the time already spent on the waiting list, and the resources required for the surgery execution compared to those available. The most critical resource in both management and expenditure senses is operating time: given a limited OR capacity, surgeries have to be scheduled in accordance with their Estimated Operating Time (EOT), that is a duration estimated by a physician in accordance with the surgery procedure that has to be performed. At the same time, decision-makers should consider that uncertainty factors could lead to a different realization of the surgical procedures with respect to the planned one. Common practices to alleviate the negative effects of uncertainty are to reserve slack times that make the schedule more robust [115, 135, 137] or to create tailored mixes of patients scheduled in the same OR blocks [2, 141].

The allocation scheduling includes a *sequencing procedure* and a *timing procedure*: in the former the order in which the surgeries have to be executed is decided, while the latter determines a start time for each surgery, that is the moment from which the patient will be available to be operated on. The timing procedure includes the management of possible slack time, which can be distributed between subsequent surgeries and/or allocated at the end of the OR block to deal with the negative impact of uncertainty.

Addressing the three procedures, multiple criteria can be defined to determine the optimal global schedule. From a patient-centered point of view, important objectives to be minimized are (i) the scheduling costs related to patient urgency and indirect waiting times (days spent on the waiting list) expressed as a penalty when their surgery is not assigned to any OR block, (ii) the direct waiting time costs as a function of the time elapsed between the planned and the actual start times, and (iii) the cancellation costs consisting of penalties for surgery cancelled for reasons that do not depend on the patients. As claimed by Wang et al. [142], studies focusing on inpatient and outpatient settings give different importance to these objectives. While the indirect waiting time is more relevant for inpatients, the direct waiting time has a greater interest in outpatients. In addition, a cancellation of an inpatient surgery would have a negative impact on the occupation of resources in the related ward due to the possible growth of the LoS. From an efficiency perspective, two further important objectives to be minimized are (iv) the idle time, in order to avoid wasting resources and the consequent lengthening of the waiting list, and (v) the overtime, which allows the hospital to contain extra costs. Since the OR Planning & Scheduling literature stressed the strong trade-off among all objectives [4, 7, 37, 117, 135, 142, 149], a challenging task for the decision maker is to find a good balancing.

Furthermore, we can identify three factors of uncertainty of paramount importance: (a) the deviation between the EOT and the actual surgery duration, called Real Operating Time (ROT), (b) the possible insertion of emergency patients within the planned OR blocks, and (c) the patients' no-show. To the best of our knowledge, the surgical scheduling problem has never been solved under objectives (i)–(v) and three uncertainty factors (a)–(c) simultaneously.

In this work, we present a stochastic optimization approach to investigate the potential of chance-constrained and stochastic programming to efficiently exploit the ORs and to guarantee the management of heterogeneous groups of patients, that is patients with different characteristics (e.g. ROT distribution, no-show rate) and needs (scheduling, waiting time, and cancellation costs). Although no-shows are unpredictable, the rationale behind the choice of considering this factor of uncertainty is that such a phenomenon drastically impacts on idle time, and we guess it could be balanced with (or relieved by) other uncertainty components, such as the arrival of emergency patients or surgery duration longer than the expectation. Another contribution of our study is the development of a flexible decision support tool that can be adopted by practitioners from different operative contexts. According to [142], a deeper investigation should be made about the simultaneous optimization of: the inpatient and outpatient patient flows, characterized by different characteristics (e.g. predictability of the ROT, no-show rates, and costs); the direct and indirect waiting times of outpatients; elective and non-elective patient flow considering patients' no-shows. All these research questions can be addressed through a flexible optimization approach as the one proposed in this work.

The contribution of this work is three-fold. Firstly, from a modeling point of view, we formalize the advance scheduling problem through a new Chance-Constrained Integer Programming (CCIP) model and the allocation scheduling problem through a two-stage Stochastic Mixed Integer Programming (SMIP) model under objectives (i)–(v) and uncertainty factors (a)–(c). Secondly, from a methodological point of view, we propose two new alternative heuristics for the allocation scheduling to the standard Monte Carlo sampling: a SMIP-based approach and a genetic algorithm with a custom encoding. Lastly, we provide a computational analysis that shows the effectiveness of both approaches depending on the instance size, and we analyze and discuss obtained results, providing the reader with managerial insights concerning inpatients and outpatients scheduling.

The chapter is organized as follows. Section 3.2 reports the state of art about the surgical scheduling problem that consider at least two of the three mentioned decision procedures. In Section 3.3, we present the problem statement and the two-phase stochastic optimization framework. In Section 3.4, we propose two novel mathematical programming models for the advance and allocation scheduling. In Section 3.5 we present solution approaches based on Monte Carlo sampling and a metaheuristic to solve the two models. After presenting the experimental setup in Section 3.6, the proposed approaches are

analyzed in Section 3.7 and used to provide general insights for operational contexts in which inpatients and outpatients are scheduled within the same operating theater. In Section 3.8, we draw conclusions and further research directions.

3.2 Literature review

Under deterministic settings, several approaches can be found in the literature considering assignment, sequencing, and timing procedures simultaneously. Roshanaei et al. [114] propose a method based on a branch-and-check decomposition to deal with strategical decisions, tactical decisions, and all three operational procedures simultaneously, maximizing the OR utilization and proposing the consideration of stochasticity as a future research direction. Jebali et al. [72] present a two-step approach to deal with the assignment procedure and the sequencing procedure, also considering pre-operative and post-operative resources, with the purpose of optimizing idle time, overtime, and direct waiting time. Marques et al. [95] introduce a genetic algorithm to find a near-optimal solution of the assignment procedure and the sequencing procedure, deciding at the same time the assignment of ORs to specialties, in order to maximize OR utilization and number of scheduled patients.

The inclusion of stochasticity in the decision problem increases its complexity. To the best of our knowledge, no work optimizes all three procedures of the surgery scheduling problem in a unified approach. We can identify two main categories of prior articles that take into account two procedures among the assignment procedure, the sequencing procedure, and the timing procedure. The first category concerns studies dealing only with the allocation scheduling, by considering the assignment of patients to OR blocks as an input to optimize both the sequence and the starting times of the surgeries [39, 43, 87, 93, 145]. The main objective of papers belonging to this category are the minimization of direct waiting times and the minimization of costs related to overtime and the occupation of the medical staff. Several works take into account constraints about the allocation of physical or human resources for activities concerning anesthesia [39] and post-surgery [87]. The second category includes works considering both the advance scheduling and the allocation scheduling, focusing only on the sequencing of the surgeries, without optimizing the starting times. Generally, from a computational complexity perspective, this requires a higher effort because the combination of assignment with sequencing considerably widens the search space. For this reason, we assume that the coordination of pre-operative and post-operative activities can be done *ex post* without limiting the admissible solutions.

In Table 3.1 we summarize the papers that lie in this last category, as well as our work, to provide a comparison and to highlight the novelties of this work.

A first stream of works addresses the problem with approaches based on

Table 3.1: Summary of prior works that deal with both the advance scheduling and the allocation scheduling under uncertainty. AP, SP, and TP columns indicate if the assignment, sequencing, and timing procedures are addressed, respectively.

Work	AP	SP	TP	Other decisions	Uncert.	Objectives	Methodology
Batun et al. [11]	✓	✓	×	· OR blocks to be opened · physician assignment	· ROTs	· overtime · idle time · financial costs	· SMP
Landa et al. [84]	✓	✓	×	· overtime allocation	· ROTs	· OR utilization · cancellations	· SMP · CCIP · metaheuristics
Testi et al. [128]	✓	✓	×	· MSS	· ROTs	· overtime · OR utilization · throughput · bed utilization	· DES · ILP · heuristics
Duna et al. [45]	✓	✓	×	· real-time management	· ROTs	· overtime · OR utilization · throughput · cancellations · ind. waiting time · due date violation	· DES · metaheuristics · online
Duna et al. [46]	✓	✓	×	· emergency OR policy · real-time management	· ROTs · emergency	· overtime · OR utilization · throughput · cancellations · ind. waiting time · due date violation	· DES · metaheuristics · online
Wang et al. [141]	✓	✓	×	· surgery partitioning	· ROTs · emergency	· overtime · idle time · OR utilization · throughput · cancellations · ind. waiting time · due date violation	· DES
Agrawal et al. [2]	✓*	✓	✓		· ROTs	· idle time · dir. waiting time	· SMP · heuristics
This work	✓	✓	✓		· ROTs · emergency · no-shows	· overtime · idle time · cancellations · ind. waiting time · dir. waiting time	· SMP · CCIP · metaheuristics

stochastic programming. Batun et al. [11] propose a two-stage SMIP model for the assignment procedure and the sequencing procedure, which are addressed jointly with the number of OR blocks to be used, adopting a pooling strategy for a flexible assignment of the OR blocks. The physician-patient assignment is taken into account, the considered aspect of uncertainty is the surgery duration, and only facility-centered objectives are defined, that is overtime, idle time, and OR financial costs. The authors claim that the L-shaped method fails for realistic instances and they present several structural properties of the SMIP model that lead to computational advantages. Landa et al. [84] introduce a two-stage CCIP model, where the assignment procedure is solved at the first stage by defining the OR utilization as objective and by fixing a maximum probability of needing overtime in each OR block, then the sequencing procedure and the overtime allocation are performed by the second stage model by minimizing the number of cancellations. Because of the computational complexity of the two stochastic optimization problems, the authors provide a two-phase metaheuristic based on neighborhood search and Monte Carlo simulation.

Another stream of research combine Discrete Event Simulation (DES) with deterministic optimization and/or online optimization approaches. Testi et al. [128] address the assignment procedure and the sequencing procedure in the last two phases of their three-phase optimization approach, based on an Integer Linear Programming (ILP) model and three greedy heuristics, respectively. After providing a solution for the MSS and the assignment procedure, a DES model is used to analyze the impact of three simple heuristics for the timing procedure, observing indices such as throughput, overtime, idle time, and bed utilization. DES is also proposed by Duma et al. [45] for evaluating the impact of a deterministic metaheuristic for the assignment procedure, when it is used jointly with several sequencing policies and online optimization algorithms. Such an approach focuses on the problem of monitoring the actual execution of the surgery, which requires making real-time decisions to deal with the uncertainty of surgery durations and the consequent dynamicity of the operating theater. The authors compare different configurations of their approach with respect to several performance measures, such as the fraction of patients operated within the time limit, throughput, idle time, overtime, and cancellations. The authors provide a more general analysis in [46] by proposing online algorithms for the insertion of emergency patients within the operating theater and by evaluating the impact of operating them in dedicated, flexible, or hybrid ORs. Both the advance scheduling and the allocation scheduling are also considered by Wang et al. [141], which propose a DES model to analyze the impact of two alternative policies when dealing with inpatients having different levels of surgery duration variability, that is the pooling of surgeries with more or less predictable ROTs or the partitioning of the ORs to be assigned to different groups of patients. The arrival of emergency patients to be operated on within a short time limit has been considered, as well as the

impact of the proposed policies on multiple criteria. The authors conclude that partitioning patient into two groups reduces the indirect waiting time of elective patients and increases the OR utilization, at the cost of a slight worsening of the cancellation rate and the emergency patients' waiting time. A recent work by Agrawal et al. [2] is placed outside this classification, as it combines the sequencing procedure and the timing procedure with the decision of assigning surgeries to OR blocks. Although this decision falls within the assignment procedure, it does not include the selection of patients within the waiting list since it is given as input. The authors formulate the problem with a SMIP model with an objective function that includes penalties for idle time and direct waiting time, and with the surgery duration as a factor of uncertainty. Due to its complexity, the problem is addressed with heuristic approaches for the patient-OR assignment and the sequencing procedure based on a prioritization that depends on the standard deviation of the ROTs, then they use a Monte Carlo simulation for the timing procedure.

In general, most of the prior studies formulate the advance and/or allocation scheduling with stochastic or robust optimization models. While stochastic optimization is more indicated when the probability distribution used to model the uncertainty is known and reliable, robust optimization is suggested when true distributions are not available. In order to deal with conservative solutions provided by robust optimization, distributionally robust optimization models are proposed in [121], providing robust patient scheduling over an ambiguity set built on little information about the surgical durations, such as mean and variance. We remark that distributionally robust optimization is recommended when dealing with poor historical data or with rare surgical procedures. However, with the rise of healthcare data accessibility in the last decade, it is possible to incorporate surgical time variability into OR scheduling effectively (e.g. see [9]). Thus, we assume that sufficient information is available to define the surgery duration distribution of the surgical procedures under consideration. Firstly, approximation methods based on Monte Carlo sampling [92] is proposed to solve the CCIP model for the advance scheduling and the SMIP model for the allocation scheduling. When solving the latter with the Monte Carlo sampling, that is the Sample Average Approximation (SAA), computational complexity issues arise when the number of patients scheduled within the same OR block increases. Therefore, we propose two heuristic approaches for solving the allocation scheduling. The first is called N -fold SAA and consists of computing the SAA over a partition of the sample into N folds. The second is a genetic algorithm, which is a successful methodology in stochastic combinatorial optimization problems arising in various contexts [21, 57], for which we introduce by introducing a custom encoding and a fitness function computed through a Monte Carlo simulation.

3.3 Problem statement

Let us consider a set of specialties S and a MSS fixed at the tactical level. This means that we have a set J of ORs assigned to the specialties $s \in S$ over a certain set of days K , which is the planning horizon (e.g. $K = \{1, 2, 3, 4, 5\}$ for a workweek). Then, the set of all the OR blocks $B \subseteq J \times K$ under consideration can be partitioned in $|S|$ subsets B_s assigned to the different specialties $s \in S$. OR blocks are indicated with an ordered pair $(j, k) \in B_s$, where j is the identifier of an OR assigned to the specialty s on the k -th day of the planning horizon. From the starting time of an OR block in B_s , the block has an ordinary duration L_{jk} to operate on patients. Let W be the set of all elective patients to be scheduled, namely the waiting list, then $W = \bigcup_{s \in S} W_s$, where W_s is the set of elective patient of the specialty s . For every specialty s , we have to select a subset of surgeries in W_s and assign them to a specialty's OR block. OR blocks can use additional time with respect to the planned ending, that is a limited overtime is available with a certain economic cost.

For each patient $i \in W = \bigcup_{s \in S} W_s$, some information is known and should be taken into account when the OR schedule is defined: the surgical procedure group, the EOT μ_i , the scheduling cost c_i^{sched} (e.g. the ratio of the waiting time over the maximum time before treatment [46, 134]), the waiting time cost c_i^{wait} (per minute), and the cancellation cost c_i^{canc} . During the definition of the surgical schedule, the decision maker can take into account several estimators about stochastic aspects concerning the surgery of the patient i , such as the mean (i.e. approximately the average duration of their surgical procedure [106]) and the standard deviation σ_i , and the probability of no-show r_i , that is the no-show rate of previous patients of the same specialty or a general group that could be mined from historical data.

For each specialty s and for each OR block $(j, k) \in B_s$, the surgical scheduling defines:

Advance scheduling: the set $I_{jk} \subseteq W_s$ of scheduled patients (assignment procedure);

Allocation scheduling: the sequence in which the patients $i \in I_{jk}$ are scheduled within the OR block (sequencing procedure), and the scheduled start time of each patient $i \in I_{jk}$ (timing procedure).

In this study, we deal with the advance scheduling and the allocation scheduling in two phases, that is pertaining to common practice. In fact, the former is set a few days before the planning horizon, while the latter is decided day-by-day since cancellations and postponed surgeries could disrupt the schedule [5, 142]. For instance, this is the case of patients with comorbidities [125] or non-elective patients that arrive during the week and need to be operated on within a few days (called add-on or work-in cases [135]).

We first solve the advance scheduling problem by introducing parameters in order to provide different solutions in terms of robustness and patient mixes.

Such solutions will depend on patients' characteristics, such as their surgery duration distribution and different types of costs. Then, we solve the allocation scheduling for each of them, by providing the best overall solution.

Before presenting the stochastic programming models, we make some assumptions regarding hospital policies and patient characteristics.

Emergency patients. We assume that the number of daily emergency patients can not be greater than the number of the OR blocks. This allows us to reasonably assume that for every block only one emergency patient surgery can take place, in such a way to ensure a fair unplanned workload balancing. We observe that when this assumption does not correspond to reality, that is when the emergency patient flow increases, dedicated ORs or hybrid policies are recommended by prior studies [46, 135]. Emergency patients' arrival times are modeled through independent and identically distributed (i.i.d.) random variables (r.v.s) with uniform distribution on the OR block opening hours, that is a Poisson process. As soon as an emergency patient arrives, every OR can be assigned to them with the same probability regardless of the specialty: we randomly generate such an assignment since it could depend on exogenous factors to decision making. Then, the surgery of the emergency patient will take place as soon as possible: immediately if the assigned room is available, when the current surgery has ended otherwise. Such an insertion rule is often required for non-elective patients classified as *trauma* or *emergency* [135].

Surgical teams, beds and other resources. We assume that ORs are the patient flow's bottleneck in the considered operative context, then we assume that surgical teams, stay beds, post-anesthesia care units, and other resources are always available when needed. We also assume that each OR block has dedicated resources during its execution.

Real-time policies. We consider an operative context in which patients are always operated on according to the OR and the sequence determined by the assignment procedure and the sequencing procedure. We assume that the actual surgery start time can not be anticipated and no-shows are known only at the moment of the scheduled time. In addition to the ordinary duration L_{jk} of the OR blocks, a fixed maximum amount of overtime H can be performed. Finally, patients are operated on if and only if the estimated surgery completion time does not exceed the maximum overtime available, otherwise the surgery is postponed to the next planning horizon [46, 84].

3.4 Mathematical models

We propose a two-phase stochastic optimization approach, presenting for both the advance scheduling and the allocation scheduling a stochastic programming

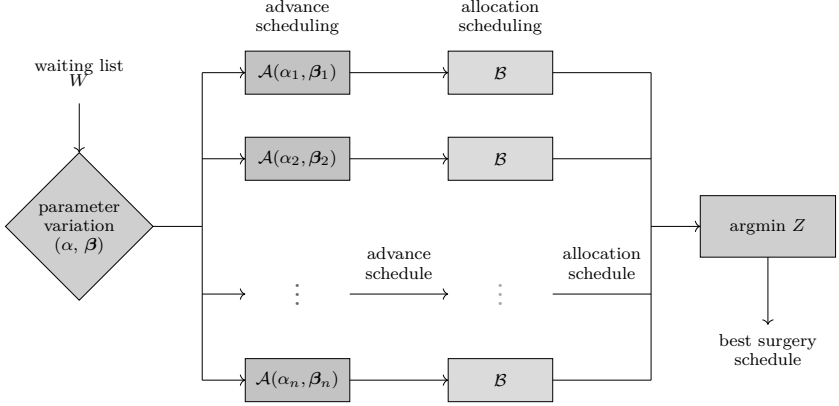


Figure 3.1: Stochastic optimization framework based on the CCIP model \mathcal{A} and the SMIP model \mathcal{B} . The robustness parameter α and the patient mix parameter β are ranged and the global surgery schedule that minimize the overall objective function Z is selected.

formulation, as shown in Figure 3.1. Given an instance of the surgical scheduling problem, the proposed framework consists of two stochastic programming models in sequence:

1. A CCIP model $\mathcal{A}(\alpha, \beta)$ provides the solution of the assignment procedure for a certain configuration of model parameters: α defines the level of robustness with respect to the probability of cancellations, while components of parameter vector β define the weights of different criteria for patient mixes within the OR blocks;
2. A SMIP model \mathcal{B} uses the previous solution(s) as an input to provide a solution for both the sequencing procedure and the timing procedure.

Although the decomposition of the surgical scheduling into two phases is often performed in literature and real-world, this division could lead to a suboptimal solution of the overall surgical scheduling problem. However, solving the assignment procedure and the sequencing procedure simultaneously is computationally complex [11, 84, 95], and it would not lead to finding a good quality overall solution within a reasonable running time.

In solving the assignment procedure, one of the most important objectives is the minimization of the patients' waiting times with respect to their urgency class [42, 128, 149], that is minimizing scheduling costs of non-scheduled patients. Other common objectives in literature are the minimization of idle time and overtime. However, the inherent uncertainty of the operating theater usually leads to significant differences between the planned schedule and

the realized one, especially when flexible policies are used to insert emergency patients within the ORs [46]. By consequence, while the total scheduling cost can be deterministically computed during the advance scheduling, the other costs (cancellation, waiting, idle time, and overtime) are affected by decisions taken during the allocation scheduling and uncertainty. Therefore, in the proposed framework, we first generate a set of advance schedules in which scheduling costs are minimized by varying the model parameters α and β . In the second phase, each of them is taken in input by the SMIP model \mathcal{B} to compute the optimal allocation schedule with respect to the other four costs, which depends on random variables.

We observe that since we assumed that specialties do not share resources, the problem defined by the CCIP model $\mathcal{A}(\alpha, \beta)$ is divided into $|S|$ independent CCIP models $\mathcal{A}_s(\alpha, \beta)$, that is one for each specialty $s \in S$. Similarly, the problem defined by the SMIP \mathcal{B} can be decomposed into $|B|$ independent SMIP models \mathcal{B}_{jk} , that is one for each OR block $(j, k) \in B$. In the end, the best overall solution is determined, that is the surgery schedule with the minimum sum of the objective functions of the CCIP models $\mathcal{A}_s(\alpha, \beta)$ and the corresponding SMIP models \mathcal{B}_{jk} , is selected.

3.4.1 Random variables

Before introducing the two models, we now define the random vector $\xi = [\rho, \delta, \tau, \theta]$, and the scenario $\omega \in \Omega$, where Ω is the sample space. Each scenario consists of the realizations of all the following r.v.s., which are independent when not differently specified.

ROT. For each patient $i \in I$, their ROT $\rho_i(\omega)$ has a lognormal distribution of mean μ_i and standard deviation σ_i .

Emergency surgery duration. For each OR block $(j, k) \in B$, the surgery duration δ_{jk} of the emergency patient assigned to (j, k) is generated according to a lognormal distribution of mean μ^{em} and standard deviation σ^{em} with probability $p^{em} \in [0, 1]$, and is equal to 0 with probability $1 - p^{em}$ (i.e. no emergency patient is assigned to that OR block).

Emergency arrival time. For each OR block $(j, k) \in B$, the emergency arrival time $\tau_{jk}(\omega)$ has uniform distribution in $[0, L_{jk}]$ (if such a surgery exists, otherwise $\tau_{jk}(\omega) = 0$) and represents the instant from the beginning of the OR block in which the emergency patient arrives. Since we generate such an arrival time independently for each OR, then the overall emergency arrivals within the whole operating theater consist of a Poisson process with rate $\lambda^{em} = p^{em} \cdot |J|$ patients per day, that is the interarrival time of emergency patients has an exponential distribution with mean $1/\lambda^{em}$ over the overall duration of the OR blocks.

No-show. For each scheduled patient $i \in I = \cup_B I_{jk}$, the r.v. $\theta_i(\omega)$ has

Bernoulli distribution of parameter $1 - r_i \in [0, 1]$, that is

$$\theta_i(\omega) = \begin{cases} 1 & \text{if in scenario } \omega \text{ patient } i \text{ is available to be operated on,} \\ 0 & \text{otherwise (no-show).} \end{cases}$$

We remark that all these r.v.s are all realized after the decisions taken at the allocation level, that is no r.v. realizes between advance scheduling and allocation scheduling.

A summary of the notation introduced in the problem statement and in the rest of this section is reported in Tables 3.2 and 3.3.

3.4.2 Advance scheduling: Chance Constrained Integer Programming model

We present the CCIP model for the advance scheduling, that is the assignment procedure. The main modeling aspects of this model are: (i) a chance constraint that sets a minimum level of robustness with respect to cancellations, and (ii) a hierarchical objective function in which three proxies are defined for patient mixes with respect to three different criteria.

The chance constraint is defined by estimating the probability of cancellation under a simplifying assumption, that is the absence of slack time in the actual OR block execution. The value of the model parameter $\alpha \in (0, 1)$ indicates that the surgeries assigned to the same OR block must have a probability of exceeding the maximum overtime lower than α . In other words, if $\alpha = 0.1$ a probability of at least 90% of not having any cancellations is required.

By varying the parameter α , we expect to have an impact on the set I of scheduled patients, as a consequence of having more or less possible feasible solutions. Contrariwise, the vector parameter β is designed to define a criterion to set a preference in patient mixes, that is the characteristics (surgery duration and costs) of patients assigned to the same OR block.

To this aim, we present a hierarchical objective function where the total scheduling cost is at the upper level and a linear combination of three proxies is at the bottom level. The proxies are defined through the following variables:

Γ^c : maximum sum of cancellation costs associated to patient within the same OR block;

Γ^w : maximum sum of waiting costs associated to patient within the same OR block;

Γ^t : sum of differences between the estimated average OR utilization and the estimated OR utilization of all OR blocks (both expressed in minutes).

The first two proxies (Γ^c and Γ^w) are introduced to balance the sum of the cancellation and waiting costs among the OR blocks of the same specialty, respectively. The rationale is that advance schedules with all (or many) patients with high cancellation or waiting costs lead to poor allocation schedules. In

Table 3.2: Notation, part one.

Sets and indices	
B	set of all OR blocks
B_s	set of all OR blocks assigned by the MSS at specialty s
i, i', \hat{i}	elective patients (or their surgery)
I	set of all scheduled patients after the advance scheduling
I_{jk}	set of patients scheduled in the OR block (j, k) after the advance scheduling
j	OR
J	set of all ORs
k	working day
K	set of days of the planning horizon
s	specialty
S	set of all specialties
W	set of all elective patients in the waiting list
W_s	set of elective patients in the waiting list of specialty s
ω	scenario
Ω	sample space
Parameters	
b_1, b_2, b_3	hierarchy and standardization coefficients
$c_i^{canc}, c_i^{sched}, c_i^{wait}$	cancellation cost, scheduling cost, and waiting cost of patient i
c^g, c^h	idle time and overtime costs (both per minute)
c_i^{sched}	scheduling cost of patient i
c_i^{wait}	direct waiting time cost (per minute) of patient i
H	maximum overtime per OR block
L_{jk}	ordinary duration of the OR block (j, k)
m_s	maximum number of patients that can be inserted within the same OR block B_s
$M, M_{ii'}$	big-M
p^{em}	probability of the insertion of an emergency patient in each OR block
r_i	probability of no-show of patient i
α	robustness parameter, i.e. approximated probability of cancellation
$\beta = \beta_1, \beta_2, \beta_3$	proxy weight parameters
$\mu_i (\mu^{em})$	expected surgery duration of elective patient i (emergency patient)
$\sigma_i (\sigma^{em})$	standard deviation of surgery duration of elective patient i (emergency patient)

fact, in this undesired case, some of those surgeries will be inevitably sequenced at the end of the OR blocks with high costs. The third proxy (Γ^t) is designed to balance the estimated OR utilization between OR blocks, promoting lower idle time and overtime costs.

Table 3.3: Notation, part two.

R.v.s	
δ_{jk}	actual surgery duration of emergency patient in OR block (j, k)
θ_i	patient presence (1 if available in the operating theater, 0 for no-show)
ξ	random vector including all r.v.s
ρ_i	actual surgery duration of elective patient i
τ_{jk}	arrival time of the emergency patient in OR block (j, k)
Decision variables	
a_i	direct waiting time of patient i
$c_i(\hat{c}_i)$	completion time of surgery i without (by) considering the emergency surgery
C	completion time of the last executed surgery of the OR block
e_i	1 if the emergency surgery is subsequent to surgery i , 0 otherwise
g_{jk}	idle time in OR block (j, k)
h_{jk}	overtime in OR block (j, k)
$o_{ii'}$	1 if surgery i' is subsequent to surgery i in the same OR block, 0 otherwise
$q_i(\hat{q}_i)$	start time of surgery i without (by) considering the emergency surgery
t_i	scheduled start time of surgery i
\bar{u}	estimated average OR utilization among all OR blocks
u_{jk}	estimated OR utilization of OR blocks (j, k)
t_i	scheduled start time of surgery i
x_{ijk}	1 if patient i is scheduled in OR block (j, k) , 0 otherwise
y_i	1 if the surgery i is cancelled due to insufficient residual time, 0 otherwise
z_i	idle time between surgery i and emergency surgery if subsequent, 0 otherwise
$\Gamma^c, \Gamma^w, \Gamma^t$	patient mix proxies

Let us introduce the decision variables

$$x_{ijk} = \begin{cases} 1 & \text{if the patient } i \in W_s \text{ is scheduled into the OR block } (j, k) \in B_s, \\ 0 & \text{otherwise.} \end{cases}$$

The CCIP model $\mathcal{A}_s(\alpha, \beta)$ is defined as follows:

$$\min \sum_{i \in W_s} c_i^{sched} \left(1 - \sum_{(j,k) \in B_s} x_{ijk} \right) + b_1 (\beta_1 \Gamma^c + \beta_2 b_2 \Gamma^w + \beta_3 b_3 \Gamma^c) \quad (3.1a)$$

$$\text{s.t.} \quad \sum_{(j,k) \in B_s} x_{ijk} \leq 1, \quad i \in W_s, \quad (3.1b)$$

$$\sum_{i \in W_s} \mu_i x_{ijk} \leq L_{jk}, \quad (j, k) \in B_s, \quad (3.1c)$$

$$\mathbb{P}_{\xi} \left[\delta_{jk} + \sum_{i \in W_s} \theta_i \rho_i x_{ijk} > L_{jk} + H \right] \leq \alpha, \quad (j, k) \in B_s, \quad (3.1d)$$

$$\Gamma^c = \max_{(j,k) \in B_s} \sum_{i \in W_s} c_i^{canc} x_{ijk}, \quad (3.1e)$$

$$\Gamma^w = \sum_{(j,k) \in B_s} |\bar{u} - u_{jk}|, \quad (3.1f)$$

$$\Gamma^t = \sum_{(j,k) \in B_s} |\bar{u} - u_{jk}|, \quad (3.1g)$$

$$\bar{u} = \frac{1}{|B_s|} \sum_{(j,k) \in B_s} u_{jk}, \quad (3.1h)$$

$$u_{jk} = \mathbb{E}_{\xi} [\delta_{jk}] + \sum_{i \in W_s} \mathbb{E}_{\xi} [\theta_i \rho_i] x_{ijk}, \quad (j, k) \in B_s, \quad (3.1i)$$

$$x_{ijk} \in \{0, 1\}, \quad i \in W_s, (j, k) \in B_s, \quad (3.1j)$$

$$u_{jk} \geq 0, \quad (j, k) \in B_s, \quad (3.1k)$$

$$\Gamma^c, \Gamma^w, \Gamma^t, \bar{u} \geq 0. \quad (3.1l)$$

Objective function (3.1a) is defined as the sum of the total scheduling cost (first summation) and the linear combination of the three proxies, multiplied by the coefficient b_1 to create the hierarchy between the two levels. Furthermore, coefficients b_2 and b_3 are defined to normalize the three proxy variables in order to have the same order of magnitude. To this aim, we first estimate an upper bound of the number of patients that can be scheduled within the same OR block of the specialty $s \in S$

$$m_s = \left\lfloor \frac{\max_{(j,k) \in B_s} L_{jk}}{\min_{i \in W_s} \mu_i} \right\rfloor.$$

Then, we define the coefficient

$$b_2 = \frac{\max_{i \in W_s} c_i^{canc}}{\max_{i \in W_s} c_i^{wait}},$$

which balance the first two proxies with respect to their worst case, that is when in an OR block the maximum number of patients is scheduled and they all have the maximum cancellation and waiting cost, respectively. We introduce the coefficient

$$b_3 = \frac{m_s \max_{i \in W_s} c_i^{canc}}{|B_s| \max_{(j,k) \in B_s} L_{jk}/2},$$

which is the ratio between the worst case for the first proxy and an approximation of the one for the third proxy (i.e. one half of OR block has full occupation and the other half is empty). The coefficient b_1 is finally introduced in such

a way to establish a hierarchy in which the sum of the scheduling cost is at the upper level and the proxies are at the bottom level. Let us suppose that the scheduling cost of a surgery $i \in W$ is defined in such a way to have $c_i^{sched} = n\Delta^{sched}$ for some $n \in \mathbb{N}$. Then, we set

$$b_1 = \frac{\Delta^{sched}}{1 + m_s \max_{i \in W_s} c_i^{canc}},$$

ensuring that proxies cannot lead to choosing as the optimal solution a solution that has a total scheduling cost higher than another feasible solution. Then, by setting the vector of parameters $\beta = (\beta_1, \beta_2, \beta_3)$, with $\beta_1, \beta_2, \beta_3 \in [0, 1]$ and $\|\beta\|_1 = 1$, it is possible to define the weights of the three different proxies in such a way to set the level of preference between the patient mix criteria. We highlight that β could or could not have also an impact on the set I of scheduled patients. We distinguish to cases: (i) there exists a unique set I that minimizes the total scheduling cost in the search space, or (ii) there are two or more sets I with the same minimum total scheduling cost. In the former case, the value of β does not affect the decision about the surgery to be scheduled, because of the hierarchy of the objective function imposed through b_1 . In the latter case, different sets I could lead to different values of the proxy variables, as a consequence the bottom level of the objective function also acts on the selection of the surgeries. Constraints (3.1b) impose that each surgery $i \in W_s$ can be scheduled at most once. Constraints (3.1c) and (3.1d) are the deterministic and stochastic OR capacity constraints. The former constraints ensure that the sum of the EOT of patients scheduled within an OR block does not exceed the ordinary duration L_{jk} . The latter are chance constraints imposing the robustness with respect to the probability of cancellation discussed above. Constraints (3.1e)–(3.1f) define the cancellation and waiting time proxies, respectively. Constraints (3.1g)–(3.1i) are used to compute the third proxy. In particular, stochastic constraints (3.1i) estimate the OR utilization u_{jk} of every single OR block $(j, k) \in B_s$ under the same assumption of the chance constraints (3.1d) and without considering the effect of possible cancellations due to the same considerations about the lack of knowledge of decision that will be taken in the allocation scheduling. Then, constraint (3.1h) computes the average OR utilization \bar{u} over all the estimates and constraint (3.1g) computes the 1-norm of the vector of differences between the estimated OR utilizations and their mean. Finally, domain constraints are reported in (3.1j)–(3.1l).

3.4.3 Allocation scheduling: two-phase Stochastic Integer Programming model

Given an optimal solution \mathbf{x}^* of the CCIP model $\mathcal{A}_s(\alpha, \beta)$, we define the set $I_{jk} = \{i \in W_s \mid x_{ijk}^* = 1\}$ of all patients scheduled in the OR block $(j, k) \in B_s$. Then, for each OR block we can solve the allocation scheduling problem independently. We propose a two-stage SMIP model \mathcal{B}_{jk} that solves

the sequencing procedure and timing procedure simultaneously: at the first-stage (model \mathcal{B}_{jk}^I) the expected sum of all the costs is optimized, while in the second-stage (model $\mathcal{B}_{jk}^{II}(\omega)$) we compute the value of several variables in the scenario $\omega \in \Omega$ under the real-time policies defined in Section 3.3. In order to provide a solution for the sequencing procedure and the timing procedure, two types of decision variables are used in \mathcal{B}_{jk}^I , that is

$$o_{ii'} = \begin{cases} 1 & \text{if the surgery } i' \in I_{jk} \text{ is subsequent to the surgery } i \in I_{jk}, \\ 0 & \text{otherwise,} \end{cases}$$

and the scheduled time t_i of the patient $i \in I_{jk}$ since the OR block start time (min). The first stage of the SMIP model \mathcal{B}_{jk}^I is defined as follows:

$$\min \quad \mathbb{E}_{\xi}[Q(\mathbf{o}, \mathbf{t}; \xi(\omega))] \quad (3.2a)$$

$$\text{s.t.} \quad t_i \leq (L_{jk} - \mu_i) \sum_{i' \in I_{jk} \setminus \{i\}} o_{ii'}, \quad i \in I_{jk}, \quad (3.2b)$$

$$t_i + \mu_i \leq t_{i'} + (1 - o_{ii'})M_{ii'}, \quad i, i' \in I_{jk}, i \neq i', \quad (3.2c)$$

$$\sum_{i \in I_{jk}} \sum_{i' \in I_{jk} \setminus \{i\}} o_{ii'} = |I_{jk}| - 1, \quad (3.2d)$$

$$o_{ii'} \in \{0, 1\}, \quad t_i \geq 0, \quad i, i' \in I_{jk}, i \neq i'. \quad (3.2e)$$

Objective function (3.2a) represents the expected value of the recourse function $Q(\mathbf{o}, \mathbf{t}; \xi(\omega))$ over the joint distribution of $\xi(\omega)$. Constraints (3.2b) ensure that each patient i 's planned completion time (i.e. the scheduled start time t_i plus the EOT μ_i) does not exceed the OR block capacity L_{jk} , forcing the first patient \hat{i} to be equal to the OR block start time ($t_{\hat{i}} = 0$). Constraints (3.2c) impose scheduled start times t_i to be consistent with the order defined by $o_{ii'}$ and the planned completion times $t_i + \mu_i$. Constraints (3.2d) express the logical nature of variables $o_{ii'}$, by guaranteeing to have a complete sequence of all the surgeries in I_{jk} . Domain constraints are defined in (3.2e).

The recourse function $Q(\mathbf{o}, \mathbf{t}; \xi(\omega))$ corresponds to the objective function of the second-stage stochastic programming model. Thus, the expected value in objective function (3.2a) is a multidimensional integral of a function that is implicitly defined by a deterministic programming model. The aim of this model is to calculate a weighted sum of costs associated with overtime, idle time, cancellations, and direct waiting time of a scenario ω . This calculation takes into account a set of rules that define all the decisions about overtime allocation, cancellation, and insertion of the (possible) emergency surgery.

In Figures 3.2 and 3.3, we report two examples of schedule for a general OR block (j, k) with 3 elective patients ($i = 1, 2, 3$) and 6 possible scenarios: 4 different scenarios $\omega_1, \omega_2, \omega_3, \omega_4 \in \Omega$ without the insertion of emergency surgeries ($\delta_{jk} = 0$) are presented in Figure 3.2, while 2 further scenarios $\omega_5, \omega_6 \in \Omega$ with the insertion of an emergency surgery ($\delta_{jk} > 0$) are shown in Figure 3.3. Since the direct waiting time a_i of the patient i is defined as the difference between the actual start time and the scheduled start time t_i , we need to

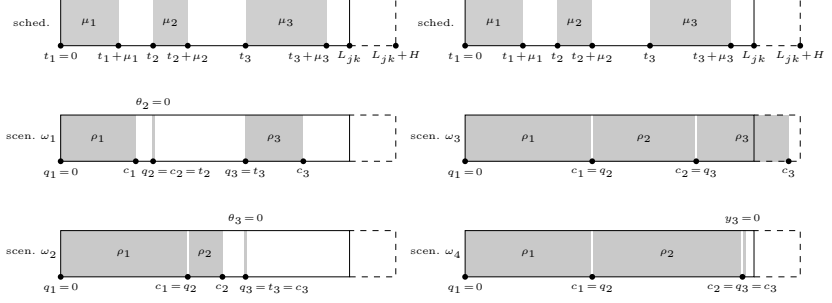


Figure 3.2: Example of OR block schedule and 4 possible scenarios ($\omega_1, \omega_2, \omega_3, \omega_4 \in \Omega$) such that the emergency surgery is not inserted in the considered OR block ($\delta_{jk} = 0$).

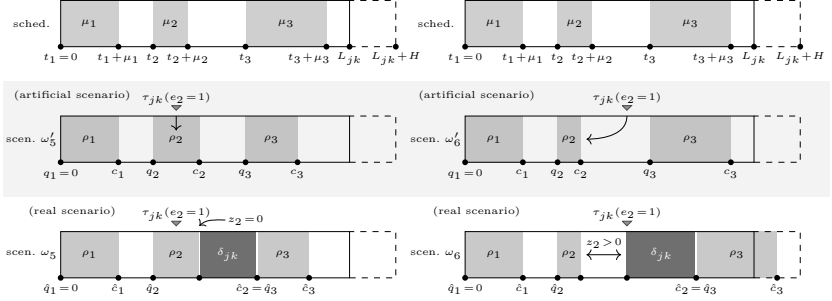


Figure 3.3: Example of OR block schedule and 2 possible scenarios ($\omega_5, \omega_6 \in \Omega$) such that the emergency surgery is inserted in the considered OR block ($\delta_{jk} > 0$). Artificial scenarios (middle frame) ω_5', ω_6' are auxiliary scenarios built with only elective surgeries to identify the surgery i to which the emergency surgery is subsequent (indicated with an arrow), and their actual start time, accordingly. When it is not specified, $z_i = e_i = 0$.

compute the first one with respect to all the uncertainty factors that could have an effect on it. For the same reason, we need to determine the actual start and completion times of all patients to compute both the overtime and the idle time, and to establish when surgeries must be canceled.

Firstly, let us consider a general scenario with no emergency surgery to be inserted, such as in Figure 3.2. We indicate with q_i the actual start time of the surgery i . Such time is equal to t_i if the completion time $c_{i'}$ of the previous surgery does not exceed t_i . Otherwise, the surgery of i starts as soon as the previous surgery ends, that is at time $c_{i'}$. The completion time is computed as the sum of the actual start time and the ROT ($c_{i'} = q_{i'} + \rho_{i'}$).

For instance, in scenario ω_1 the actual start time of all patients is equal to the scheduled one, since no uncertainty factor compromises the execution of the ex-ante schedule. We remark that, in this scenario, we have a no-show for the patient $i = 2$ (i.e. $\theta_2 = 0$) and, by convention, they will also have a start and completion time set in such a way that they coincide ($q_i = c_i$), although their possible waiting time and cancellation are not considered in terms of cost. An example of direct waiting time greater than 0 can be observed for patient $i = 2$ in scenario ω_2 , where the actual duration ρ_1 of surgery $i = 1$ deviates significantly from the expected value μ_1 , leading to a delay in the completion time c_1 and the consequent postponement of the start of the next surgery, which is equal to c_1 .

Furthermore, we need to introduce the decision variable

$$y_i = \begin{cases} 1 & \text{if surgery } i \text{ is executed or is a no-show} \\ 0 & \text{otherwise} \end{cases}, \quad i \in I_{jk},$$

to deal with cases in which the maximum overtime H is not sufficient to operate on a patient i with respect to their actual start time and the expected surgery duration μ_i . Two opposite situations are presented in scenarios ω_3 and ω_4 depicted in Figure 3.2: an amount of overtime equal to $c_3 - L_{jk}$ is used in the former to operate on the patient $i = 3$, while their cancellation ($y_3 = 0$) is necessary in the latter.

To deal with the possible insertion of an emergency surgery (see Figure 3.3), we need to introduce two further variables \hat{c}_i and \hat{q}_i for every patient $i \in I_{jk}$. In these scenarios, c_i and q_i are not the actual completion and start times because the insertion of the emergency surgery could lead to a forward movement of the following elective surgeries. In this case, the actual completion and start time are represented by \hat{c}_i and \hat{q}_i , respectively. By consequence, we use c_i and q_i as auxiliary variables to build an artificial scenario ω' where the emergency surgery is not inserted, then we define \hat{c}_i and \hat{q}_i by taking into account the emergency insertion. In scenarios without an emergency patient (i.e. Figure 3.2), the actual completion and start times coincide with the corresponding auxiliary variables, that is $c_i = \hat{c}_i$ and $q_i = \hat{q}_i$.

When the emergency patient arrives (i.e. at time τ_{jk}) they need to be inserted as soon as possible. Two alternative situations could happen.

- A surgery i is currently being performed in the OR (e.g. see scenario ω_5 in Figure 3.3), then the emergency surgery will start at the completion time c_i . By consequence, the idle time z_i between the surgery i and the emergency one is equal to 0.
- The OR is currently available (e.g. see scenario ω_6 in Figure 3.3), then the surgery can be started immediately. We compute the idle time between the previous elective surgery i and the emergency one, which is $z_i = \tau_{jk} - c_i$.

In both situations, we use the auxiliary variable

$$e_i = \begin{cases} 1 & \text{if the emergency surgery follows the surgery } i \\ 0 & \text{otherwise} \end{cases}, \quad i \in I_{jk},$$

to indicate the point in the surgery sequence in which the emergency patient is inserted. Conventionally, we set the variables $z_{i'} = 0$ for the elective surgeries $i' \in I_{jk}$ that do not immediately precede the emergency surgery. Then, we set the variable \hat{c}_i , associated with the elective surgery i such that $e_i = 1$, equal to the completion time of the emergency surgery instead of its own completion time. Finally, we compute the cascade of actual starting times \hat{q}_i and completion times \hat{c}_i of the following surgeries. We remark that it holds $c_i = \hat{c}_i$ and $q_i = \hat{q}_i$ for all surgeries that precede the surgery i' such that $e_{i'} = 0$ and for all surgeries in scenarios without the emergency insertion.

The second-stage problem is represented for each scenario by the following programming model $\mathcal{B}_{jk}^{II}(\omega)$:

$$\min c^h h_{jk} + c^g g_{jk} + \sum_{i \in I_{jk}} c_i^{anc}(1 - y_i) + \sum_{i \in I_{jk}} c_i^{wait} a_i \quad (3.3a)$$

$$\text{s.t. } o_{ii'} = 1 \Rightarrow q_{i'} = \max\{c_i, t_{i'}\} \wedge \hat{q}_{i'} = \max\{\hat{c}_i, t_{i'}\}, i, i' \in I_{jk}, i \neq i', \quad (3.3b)$$

$$q_i, \hat{q}_i \leq M \sum_{i' \in I_{jk} \setminus \{i\}} o_{ii'}, \quad i \in I_{jk}, \quad (3.3c)$$

$$c_i = q_i + \rho_i(\omega) \theta_i(\omega) y_i + z_i + \delta_{jk}(\omega) e_i, \quad i, i' \in I_{jk}, i \neq i', \quad (3.3d)$$

$$\hat{c}_i = \hat{q}_i + \rho_i(\omega) \theta_i(\omega) y_i, \quad i, i' \in I_{jk}, i \neq i', \quad (3.3e)$$

$$C \geq \theta_i(\omega)(q_i + \rho_i(\omega) y_i) + z_i + \delta_{jk}(\omega) e_i - (1 - y_i)M, \quad i \in I_{jk}, \quad (3.3f)$$

$$C \geq \tau_{jk}(\omega) + \delta_{jk}(\omega), \quad (3.3g)$$

$$z_i \leq M e_i, \quad i \in I_{jk}, \quad (3.3h)$$

$$\sum_{i \in I_{jk}} e_i = 1, \quad i \in I_{jk}, \quad (3.3i)$$

$$e_i = 1 \wedge o_{ii'} = 1 \Leftrightarrow \hat{q}_i \leq \tau_{jk}(\omega) < \hat{q}_{i'}, \quad i, i' \in I_{jk}, i \neq i', \quad (3.3j)$$

$$\begin{cases} e_i = 1 \\ \tau_{jk}(\omega) > q_i + \rho_i(\omega) \theta_i(\omega) y_i \end{cases} \Rightarrow \begin{cases} z_i = \tau_{jk}(\omega) - \\ (q_i + \rho_i(\omega) \theta_i(\omega) y_i), \end{cases} \quad i \in I_{jk}, \quad (3.3k)$$

$$\theta_i(\omega)(q_i + \mu_i) \leq L_{jk} + H \Leftrightarrow y_i = 1, \quad i \in I_{jk}, \quad (3.3l)$$

$$y_i \geq 1 - \theta_i(\omega), \quad i \in I_{jk}, \quad (3.3m)$$

$$a_i \geq q_i - t_i - M(1 - y_i \theta_i(\omega)), \quad i \in I_{jk}, \quad (3.3n)$$

$$h_{jk} \geq C - L_{jk}, \quad (3.3o)$$

$$g_{jk} \geq \max\{L_{jk}, C\} - \sum_{i \in I_{jk}} \rho_i(\omega) \theta_i(\omega) y_i - \delta_{jk}(\omega), \quad (3.3p)$$

$$h_{jk}, g_{jk}, q_i, \hat{q}_i, c_i, \hat{c}_i, C, z_i, a_i \geq 0, y_i, e_i \in \{0, 1\}, \quad i \in I_{jk}. \quad (3.3q)$$

Objective function (3.3a) is multi-objective and includes four terms. The first term $c^h h_{jk}$ is the total overtime cost in the considered OR block (j, k) ; the second term $c^g g_{jk}$ is the total idle time cost in the OR block (j, k) ; then

we consider the sum of cancellation costs c_i^{canc} for canceled surgeries i due to insufficient residual time; finally, we consider the sum of waiting costs, computed as the cost per minute c_i^{wait} multiplied by the minutes of direct waiting time a_i of the patient i .

Constraints (3.3b)–(3.3c) define the actual start times \hat{q}_i in the considered scenario and the start time q_i in the corresponding artificial scenario. Constraints (3.3b) define the actual start times as the maximum between the scheduled start times and the completion times of the previous patients (if they exist). Constraints (3.3c) impose that the start time of the first surgery is equal to the OR block's start time, that is 0, while it becomes redundant with respect to (3.3b) for all other surgeries due to the big-M.

Similarly, variables \hat{c}_i and c_i are defined by constraints (3.3d)–(3.3e) as the actual completion times in the real scenario and in the artificial scenario, respectively. In both the equalities, the ROT of i is summed to the actual start times only if the patient is available to be operated on ($\theta_i = 1$) and is not necessary to cancel their surgery ($y_i = 1$). In the particular case in which the emergency surgery is inserted immediately after the patient i ($e_i = 1$), \hat{c}_i is defined as the completion time of the emergency patient, to which two further durations are added: the idle time z_i between the elective patient i and the emergency patient, and the surgery duration $\delta_{jk}(\omega)$ of the emergency patient inserted in the OR block (j, k) . For instance, in both scenarios ω_5 and ω_6 of Figure 3.3 the emergency surgery is subsequent to the elective surgery $i = 2$, then $\hat{c}_2 = \hat{q}_2 + z_2 + \delta_{jk}$.

Constraints (3.3f)–(3.3g) compute the maximum completion time C , that is the actual end time of the OR block, remarking that one among constraints (3.3f) and constraint (3.3g) is more strict with respect to all the others. There are two possible cases that cause the redundancy of a constraint with respect to the strictest one: (i) the considered surgery is not the last executed in the OR block, and (ii) the considered surgery is not executed. For instance, in scenario ω_2 in Figure 3.2 we have $C = \hat{c}_2 = c_2$ because when considering the constraint (3.3f) for $i = 1$, we have that $C \geq \hat{c}_1$, but for $i = 2$ it holds that $C \geq \hat{c}_2 = c_2 > \hat{c}_1 = c_1$. At the same time, we do not consider the completion time of the surgery $i = 3$ due to its no-show and the absence of an emergency patient after it, which leads to the trivial constraint $C \geq 0$ when considering the constraint (3.3f) for $i = 3$. Similarly, in scenario ω_4 it holds that $C = \hat{c}_2 = c_2$ since the cancellation of the patients $i = 3$ enables the big-M when considering the same constraint. Furthermore, constraint (3.3g) is necessary for the particular case in which the emergency arrival occurs after the auxiliary actual starting time \hat{q}_i of the last surgery of the OR block, but it is a no-show or is canceled.

Furthermore, when the emergency patient does not succeed the patient i ($e_i = 0$), variable z_i is fixed to 0 by constraints (3.3h) because there is no idle time to compute between such a surgery and the emergency one (e.g. this is the case of scenario ω_5 in Figure 3.3).

Constraints (3.3i) ensure that the emergency surgery is inserted exactly one time. Constraints (3.3j) implement the emergency surgery's insertion policy, that is (i) at the exact moment $\tau_{jk}(\omega)$ of the emergency patient arrival if the OR is available (such as in scenario ω_6 in Figure 3.3), (ii) immediately after the surgery currently in progress otherwise (such as in scenario ω_5 in Figure 3.3). Constraints (3.3k) compute the idle time between the emergency surgery and the previous elective surgery i , that is equal to 0 when the emergency patient arrives while elective patient i is within the OR and they is operated immediately after the end of their surgery.

Constraints (3.3l) apply the cancellation policy, that is a certain surgery is canceled if and only if its expected completion time (i.e. $\hat{q}_i + \mu_i$) would exceed the maximum overtime (i.e. $L_{jk} + H$), then constraints (3.3m) avoid the cancellation if the patient is a no-show.

Constraints (3.3n) compute the waiting time of the patient i as the difference between the actual and the scheduled start time, using a big-M to set to 0 the waiting time of no-shows and patient whose surgery has been canceled. Constraint (3.3o) defines the overtime as the difference between the maximum completion time C and the OR block ordinary duration L_{jk} . Constraint (3.3p) defines the idle time as the difference between the maximum completion time C and the sum of the ROTs of the patients operated on.

All variables are bounded due to domain constraints (3.3q). All non-trivial constraints presented here in a non-linear form for the sake of summary are reported in the corresponding linear form in Appendix C.

3.4.4 Overall objective function

We can define the overall objective function of the surgical scheduling problem defined in the stochastic optimization framework at the beginning of this section (see Figure 3.1) as follows:

$$\begin{aligned}
 Z = & \sum_{s \in S} \sum_{i \in W_s} c_i^{sched} \left(1 - \sum_{(j,k) \in B_s} x_{ijk} \right) + \\
 & + \sum_{(j,k) \in B} \mathbb{E}_{\xi} \left[c^h h_{jk} + c^g g_{jk} + \sum_{i \in I_{jk}} \left(c_i^{canc} (1 - y_i) + c_i^{wait} a_i \right) \right], \tag{3.4}
 \end{aligned}$$

where variables at the bottom level of the hierarchical objective function of $\mathcal{A}_s(\alpha, \beta)$ are not included since they are only used as proxies to link the two phases of the stochastic optimization framework.

3.5 Methodology

Because of the complex structure of the stochastic process under which the surgical scheduling is executed, closed-form expressions of the chance constraints

(3.1d) of $\mathcal{A}_s(\alpha, \beta)$ and of the stochastic objective function (3.2a) of \mathcal{B}_{jk} are complex to be defined in an exact way or by introducing a good approximation. For this reason, we propose a Monte Carlo sampling to deal with the chance constraints (3.1d) for the advance scheduling in Section 3.5.1 and two different versions of the SAA method to deal with the stochastic objective function of the allocation scheduling model in Section 3.5.2. Nevertheless, as we will show in Section 3.6, the combination of the stochastic and computational complexities of \mathcal{B}_{jk} makes the SAA methodology not effective for some instances, because it consists of the solution of a mixed integer linear programming model with a high number of variables. Therefore, in Section 3.5.3 we propose a genetic algorithm with a custom encoding and a fitness function evaluation based on Monte Carlo sampling to find a near-optimal solution with a reasonable computational cost.

3.5.1 Advance Scheduling: Monte Carlo sampling

The CCIP model $\mathcal{A}_s(\alpha, \beta)$ is solved by approximating the chance constraints (3.1d) and the stochastic constraints (3.1i) through a Monte Carlo sampling. Firstly, a random sample \mathcal{S} is generated. Then, the probability in (3.1d) is approximated by

$$P_{jk} = \frac{1}{|\mathcal{S}|} \sum_{\omega \in \mathcal{S}} \mathbf{1}_{(L_{jk}+H, +\infty)} \left(\delta_{jk}(\omega) + \sum_{i \in W_s} \theta_i(\omega) \rho_i(\omega) x_{ijk} \right), \quad (j, k) \in B_s, \quad (3.5)$$

where the indicator function $\mathbf{1}_{(L_{jk}+H, +\infty)}$ is used to compute the number of scenarios in which the internal inequality is not satisfied. Similarly, auxiliary variables u_{jk} defined in constraints (3.1i) are approximated with their sample average

$$\tilde{u}_{jk} = \frac{1}{|\mathcal{S}|} \sum_{\omega \in \mathcal{S}} \left(\delta_{jk}(\omega) + \sum_{i \in W_s} \theta_i(\omega) \rho_i(\omega) x_{ijk} \right), \quad (j, k) \in B_s. \quad (3.6)$$

3.5.2 Allocation Scheduling: SAA and N -fold SAA

We propose two alternative SAA methods, that is objective function (3.2a) computed through a randomly generated sample \mathcal{S} of size $|\mathcal{S}|$, to find a near-optimal solution of \mathcal{B}_{jk} .

SAA: The SMIP model is solved by computing the SAA method [78] on \mathcal{S} .

N -fold SAA (SAA_N): The sample \mathcal{S} is partitioned into N folds of size $|\mathcal{S}|/N$. Then, the stochastic programming is solved by computing the

SAA method on each subset $\mathcal{F}_n \subset \mathcal{S}$, $n = 1, \dots, N$. The solution obtained for each fold \mathcal{F}_n is then reevaluated on the entire sample \mathcal{S} through an algorithmic straightforward implementation of \mathcal{B}_{jk}^{II} . Finally, the one with the minimum objective function value is selected.

3.5.3 Allocation scheduling: a genetic algorithm

We present a custom version of the **Biased Random-Key Genetic Algorithm (BRKGA)**, which is an effective method for tackling sequencing problems [57] and we adapt it in order to solve also the timing procedure. In our customization, each solution of the optimization problem consisting of the sequencing procedure and the timing procedure for an OR block (j, k) is encoded through a vector

$$\Gamma = (\gamma_1, \dots, \gamma_{|I_{jk}|}, \gamma'_1, \dots, \gamma'_{|I_{jk}|})$$

of length $2|I_{jk}|$ called chromosome. Every component of the chromosome, called gene, contains a real number in the unit interval: the first half of the chromosome is used for the sequencing procedure, while the second half is used for the timing procedure.

Let us indicate with $i = 1, \dots, |I_{jk}|$ the surgeries in I_{jk} , then γ_i , $1 \leq i \leq |I_{jk}|$, represents the order of patient i in the sequence of surgeries, that is the surgeries are sequenced in increasing order of the corresponding gene. By consequence, variables $o_{ii'}$ are fixed according to their definition.

The i -th gene of the second half γ'_i is used to determine the slack time after the i -th surgery. Here, the slack time corresponds to the duration of the time interval between the end of the i -th surgery and the beginning of the subsequent one. If the considered surgery is the last of the OR block, the slack time is considered with respect to the ordinary end of the OR block. The values of genes γ'_i are re-scaled according to the total idle time during the encoding phase. Without loss of generality, let us suppose that the patient i is the i -th in the surgeries sequence, otherwise the patients can be renamed after the decoding of the first half of the chromosome. Firstly, the total scheduled idle time

$$G = L_{jk} - \sum_{i \in I_{jk}} \mu_i,$$

is computed as the difference between the OR block duration and the sum of the EOTs of the scheduled surgeries. Then, the scheduled start time t_i of the patient in position i in the surgeries sequence is computed as follows

$$t_1 = 0, \\ t_{i+1} = t_i + \mu_i + \frac{\gamma'_i}{\sum_{i'=1}^{|I_{jk}|} \gamma'_{i'}} G, \quad i = 1, \dots, |I_{jk}| - 1,$$

where $t_i + \mu_i$ is the scheduled completion time of the surgery i , while the other term is the decoded slack time left after it.

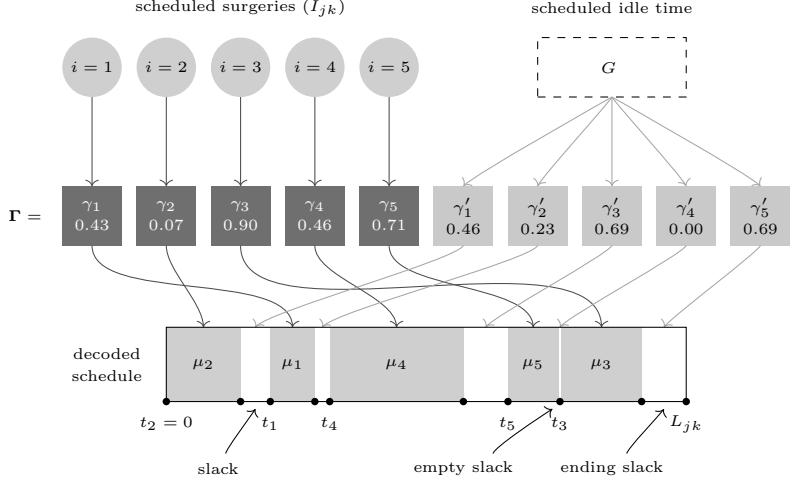


Figure 3.4: Example of the encoding used in the custom version of the BRKGA.

An example of a chromosome with the proposed encoding for an OR block with 5 scheduled patients is represented in Figure 3.4. In this particular instance, the first half of genes in the chromosome Γ are such that $\gamma_2 < \gamma_1 < \gamma_4 < \gamma_5 < \gamma_3$, then the associated solution of the sequencing procedure is given by the sequence $(2, 1, 4, 5, 3)$. Furthermore, we need to decode the solution of the timing procedure. The surgery $i = 2$ is implicitly scheduled at the OR block start time, that is $t_2 = 0$. To define the scheduled start times of the other 4 surgeries, we consider the second half of genes of Γ . Since γ'_1 is equal to $2/9$ of the sum $\gamma'_1 + \gamma'_2 + \gamma'_3 + \gamma'_4 + \gamma'_5$, then the slack between the first and the second surgery is set equal to $2/9$ of the scheduled idle time G . Similarly, due to the values of γ'_2 and γ'_3 we set a slack of duration $G/9$ between the second and the third surgeries, and a slack of duration $G/3$, respectively. Since it holds that $\gamma'_4 = 0$, there is no slack between the fourth and the fifth surgeries, then the scheduled start time of the surgery $i = 3$ is set equal to the scheduled completion time of the surgery $i = 5$. Finally, the last gene γ'_5 defines the ending slack, that is the scheduled idle time after the last scheduled surgery.

The BRKGA applies evolutionary dynamics over a population of individuals. Each individual of the population is characterized by a chromosome. Each chromosome is associated with a fitness value, defined as follows:

$$\text{fitness}(\mathbf{t}) = \frac{1}{|\mathcal{S}|} \sum_{\omega \in \mathcal{S}} Q(\mathbf{o}, \mathbf{t}; \xi(\omega)), \quad (3.7)$$

which is an approximation of the objective function (3.2a) through a Monte Carlo simulation over the sample \mathcal{S} . Fitness values are computed through an algorithmic straightforward implementation of \mathcal{B}_{jk}^I that reduces the computational time.

Each gene of each chromosome of each individual of the first population is sampled uniformly at random within the unit interval. For each following generation, that is any following iteration of the algorithm, the BRKGA uses an elitist strategy for generating new offspring. A fixed user-defined fraction of individuals with the lowest fitness values, called elites, is entirely copied to the new population. In this way, good solutions monotonically improve over generations. Then, a user-defined fraction of non-elite individuals, called mutants, is generated uniformly at random and added to the new population. The remaining individuals are generated through a mating mechanism between one member of the elite set and one member of the non-elite set of the previous population. Each gene of the offspring is inherited with a user-defined probability \tilde{p} from the elite parent, and with probability $1 - \tilde{p}$ from the non-elite parent. In this way, the generation of new individuals is biased toward the fittest chromosomes. This process terminates either within a fixed number of generations or a given time limit, returning the individual with the minimum fitness value.

We remark that the proposed chromosome encoding is designed in such a way that it always satisfies constraints (3.2b)–(3.2d), guaranteeing the feasibility of the solution associated with every gene. This is a good property that allows us to exploit the whole population to explore the search space and speed up convergence.

3.6 Experimental setup

In this section, we present the creation of the instances that will be used in the computational analysis in Section 3.7. In order to test the proposed stochastic optimization approach on realistic scenarios and to provide reliable managerial insights, we start from data from a real case and we deal with the lack of missing data enhancing it with information available in the literature and data fitting, providing realistic assumptions arising from logical considerations.

3.6.1 *Specialties: MSS, surgical procedures, and distributions*

We generated several instances starting from the open data in [94] about the major hospital in the city of Oslo (*Sykehuset Asker og Baerum HF*), including information about surgeries of 5 specialties over a period of 3 years: Cardiology (CARDIO), Gastroenterology (GASTRO), Gynecology (GYN), Orthopedics (ORTH), and Urology (URO). We do not include General Medicine (MED) in our experiment due to the low number of surgeries of this specialty, with only the 3% of total cases and an OR block allocated occasionally, which would lead to a trivial scheduling. The patient type (elective or emergency), the surgical

Table 3.4: MSS mined from data in [94].

OR	Mon	Tue	Wed	Thu	Fri
1	GASTRO	GYN	GYN		
2	GASTRO	GYN	GYN	GYN	GASTRO
3		CARDIO		CARDIO	CARDIO
4		URO		URO	URO
5	ORTH	ORTH		URO	
6		ORTH	ORTH		
7	ORTH		ORTH	ORTH	
8			GASTRO	GASTRO	
9	GYN			GASTRO	

specialty, and the start and completion time of the surgery are available in the data set. Therefore, we are able to compute the elective patients' ROT distribution for each specialty and the daily emergency arrival rate. However, other missing information are usually known in practice when the surgical scheduling is executed, such as the surgical procedure, the EOT, the patient's age, the OR, and other attributes defining scheduling, cancellation and waiting costs [9, 37, 106, 128, 134]. For this reason, existing data have been integrated through a realistic random generation of the missing ones.

We deduced the MSS, that is assignment of the OR blocks to the 5 specialties over a planning horizon of 5 weekdays from the timestamps of the surgeries in the data set. Since we observed significant variations over the 3-year period, we considered only the last year (2008). For each week and for each day, we compute the total duration of the elective surgeries that happened that day. We assumed the availability of 9 ORs and that the duration of every OR block is 480 min. Thus, we deduced a fixed weekly pattern for the last 8 weeks of 2008. Such a pattern, summarized in Table 3.4, is the MSS used in our analysis.

The elective patients' ROT distribution of every specialty presents several peaks, which suggest the underlying presence of different surgical procedures. In the hypothesis of having the necessary data (which we are artificially reconstructing here), we assume that machine learning techniques can be used to identify a certain number of surgical procedure groups. The surgical procedures of the same groups belong to the same specialty and have similar ROT distribution. Contrarily, surgical procedures in different groups have significantly different ROT distribution, in accordance with the medical literature. For instance, possible urology surgical procedures include dilation of urethra (<30 min), nephrectomy (≈ 3 h), and total cysectomy (>6 h) [106]. Since the durations of surgical procedure groups can be approximated with a log-normal distribution [11, 60, 124], we assume that the elective patients' ROT distribution is a lognormal mixture, where each component is the distribution related to a surgical procedure group. We remark that our purpose is not to predict the surgical procedure durations of the real cases in [94], but to

generate realistic EOTs consistently with the available ROTs ρ . We used the function `normalmixEM` from the package `mixtools` [12] in R 4.1.1 to deduce the distributions that lead to the best fitting Gaussian mixture with respect to the logarithm of the ROTs. We ranged the number m of mixtures' components between 1 and 8, and for each Gaussian component we allowed values of standard deviation in $\{0.15, 0.20\}$, which consist in coefficients of variation of the surgical durations equal to about 0.151 and 0.202, allowing us to model two different level of variability. Finally, we selected the mixture with the configuration that approximates better the specialty's mean and standard deviation.

From the fitted mean and standard deviation of the logarithm of surgery durations, it is possible to deduce the mean μ (EOT) and the standard deviation σ of the surgery duration for each surgical procedure group, while the weights establish their probability to be generated within the specialty. Then, the ROTs of a surgical procedure group have lognormal distribution of parameters μ and σ . In Table 3.5 we list for each specialty $s \in S$ the number m_s of components (surgical procedure groups) of the best-fit mixture, with their frequencies \mathbf{f} , average values $\boldsymbol{\mu}$, and standard deviations $\boldsymbol{\sigma}$. We remark that such measures are expressed in minutes. In the right column, we illustrate the graph of the duration distribution by representing the logarithm of the ROT on the x-axis. Thus, the probability density function results in a Gaussian mixture. Furthermore, we report the relative errors ϵ_μ and ϵ_σ of mean and standard deviation of the best-fitting mixtures with respect to the empirical distribution of the ROTs.

We obtained 29 different surgical procedure groups among the 5 specialties, with significantly different means and standard deviations. Furthermore, we notice that all the fitting mixtures provide an accurate approximation of the sample mean, that ϵ_μ is always less than or equal to 0.1%, while the standard deviation varies from an error $\epsilon_\sigma = 0.5\%$ for GASTRO up to the 19.8% of CARDIO. However, we deem the mined fitting distributions sufficiently adequate for the purpose of generating realistic scenarios.

3.6.2 *Inpatients, outpatients, and emergencies: attributes and costs*

While emergencies are indicated within the considered data set with a special attribute, it is not specified if elective surgeries are inpatient or outpatient. In order to have a realistic inpatient and outpatient population with different surgery duration predictability, no-show rates and costs [142], the attributes of the elective patients have been generated as listed in Table 3.6.

Patients with the lower coefficient of variation ($LCV = \sigma_i/\mu_i = 0.151$) have been labeled as outpatients with a probability of 0.7 and inpatients with a probability of 0.3. Contrarily, patients with the higher coefficient of variation ($HCV = \sigma_i/\mu_i = 0.202$) have been labeled as outpatients with a probability of 0.3 and inpatients with a probability of 0.7. This procedure generated 55% of outpatients and 45% of inpatients, which is an operational context consistent

Table 3.5: Parameters and density probability function graphs of best fitting mixtures and their components with respect to empirical distributions (histograms). Parameters μ and σ are expressed in minutes.

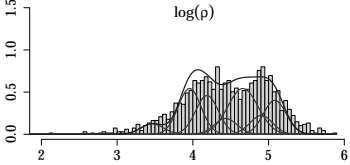
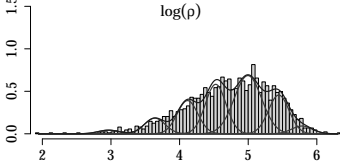
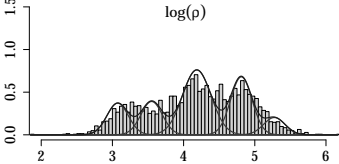
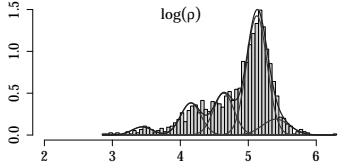
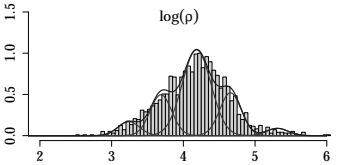
<p>CARDIO (14.32% of total cases) $m = 7$ surgical procedure groups</p> <p>$\mathbf{f} = 0.047, 0.204, 0.172, 0.07, 0.272, 0.083, 0.151$ $\mu = 32.3, 53.4, 67.2, 85.9, 107.9, 136.3, 162.9$ $\sigma = 4.9, 10.8, 13.6, 17.4, 21.8, 20.6, 24.6$ $\epsilon_\mu = 0.001$ $\epsilon_\sigma = 0.198$</p>	
<p>GASTRO (18.76% of total cases) $m = 7$ surgical procedure groups</p> <p>$\mathbf{f} = 0.016, 0.089, 0.224, 0.213, 0.243, 0.182, 0.033$ $\mu = 20.0, 41.6, 72.6, 108.7, 159.7, 234.0, 330.4$ $\sigma = 4.0, 8.4, 14.7, 16.4, 24.1, 35.3, 49.8$ $\epsilon_\mu = 0.000$ $\epsilon_\sigma = 0.005$</p>	
<p>GYN (30.43% of total cases) $m = 5$ surgical procedure groups</p> <p>$\mathbf{f} = 0.206, 0.15, 0.271, 0.328, 0.045$ $\mu = 24.6, 43.8, 68.2, 124.2, 216.4$ $\sigma = 5.0, 6.6, 10.3, 25.1, 32.6$ $\epsilon_\mu = 0.000$ $\epsilon_\sigma = 0.010$</p>	
<p>ORTH (15.90% of total cases) $m = 5$ surgical procedure groups</p> <p>$\mathbf{f} = 0.04, 0.126, 0.239, 0.535, 0.06$ $\mu = 32.9, 63.9, 108.3, 174.3, 243.6$ $\sigma = 6.7, 9.6, 21.9, 26.3, 49.2$ $\epsilon_\mu = 0.000$ $\epsilon_\sigma = 0.008$</p>	
<p>URO (20.59% of total cases) $m = 4$ surgical procedure groups</p> <p>$\mathbf{f} = 0.158, 0.451, 0.36, 0.031$ $\mu = 32.1, 58.1, 94.7, 208.8$ $\sigma = 6.5, 11.7, 19.1, 31.5$ $\epsilon_\mu = 0.000$ $\epsilon_\sigma = 0.041$</p>	

Table 3.6: Parameters of inpatient and outpatient population.

	Inpatient (45% of total cases)	Outpatient (55% of total cases)
$\mathbb{P}(\text{type} \mid \text{LCV})$	0.30	0.70
$\mathbb{P}(\text{type} \mid \text{HCV})$	0.70	0.30
No-show rate (r_i)	0.08	0.24
Scheduling costs (c_i^{sched})	$\mathcal{U}(\{10, 20, \dots, 100\})$	$\mathcal{U}(\{10, 20, \dots, 100\})$
Cancellation costs (c_i^{canc})	$4 c_i^{sched}$	$2 c_i^{sched}$
Waiting costs (c_i^{wait})	$\mathcal{U}([1/180, 1/18])$	$\mathcal{U}([1/360, 1/36])$

with reality [105].

Furthermore, we set the outpatient and inpatient no-show rate to 0.08 and 0.24, which is the average case in the literature, as reported in [142]. The scheduling costs have been generated with uniform distribution over the set $\{10, 20, \dots, 100\}$ for all patients (i.e. $\Delta = 10$). Then, cancellation costs were set equal to twice the scheduling costs for outpatients and to its quadruple for inpatients. The rationale is that scheduling a patient and canceling them is less preferable than not scheduling them from the beginning. Furthermore, a cancellation of an inpatient could have a higher impact because they could occupy other hospital resources (e.g. upstream units) waiting for rescheduling. By contrast, the direct waiting times of outpatients are less preferable than those of inpatients, as confirmed by the high interest in minimizing waiting times in outpatient clinics. The reason is that a delay may have a major impact on outpatient satisfaction. Therefore, we uniformly sampled a waiting cost (per minute) in $[1/180, 1/18]$ for outpatients and $[1/360, 1/36]$ for inpatients in order to have waiting costs for outpatients that are twice the ones of inpatients on average. The scale of these values has been set in order to have a common sense ratio between the waiting costs and the other two patient costs. For instance, we consider the two following scenarios as equivalent in terms of costs: consider an outpatient i with maximum waiting cost and minimum scheduling/cancellation cost

scenario 1: the outpatient i is scheduled, but the actual starting time is 3 h after the planned one;

scenario 2: the same outpatient i is not scheduled at all.

However, it is always preferable to delay the surgery of inpatients rather than scheduling and then canceling them, due to the considerations made above.

Two different cases are analyzed to explore two possible cost preferences of the decision maker from an efficiency perspective, that is $c^g = 1/9, c^h = 1/6$, and $c^g = c^h = 1/3$. In the former, the patient-centered costs have a higher impact on the solution, while the facility-centered objectives have greater importance in the latter. The two cases have also a different mutual

balancing, which is the same as the two scenarios analyzed in [121], where the same real case has been considered.

The daily arrival rate of emergency patients is defined as the ratio between the number of emergency patients in the real data set and the number of weekdays. Thus, we computed the probability to insert an emergency surgery within an OR block, that is $p^{em} = 0.2$. Setting this parameter, we assumed that emergency surgeries can be performed in all ORs, including those for which an OR block is not planned for the current day (e.g. OR 9 on Monday). Furthermore, the parameters of the lognormal distribution used to generate the emergency surgery durations are set according to the empirical mean and standard deviation of the emergency surgical cases in [94], that is $\mu^{em} = 93$ min and $\sigma^{em} = 60$ min. Then, the lognormal distribution has been truncated to a maximum of 240 min.

3.6.3 Instance generation

We generate 10 different instances for each elective surgery waiting list size $|W| = 500$ and $|W| = 1100$. Based on average results performed in a preliminary analysis, we need about 3 and 5 weeks to schedule all patients within the fixed MSS, respectively. This means that if the number of executed surgeries per week is approximately the same number of new patients inserted within the waiting list, then the average indirect waiting time will be about 3 and 5 weeks, respectively. The patient population of each instance is randomly generated following the empirical distribution of elective patients among the 5 specialties. At this point, we consider the set Π_s of all surgical procedure groups of the patient's specialty $s \in S$. Then, a procedure group $\pi \in \Pi_s$ is generated with the categorical distribution of parameters $\mathbf{f} = (f_1, \dots, f_m)$, where $m = |\Pi_s|$, as reported in Table 3.5. By consequence, the EOT is set equal to the mean μ_π of the lognormal distribution associated with the surgical procedure group π , which means that in the scenario generation the ROT will have a lognormal distribution of mean μ_π and standard deviation σ_π .

In all instances, the planning horizon is one week. We set a total duration of $L_{jk} = 480$ min for all OR blocks $(i, j) \in B$, with a maximum overtime of $H = 60$ min per OR block. The same instances have been used to compare different configurations of the model parameters and the proposed methods.

3.6.4 Tests

For each instance, a sample \mathcal{S} of size $|\mathcal{S}| = 1000$ has been generated and used for the Monte Carlo sampling, the SAA, the SAA_N , and the BRKGA. All methods have been implemented in `Python 3.10`, with `Gurobi 10.0.0` for the mathematical programming and the `Pymoo` Python library [26] for the BRKGA. All tests have been run on an HPC cluster running `CentOS`, using 4 `Intel` CPU cores working at 2.1 GHz and 16 GB of RAM to simulate the performance of a standard desktop computer that may be used by an OR

planner.

3.7 Computational analysis

We present a computational analysis to evaluate the proposed approaches for the two phases of the stochastic optimization problem introduced in Section 3.3. In Section 3.7.1, we present the results for the advance scheduling model $\mathcal{A}(\alpha, \beta)$ with the Monte Carlo sampling by fixing β , in order to prove the effectiveness such an approach in for the first phase of our framework and to evaluate the impact of the robustness parameter and the three proxies. The solutions are then used in Section 3.7.2 to compare the performance of the SAA, the SAA_N , and the BRKGA to solve the allocation scheduling models \mathcal{B}_{jk} in the second phase of the framework. Solutions obtained in these experiments are then combined in Section 3.7.3, where we present the results of the parameter variation illustrated in Figure 3.1 discussing the trade-off between robustness and amount of scheduled surgeries, and between the criteria defined through the proxies. Furthermore, in Section 3.7.4 we present an analysis focused on the patient type to provide general managerial insights for the scheduling of inpatients and outpatients within shared ORs. Finally, a sensitivity analysis is presented in Section 3.7.5 to evaluate the level of approximation introduced by the assumption regarding the possibility of inserting at most one emergency patient per OR block.

3.7.1 Results: advance scheduling

As a first analysis, we solve the model $\mathcal{A}(\alpha, \beta)$ with the Monte Carlo sampling by ranging $\alpha \in \{0.05, 0.10, 0.15, 0.20\}$ to set different levels of robustness and $\beta \in \{(1, 0, 0), (0, 1, 0), (0, 0, 1)\}$ to consider the three proxies one by one and provide a comparison between the different patient mixes determined by them. For each instance we set a total time limit of 1 h, which means that we need half day to compute all the solution for the fixed parameters, that is a reasonable maximum running time for a weekly schedule.

In Table 3.7, we report results for the two instance sets with a different number of patients ($|W| = 500$ and $|W| = 1100$) to be scheduled by varying the model parameters. Model $\mathcal{A}(\alpha, \beta)$ has been solved by its decomposition in independent subproblems for the 5 specialties ($\mathcal{A}_s(\alpha, \beta)$, with $s \in \{\text{CARDIO}, \text{GASTRO}, \text{GYN}, \text{ORTH}, \text{URO}\}$), then the objective function value (o.f.) has been computed by summing the objective function values of the 5 specialties and the gap (gap) has been computed accordingly. Firstly, we can notice that the use of a general-purpose solver like **Gurobi** is appropriate, although Monte Carlo sampling increases the computational complexity of the programming model. Average gaps range indeed between 0.26% up to 6.65%, with a general pattern that indicates a greater complexity for lower values of α . For the same reason, when α increases the solver is able to find the optimal solution of some instances within the given time limit. Then, the average running time (secs)

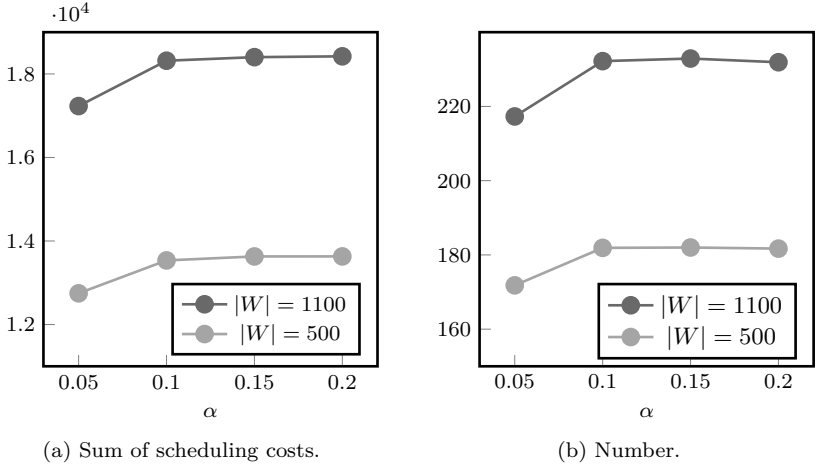


Figure 3.5: Scheduled patients (sum of the costs of scheduled surgeries and their number) varying the robustness parameter α (best solutions among values of β).

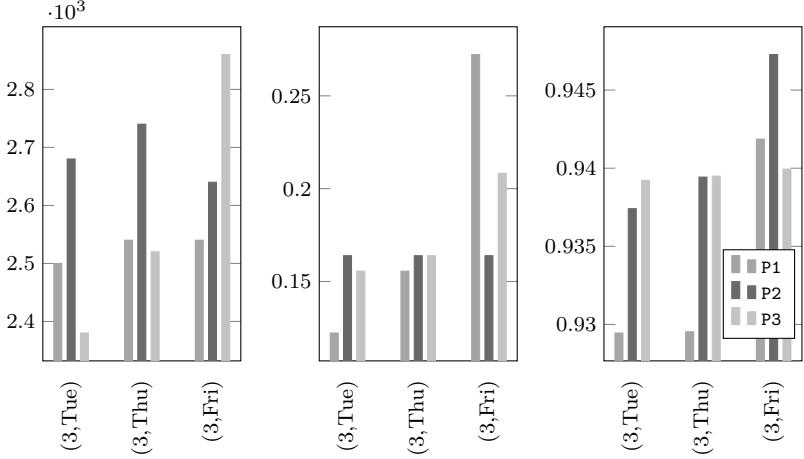
decreases, accordingly.

As expected, the lower the value of α , the lower the sum of the scheduling costs of scheduled patients (**Sched. patients cost**). However, while a significant difference can be observed between $\alpha = 0.05$ and $\alpha = 0.10$, these values tend to settle. This can be better observed in Figure 3.5, where for each value of α we considered the best configuration β with respect to the total scheduling cost. A counterintuitive result is that, while the cost of scheduled patients slightly increases, the number of scheduled surgeries (**Sched. patients number**) is almost the same for $\alpha = 0.10, 0.15, 0.20$, indicating that the chance constraints have a greater effect on the selection of surgeries to be scheduled than on their quantity. Negligible differences can be observed in the objective function values for $\alpha = 0.15$ and $\alpha = 0.20$. This indicates that, around the last value of the robustness parameter, we reach the critical point where the chance constraints (3.1d) become less strict than the deterministic capacity constraints (3.1c). Then, for the sack of simplicity, we will show only results for $\alpha \in \{0.05, 0.10, 0.15\}$ in the second phase of the stochastic optimization approach.

Another expected fact is that different proxies lead to very similar average objective function values, due to the hierarchical modeling of the objectives. Slight fluctuations in the total scheduling costs (column **Sched. patients cost**) in tests with the same values of W and α are mainly due to different complexity caused by different weights in the objective function (3.1a). However, in the worst case, we have a difference in the total scheduling costs equal

Table 3.7: Results solving $\mathcal{A}(\alpha, \beta)$ with different model parameters (average values over 10 instances).

$ W $	α	β_1	β_2	β_3	o.f.	P1	P2	P3	Sched. patients			
									secs	cost	number	gap
500	0.05	1	0	0	14608.08	8568.00	5.00	185.40	3600	12747	171.8	6.58%
		0	1	0	14607.82	9874.00	0.84	205.60	3600	12743	170.5	6.64%
		0	0	1	14602.16	9402.00	5.00	154.14	3600	12740	171.6	6.65%
		1	0	0	13820.58	8876.00	4.83	178.34	3335	13535	181.9	1.21%
	0.10	0	1	0	13817.58	10410.00	0.90	199.40	3390	13534	182.2	1.23%
		0	0	1	13810.72	10060.20	4.91	99.61	3379	13531	181.9	1.28%
		1	0	0	13746.33	8748.00	4.91	185.52	3206	13609	182.9	0.65%
		0	1	0	13720.60	10254.00	0.82	208.20	3102	13630	182.0	0.48%
	0.15	0	0	1	13725.61	10026.20	5.00	89.37	3152	13616	182.9	0.65%
		1	0	0	13738.32	8746.00	5.00	172.30	3000	13617	182.5	0.65%
		0	1	0	13719.68	10526.10	0.83	215.50	2875	13631	181.7	0.46%
		0	0	1	13717.53	10126.10	4.92	76.92	3192	13624	183.2	0.59%
1000	0.05	1	0	0	43722.09	11432.00	5.00	196.40	3600	17225	217.5	3.03%
		0	1	0	43722.86	12676.00	1.01	180.50	3600	17218	216.9	3.04%
		0	0	1	43700.26	12424.00	5.00	168.00	3600	17231	217.3	3.03%
		1	0	0	42637.07	12120.00	5.00	169.00	3144	18311	231.2	0.54%
	0.10	0	1	0	42632.28	13624.00	1.05	179.50	3145	18309	231.3	0.54%
		0	0	1	42613.69	13444.00	4.92	105.75	3175	18317	232.2	0.54%
		1	0	0	42572.77	11920.00	5.00	165.60	2880	18375	231.8	0.38%
		0	1	0	42548.07	13596.10	1.03	187.30	2793	18393	231.4	0.35%
	0.20	0	0	1	42528.48	13286.10	5.00	73.13	2865	18402	232.9	0.33%
		1	0	0	42552.75	11896.00	5.00	164.20	2890	18395	232.0	0.33%
		0	1	0	42518.83	14028.00	1.01	173.10	2799	18422	231.9	0.26%
		0	0	1	42533.54	13394.10	5.00	82.96	2921	18397	232.0	0.33%



(a) Sum of canc. costs. (b) Sum of waiting costs. (c) Exp. OR utilization.

Figure 3.6: Proxy variables of CARDIO OR blocks for different values of β (single instance with $|W| = 1100$ and $\alpha = 0.1$). P1, P2, and P3 indicate the configurations with $\beta = (1, 0, 0)$, $\beta = (0, 1, 0)$, and $\beta = (0, 0, 1)$, respectively.

to 27, which corresponds to the scheduling of a low-priority surgery.

The effectiveness of the three proxies in determining different types of patient mixes is evident. In columns P1, P2, and P3 we report the average values of the 3 proxies, computed as the sum of the proxy variables Γ^c , Γ^w , and Γ^t over the 5 specialties. Such values clearly show that proxy variables are significantly reduced when the corresponding component in β is equal to 1 with respect to 0.

The behavior of the three proxies can be further observed in Figures 3.6 and 3.7, where we reported the distribution of costs and surgery time among OR blocks for a single instance with $|W| = 1100$, $\alpha = 0.1$, and two different specialties. In Figure 3.6(a) and Figure 3.7(a) we show the sum of cancellation costs of each OR block of the considered specialties, that is defined as the expression in the argument of maximum function in constraint (3.1e). Analogously, in Figures Figure 3.6(b) and Figure 3.7(b) we represent the sum of waiting costs of each OR block that refers to constraint (3.1f). Finally, in Figures Figure 3.6(c) and Figure 3.7(c) we report the expected OR utilization for every OR block, which is equal to the variable u_{jk} in constraint (3.1i) divided by the OR block capacity $L_{jk} = 480$.

By enabling the first proxy ($\beta = (1, 0, 0)$) or the second proxy ($\beta = (0, 1, 0)$) we notice that the maximum sum of cancellation costs and the maximum sum of waiting costs of surgery within the same OR block are reduced

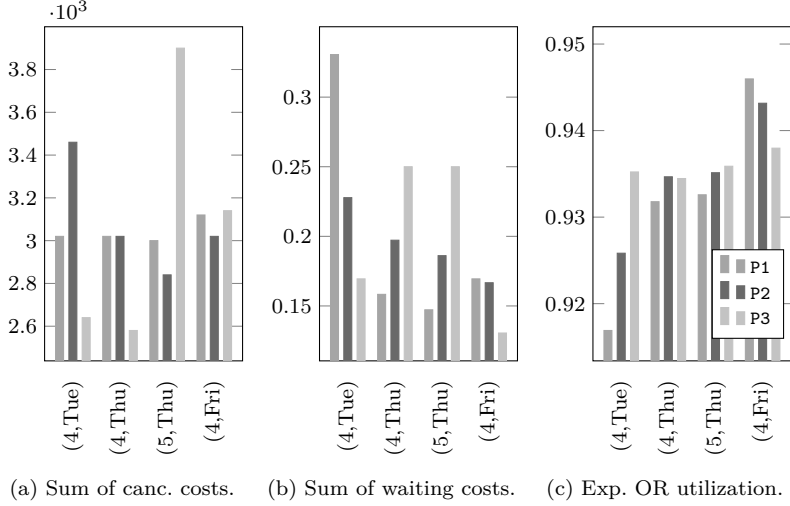


Figure 3.7: Proxy variables of single URO OR blocks for different values of β (single instance with $|W| = 1100$ and $\alpha = 0.1$). P1, P2, and P3 indicate the configurations with $\beta = (1, 0, 0)$, $\beta = (0, 1, 0)$, and $\beta = (0, 0, 1)$, respectively.

with respect to the other configurations. As a side effect, the distribution of the considered costs is balanced among the OR block of the same specialty. For instance, since the OR utilization is independent of the patients' cancellation/waiting costs, the balancing of the OR utilization provided by the parameter configuration $\beta = (0, 0, 1)$ could lead to OR blocks whose patient mix involves a high sum of such costs (e.g. see OR block (3,Fri) for CARDIO in Figures 3.6(a)–(b)). Similarly, setting β to consider only the first proxy could lead to a poor solution with respect to the second proxy (e.g. see OR block (3,Fri) for CARDIO in Figure 3.6(b) or OR block (4,Tue) for URO in Figure 3.7(b)), and viceversa (e.g. see OR block (3,Fri) for CARDIO in Figure 3.6(b) or OR block (4,Tue) for URO in Figure 3.7(b)).

Similar considerations can be made for the third proxy in Figures 3.6(c)–3.7(c). In the parameter configurations with $\beta_3 = 0$, we can observe that the expected OR utilization ranges in an interval of length equal to approximately the 1%–1.5% for CARDIO and 1.5%–3% for URO. When the third proxy is enabled ($\beta = (0, 0, 1)$) the CCIP model is able to find solutions with an almost perfect balancing. Furthermore, an interesting behavior can be observed in Figure 3.6(c) where the solution in correspondence of $\beta = (1, 0, 0)$ has a lower estimated OR utilization with respect to that with $\beta = (0, 1, 0)$ OR block by OR block, and with respect to that with $\beta = (0, 0, 1)$ on average. This is one of the cases where there exist more solutions with the minimum total scheduling

costs at the higher level of the hierarchical function, then the weights of the proxy variables have an impact also on the decision about the patients to be scheduled. As a consequence, we expect that the solution provided by $\beta = (1, 0, 0)$ for this particular instance will lead to a higher total idle time cost but a lower overtime cost compared to $\beta = (0, 1, 0)$ and $\beta = (0, 0, 1)$.

Finally, we observe that for the longer waiting list ($|W| = 1000$) the best solutions consist of a higher number of scheduled patients, whose sum of scheduling costs is also greater. Although in the optimal solutions for the shorter waiting list ($|W| = 500$) a substantial fraction of surgeries is not scheduled, the increase of patients to be scheduled allows us to find better solutions because (i) there are more patients with a more advantageous ratio between scheduling cost and surgery duration and (ii) there are more possible patient mixes that comply with the capacity constraints (3.1c)–(3.1d).

3.7.2 Results: allocation scheduling

We compare the effectiveness of SAA, SAA_N , and BRKGA to solve the SMIP models \mathcal{B}_{jk} , with $(j, k) \in B$, starting from the advance schedules found in the experiments discussed in Section 3.7.1, and by considering the two different cost preferences $c_g = 1/9$, $c_h = 1/6$ and $c_g = c_h = 1/3$. We fix a total time limit of 5 min for solving every method, for an overall time limit of 130 min per parameter configuration to solve the allocation scheduling for all the weekly OR blocks of Table 3.4. We remark that the common practice is to solve the allocation scheduling day-by-day, which in our experimental setup means a required daily maximum running time of 20–30 min per parameter configuration.

A preliminary tuning procedure through a uniform grid search allowed us to set the BRKGA parameters, that is a population size of 50, a fraction of elite and mutant population of $1/4$ and $1/5$, respectively, and an elite gene inheritance probability $\tilde{p} = 1/2$.

Since the effectiveness of the proposed approaches is strongly dependent on the number $|I_{jk}|$ of scheduled surgeries in the considered OR block $(j, k) \in B$, we present results divided into groups with similar dimensions of the subinstances used to solve the subproblems \mathcal{B}_{jk} for all parameter configurations discussed so far (except the case $\alpha = 0.2$ for the reason discussed above). In Figure 3.8 we show the distribution of the subinstances with respect to the number of surgeries scheduled in the same OR block, where we can observe a substantial difference between those related to the instance sets with $|W| = 500$ and $|W| = 1100$, which leads to the different average number of scheduled patients already discussed in Section 3.7.1. To present meaningful and concise results, all subinstances have been divided into 4 groups with respect to the 4 quartiles determined by the value of I_{jk} , that is 3–6, 7–8, 9–10, and 11–17.

Results of the general performance of the three methods are reported in Table 3.8, with the best value of N ($N = 10$) for the SAA_N . Average values of

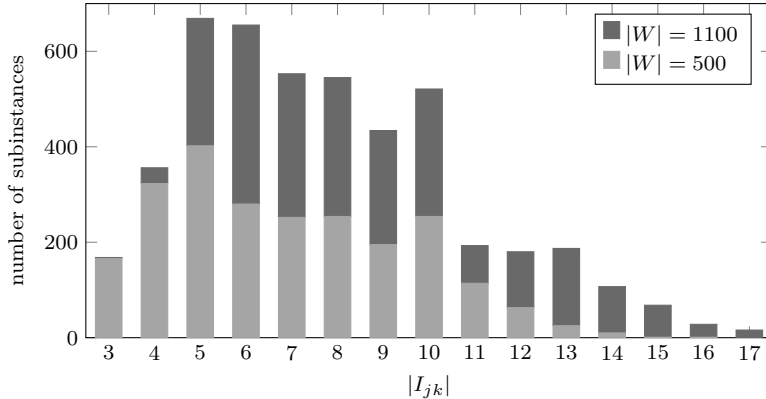


Figure 3.8: Distribution of the cardinality of the sets I_{jk} (i.e., number of patients scheduled in the same OR block) generated with all parameter configurations.

objective function values and running times are divided into groups of subinstances. The first column (**costs**) indicates the two cost preferences chosen for this analysis. In the second column ($|I_{jk}|$) we divide the subinstances with respect to the 4 quartiles determined by the value of $|I_{jk}|$. Then, for each subset \mathcal{I} of subinstances, the third column (**group**) considers 3 further subsets of subinstances for a fair comparison: \mathcal{I}_1 identifies the subinstances for which all three methods provided feasible solutions, \mathcal{I}_2 contains the subinstances for which both the SAA_N and the BRKGA found a feasible solution, and \mathcal{I}_3 is the set of subinstances such that the BRKGA was able to provide a feasible solution (it holds that $\mathcal{I}_1 \subset \mathcal{I}_2 \subseteq \mathcal{I}_3 \equiv \mathcal{I}$). The cardinality of groups $\mathcal{I}_1, \mathcal{I}_2$, and \mathcal{I}_3 is reported in the fourth column (**# inst.**).

Contrary to what happens for the advance scheduling model, the SAA is inadequate to solve the proposed SMIP model, with only 1.2% of the subinstances solved, all for lowest values of $|I_{jk}|$. In all other cases, the ILP model implemented in Gurobi for the SAA method reaches the time limit before finding a feasible solution or the computation ends due to out-of-memory, as shown in Figure 3.9 for the two cost preferences and varying the number of scheduled patients. Feasible solutions are found within the time limit only for OR blocks with no more than $|I_{jk}| = 6$ patients, and the provided solutions are poor with respect to those of the other approaches.

The SAA_N is instead able to provide solutions for the 64.9% of all subinstances, and for all except one of those with $|I_{jk}| \leq 6$, for which provides results that are slightly better than those of the BRKGA. However, the fraction of solved subproblems decreases when the number of scheduled patients within the OR block increases, as shown in Figure 3.10(a). Similarly, the qual-

Table 3.8: Objective function values (o.f.) and running time (secs) of the SAA, the SAA_N , and the BRKGA for the different cost preferences and groups of subinstances.

costs	$ I_{jk} $	group	# inst.	SAA		SAA_N		BRKGA
				o.f.	secs	o.f.	secs	o.f.
$c_g = 1/9$ $c_h = 1/6$	3-6	\mathcal{I}_1	58	28.24	300.00	22.18	65.08	23.17
		$\mathcal{I}_2 \equiv \mathcal{I}_3$	1848	-	-	22.16	300.00	22.47
	7-8	\mathcal{I}_1	0	-	-	-	-	-
		\mathcal{I}_2	813	-	-	35.56	300.00	26.85
		\mathcal{I}_3	1098	-	-	-	-	25.67
	9-10	\mathcal{I}_1	0	-	-	-	-	-
		\mathcal{I}_2	337	-	-	44.70	300.00	30.79
		\mathcal{I}_3	955	-	-	-	-	28.60
	11-17	\mathcal{I}_1	0	-	-	-	-	-
		\mathcal{I}_2	57	-	-	46.93	300.00	27.63
		\mathcal{I}_3	779	-	-	-	-	26.17
$c_g = 1/3$ $c_h = 1/3$	3-6	\mathcal{I}_1	57	54.04	300.00	44.37	66.68	45.37
		\mathcal{I}_2	1847	-	-	43.97	300.00	44.12
		\mathcal{I}_3	1848	-	-	-	-	44.12
	7-8	\mathcal{I}_1	0	-	-	-	-	-
		\mathcal{I}_2	755	-	-	55.67	300.00	45.02
		\mathcal{I}_3	1098	-	-	-	-	44.19
	9-10	\mathcal{I}_1	0	-	-	-	-	-
		\mathcal{I}_2	350	-	-	63.62	300.00	47.22
		\mathcal{I}_3	955	-	-	-	-	45.94
	11-17	\mathcal{I}_1	0	-	-	-	-	-
		\mathcal{I}_2	63	-	-	61.56	300.00	43.77
		\mathcal{I}_3	779	-	-	-	-	42.93

ity of the provided feasible solutions gets worse when $|I_{jk}|$ increases, as shown in Figure 3.10(b), where we compare the SAA_N and the BRKGA in terms of relative difference. This suggests using the proposed SMIP-based approach for OR blocks with a small number of patients, while a metaheuristic is necessary for larger instances.

As highlighted in Section 3.4.3, the encoding of the BRKGA has been designed in such a way to provide only feasible solutions and to exploit the whole running time exploring the search space. On the subinstance groups that allow the comparison with the SAA_N (i.e. \mathcal{I}_2), results suggest that the metaheuristic is able to provide near-optimal solutions whose quality is not significantly affected by the instance size. This can be viewed through a comparison with the SAA_N : while the average objective function values increase when using such a method over increasing value of $|I_{jk}|$, the BRKGA provides solutions with small fluctuations. This seems to mean that the number of surgeries to be executed within the same OR block does not significantly affect the weighted

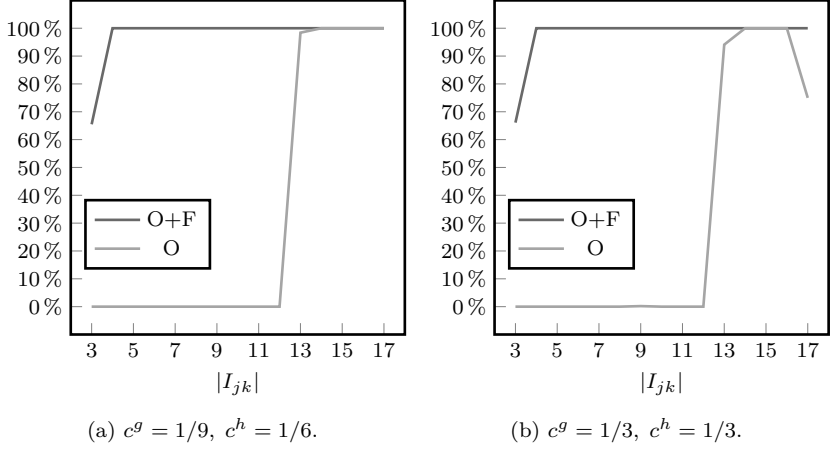


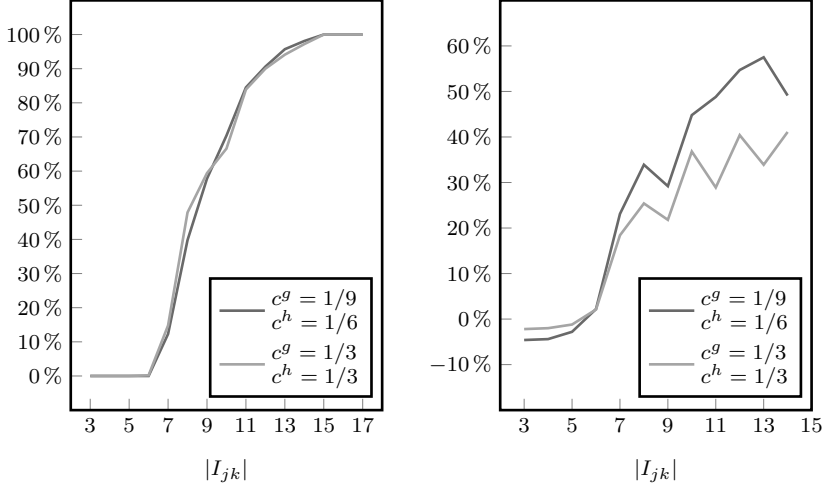
Figure 3.9: Undesiderable behavior of SAA for the two cost preferences: percentage of instances that caused out-of-memory (O) issues and percentage of instances that caused out-of-memory issues or for which no feasible solution was found within the time limit (O + F).

sum of idle time, overtime, waiting, and cancellation costs. Consequently, the higher the number of patients, the higher the difference between the average objective function values of the SAA_N and the BRKGA. No relevant difference emerges between the two different cost preferences.

3.7.3 Trade-off among objectives: parameter variation

We execute a parameter variation experiment by selecting the solution with the tighter gap of the overall objective function Z in (3.4) for every single instance after solving all the defined subproblems. For every instance and specialty, we get 9 different solutions of the assignment procedure, with different levels of robustness and patient mix within the OR blocks, by using the CCIP model $\mathcal{A}(\alpha, \beta)$ with three different values of α and three different values of β , as reported in Table 3.9. We combine each of those solutions with the best obtained by solving the SMIP model \mathcal{B}_{jk} through the proposed approaches (i.e. between the SAA_N and the BRKGA). At the end, for every OR block we obtain 9 alternative final surgical schedules. Among those, we choose the one with the minimum value of Z , that is the minimum weighted sum of the 5 defined costs (scheduling, cancellation, waiting, idle time, and overtime).

In Figure 3.11 we show the value of Z decomposed into the five different types of costs for the parameter configurations in Table 3.9, the two instance sets, and the two different cost preferences. The robustness parameter α gives the most important contribution in determining the better parameter configu-



(a) SAA_N: no feasible solution found within time limit.

(b) BRKGA vs. SAA_N: objective function difference.

Figure 3.10: Comparison between SAA_N and BRKGA varying the subinstances size $|I_{jk}|$: (a) percentage of subinstances for which the SAA_N does not find feasible solution within the time limit; (b) relative difference of the objective function value found with the BRKGA over those with the SAA_N (missing values $|I_{jk}| = 16, 17$ are due to no of feasible solution found within the time limit by the SAA_N).

Table 3.9: IDs and parameters' values of the configurations set for the parameter variation.

ID	A1	B1	C1	A2	B2	C2	A3	B3	C3
α	0.05	0.10	0.15	0.05	0.10	0.15	0.05	0.10	0.15
β	(1, 0, 0)			(0, 1, 0)			(0, 0, 1)		

rations: the lower the robustness, the lower the overall objective function value Z . In fact, differences in the total scheduling cost between more and less robust advance schedules are not sufficiently compensated by lower overtime and cancellation costs got in the allocation schedules. Furthermore, small differences are obtained by enabling the three different proxies. Fixed the value of α , the best configuration is always one between $\beta = (1, 0, 0)$ and $\beta = (0, 0, 1)$ due to the reduction of the cancellation costs, which are the ones with the higher variability among configurations. As a consequence, the configuration $\alpha = 0.15$ and $\beta = (1, 0, 0)$ is the best for the instance set with $|W| = 500$ and

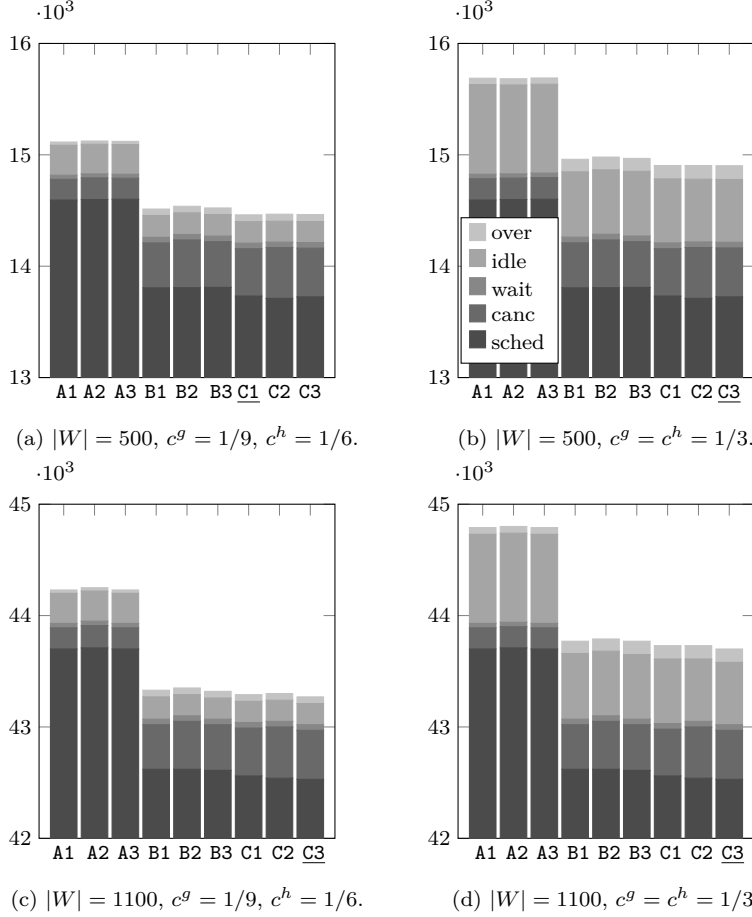


Figure 3.11: Comparing the 5 different costs and the overall objective function value Z for different model parameter configurations (best is underlined).

cost preference $c^g = 1/9$, $c^h = 1/6$, while $\alpha = 0.15$ and $\beta = (0, 0, 1)$ provides better results in all other cases.

We highlight the differences between final solutions provided by the use of different proxies in the advance allocation model in Figures 3.12 to 3.14, by reporting results of the instance set with $|W| = 1100$ as representative of the general behavior.

In Figure 3.12 we represent the higher level of the hierarchical objective function of $\mathcal{A}(\alpha, \beta)$ on the x-axis (**scheduling costs**) and the objective func-

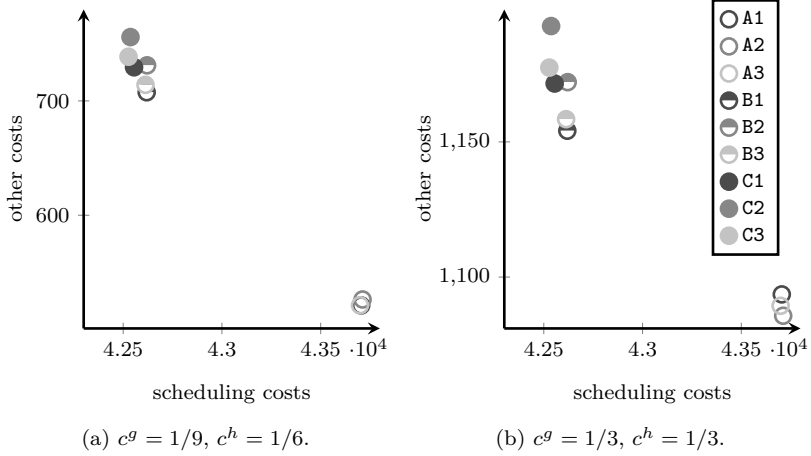


Figure 3.12: Trade-off between scheduling costs and other costs ($|W| = 1100$).

tion of β on the y-axis (**other costs**) whose sum is Z . Here, the trade-off between the minimization of the indirect waiting time and the ensemble of other objectives is more evident: for a given value of β , it is not possible to identify a value of α that is Pareto-dominated by another. Conversely, different values of α lead to a different configuration of β that minimizes the overall objective function. By enabling the first proxy ($\beta = (1, 0, 0)$), we minimize the objective function value of β for $\alpha \geq 0.15$, but not for $\alpha = 0.05$ (e.g. see Figure 3.12(b)).

Nevertheless, it is important to highlight that also for the higher values of α , the enabling of the third proxy ($\beta = (0, 0, 1)$) provides a lower overall objective function value. Theoretically, the hierarchical objective function of model $\mathcal{A}(\alpha, \beta)$ should always provide the same scheduling costs for a fixed value of α . However, the high complexity of the advance scheduling model and the setting of a time limit due to operational reasons lead to different scheduling costs. Consequently, although the advance schedule obtained enabling the first proxy seems to perform better in the allocation scheduling phase, at the end of the day it is more convenient to adopt the third one because it provides lower overall costs. We could define this phenomenon as the *cost of complexity*, which leads to choosing a parameter configuration that is conditioned by the fixed time limit and the complexity given by the different objective function coefficients.

The positive impact of robustness over the direct waiting time and the frequency of cancellations can be observed in Figure 3.13. Restricting the evaluation on these two patient-centered indices, each of the parameter configurations with the highest robustness ($\alpha = 0.05$) is Pareto-optimal for the

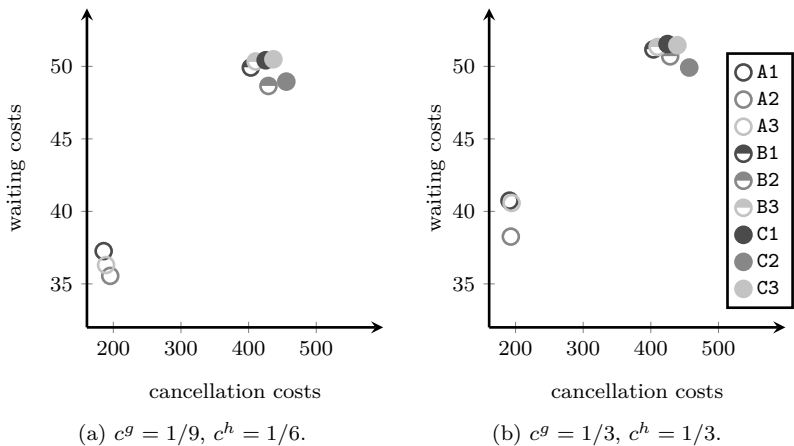


Figure 3.13: Trade-off between cancellation costs and waiting costs ($|W| = 1100$).

first cost preference (Figure 3.13(a)), with the third proxy that is a compromise between the other two. When α is fixed, the same trade-off among proxies occurs in other cases (e.g. see configurations C1, C2, and C3 in Figure 3.13(b)), while in other cases the solution provided by enabling the third proxy is Pareto-dominated by one of those obtained from the other two proxies (e.g. in both graphs in Figure 3.13, B3 is dominated by B1).

Since we assigned a significantly higher weight to cancellations with respect to reasonable direct waiting times and because of a larger order of magnitude of the interval in which the resulting costs range, the first proxy has a more important impact on both the objective function of \mathcal{B} and the overall objective function Z . We would say that in general waiting costs are not able to cope with cancellation costs, so when both objectives are optimized simultaneously, an implicit hierarchy is created between them.

With greater reason, the minimization of both direct and indirect waiting times could make sense only in a hierarchical way when inpatients and outpatients share ORs in operational contexts similar to the one considered in our analysis. Since very long direct waiting times are needed to be reasonably compared to the non-scheduling of a patient (i.e. to a higher indirect waiting time), it is unlikely to find two Pareto-optimal solutions with different scheduling costs. We expect that this can be extended to many operational contexts in public health systems unless the unlikely situation in which available OR capacity or overtime is sufficient to guarantee very short waiting lists.

Finally, the two facility-centered indices are represented on the two axis of graphs in Figure 3.14. As expected, a higher idle time and lower overtime correspond to a more robust advance schedule. Conversely, when we weaken

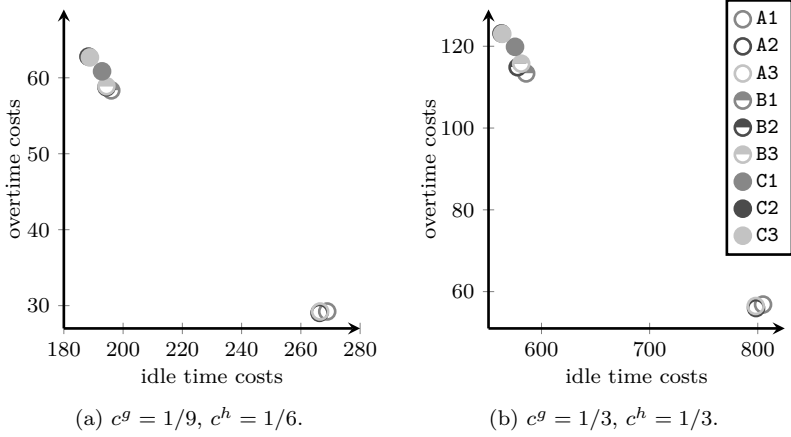
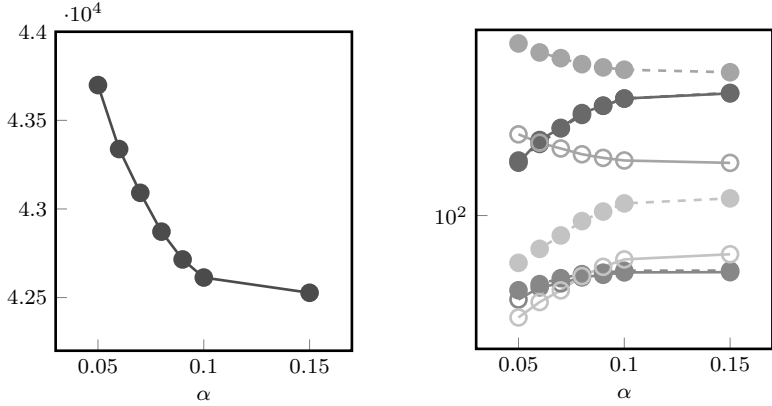


Figure 3.14: Trade-off between idle time costs and overtime costs.

the chance constraints (3.1d) by increasing α , there is a higher expected overtime but the ordinary OR block duration is also exploited more efficiently. A further contribution to this strong trade-off is also given by the parameter vector β . When $\alpha \geq 0.15$, the first proxy promotes more idle time and less overtime than the other two proxies, which almost completely overlap.

We observe that the third proxy, which balances the expected surgery time among the OR blocks of the same specialty, does not outperform the other two proxies with respect to idle time and overtime. Although the balancing of the cancellation costs and waiting costs has not been designed to act on these two objectives, it emerges that such proxies offer a good balance of the workload among OR blocks as a side effect. Nevertheless, the configuration $\beta = (0, 0, 1)$ is able to provide the minimum value of Z in 3 out of 4 of the instance sets analyzed.

Finally, we design further parameter configurations by fixing $\beta = (0, 0, 1)$ (i.e. the best configuration for instance sets with $|W| = 1100$) and ranging $\alpha \in [0.05, 0.1]$ with step 0.01. This choice is guided by the observation of rather similar behaviors in the costs determined with $\alpha = 0.1$ and $\alpha = 0.15$ compared to the substantial difference with $\alpha = 0.05$. Therefore, we enhance our analysis by setting further robustness levels that allow us to evaluate in more detail how the different costs evolve before stabilizing. Results are illustrated in Figure 3.15, where we can observe a monotonic trend for all considered criteria in our multi-objective optimization approach.



(a) Scheduling costs (independent of cost preferences). (b) Other costs (solid: $c^g = 1/9, c^h = 1/6$, dashed: $c^g = c^h = 1/3$).

Figure 3.15: Impact of robustness – further values of α ($|W| = 1100, \beta = (0, 0, 1)$). Curve colors refer to costs as in the legend of Figure 3.11.

Table 3.10: Schedule indices of the best parameter configuration ($|W| = 1100$).

costs	f_{in}	f_{out}	GI_{avg}	\widehat{GI}	q_{in}^{avg}	q_{out}^{avg}
$c^g = 1/9, c^h = 1/6$	0.5554	0.4446	0.4352	0.8813	118.0	322.7
$c^g = c^h = 1/3$					120.5	322.7

3.7.4 Inpatient and outpatients: general insights

Starting from the best parameter configurations found in Section 3.7.3, we can analyze decisions concerning the selection, the mix, and the sequencing of the two different types of elective patients, that is inpatients and outpatients. We report results about the instance sets with $|W| = 1100$, but similar conclusions can be made for $|W| = 500$. In Table 3.10 we report indices that give us summarizing information about the selection and the mix of patients within the OR blocks.

The first indices are the fraction of inpatients f_{in} and outpatients f_{out} over the total, which deviates slightly from their distribution within instances to the advantage of the former. Furthermore, we compute the average Gini Index (GI), which is used in machine learning to evaluate the impurity of a set whose elements can have different labels. Given an OR block $(j, k) \in B$, such an index is computed as follow

$$GI_{jk} = 1 - \left(\frac{n_{in}}{|I_{jk}|} \right)^2 - \left(\frac{n_{out}}{|I_{jk}|} \right)^2,$$

where n_{in} are n_{out} the number of inpatients and outpatients scheduled in the OR block (j, k) , with $n_{in} + n_{out} = |I_{jk}|$. Then, we compute the average GI over the whole set B and we get the value $GI_{avg} \in [0, 1/2]$, which is equal to 0 if the two classes of patients are perfectly separated (i.e. inpatients and outpatients are never scheduled into the same OR block) and rise up to $1/2$ when they are mixed. However, since the two classes of patients do not have the same cardinality, we can not have $1/2$ as the maximum value of GI, then we computed the value $GI_{max} \leq 1/2$ as an upper bound of the maximum average GI, that is in correspondence of $n_{in} = f_{in}|I_{jk}|$ for each $(j, k) \in B$ (it is an upper bound because the quantity $f_{in}|I_{jk}|$ can be fractional, while $n_{in} \in \mathbb{N}_0$). Finally, we compute the normalized index $\widehat{GI} = GI_{avg}/GI_{max}$ that represents an impurity index taking into account the distribution of the two scheduled patient classes. As a result, we obtain an average GI equal to 0.4352 and $\widehat{GI} = 0.88$, which indicate an almost perfect mix of inpatients and outpatients in all the OR blocks. Furthermore, in Figure 3.16 we represents the values of f_{in} and \widehat{GI} corresponding to all parameter configurations over the same instance set. Interestingly, the best configuration (C3) is also the one with the highest value of \widehat{GI} , suggesting that an adequate balancing of inpatients and outpatients in the same OR block provides better performance than allocating different OR blocks to the two class of surgeries. More generally, we can observe that higher levels of robustness lead to lower GIs, while proxies that promote the minimization of cancellation costs and waiting costs lead to a higher and a lower number of scheduled inpatients, respectively. In general, results suggest that a grouping policy that separates inpatient and outpatient surgeries in dedicated OR block is far from being a good strategy with respect to the considered criteria and their weights.

The last managerial insight emerging from our analysis is about the sequencing and the timing of the two classes of patients. The average scheduled starting times q_{in}^{avg} and q_{out}^{avg} of inpatients and outpatients are very different, suggesting a strong preference for scheduling inpatients at the beginning of the OR block and outpatients in the last positions. This behavior can be observed more in detail in Figure 3.17, where the two distributions of the scheduled starting time are significantly different. This can be motivated by the unbalanced weights of cancellations and direct waiting time already discussed and shown in Figure 3.13. Since surgeries scheduled at the end of the OR block are affected by both a higher probability of cancellation and a higher expected direct waiting time, the higher costs of cancellations with respect to the waiting costs promote the precedence of inpatients over outpatients. The effect is similar to that of pooling, which is a common strategy used to sequence surgeries with different characteristics [141]. Furthermore, the pooling effect is slightly attenuated by increasing idle time and overtime costs (e.g. comparing the boxplots of $c^g = c^h = 1/3$ with those of $c^g = 1/9, c^h = 1/6$), which reduces the weight of cancellations with respect to the overall objective function Z .

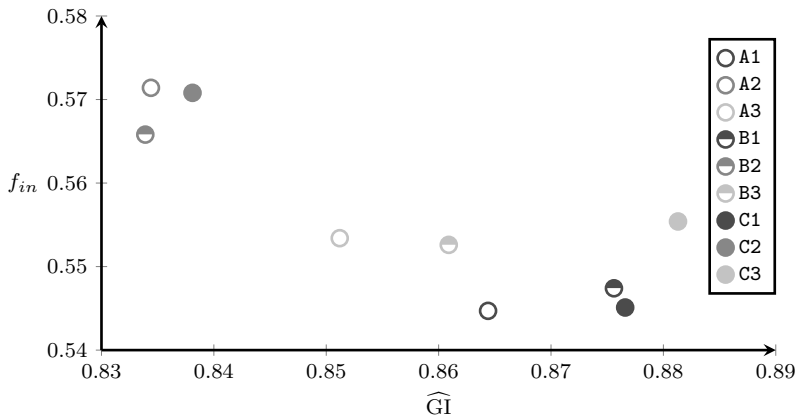


Figure 3.16: Comparing the fraction of the scheduled inpatients and the normalized average GI of all parameters configurations.

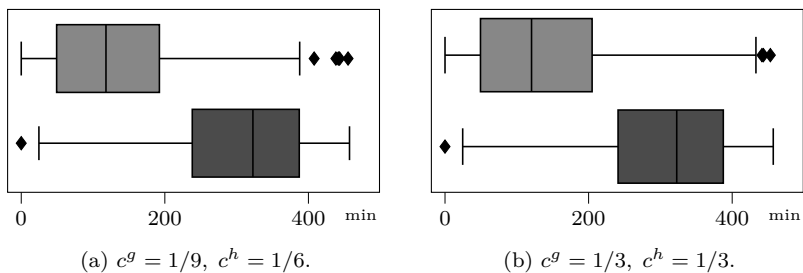


Figure 3.17: Boxplots representing scheduled starting time for inpatients (light gray) and outpatients (dark gray).

The impact of positioning inpatients before outpatients in the same OR blocks on waiting times and cancellations is shown in Figures 3.18 and 3.19, respectively. Although the average direct waiting times of inpatients and outpatients differ by only 2 min, the distributions are significantly different, with a higher variance in the direct waiting times of inpatients. This counter-intuitive behavior could be caused by the impact of the insertion of emergency patients. Although they have an immediate consequence on patients scheduled immediately after their arrival, the delay caused could be mitigated by slack times and no-shows. As a result, patients at the end of the session (usually outpatients) could benefit from this mitigation, while there is less margin for previous ones.

Contrarily, an expected consequence arises about the probability of surg-

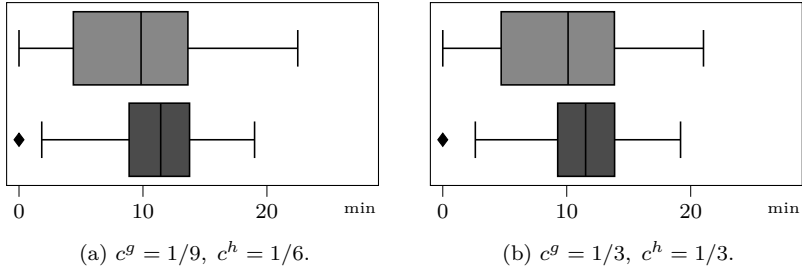


Figure 3.18: Boxplots representing direct waiting time for inpatients (light gray) and outpatients (dark gray).

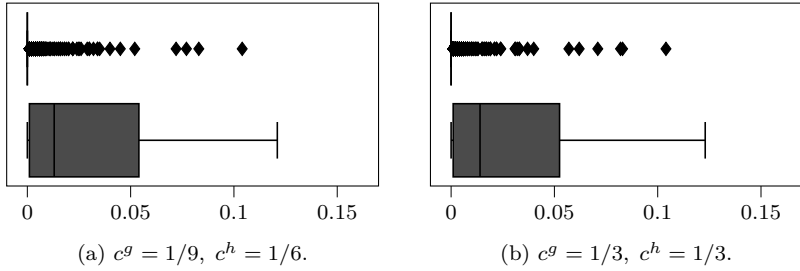


Figure 3.19: Boxplots representing probability of cancellation for inpatients (light gray) and outpatients (dark gray).

eries being canceled. As it can be observed in Figure 3.19, such a probability is significantly higher for outpatient surgeries compared to inpatient surgeries, which is always negligible except for several outliers.

3.7.5 Emergency patients: simulation

In this section we perform an analysis to provide: (i) a comparison between the average waiting time of emergency patients when they are inserted in OR block assigned to different specialties and within a schedule with different levels of robustness, and (ii) a measure of the impact of relaxing the assumption about the insertion of at most one emergency patient per OR block. The second analysis has been made through a straightforward DES model in `Python 3.10`, where emergency patients are generated in accordance with the assumption of the presented model or with a Poisson distribution with the same mean of the Bernoulli distribution described in Section 3.6.1. As well as in the previous operational context, all emergency patients are inserted with the *as soon as possible* policy within the schedule computed under the assumption of having at most one emergency patient.

Results in Tables 3.11 and 3.12 for the two cost preferences show how robust solutions lead to slightly shorter waiting times for emergency surgeries. This is due to a higher idle time, during which the emergency patient could arrive and be inserted without waiting for the end of the current surgery in the assigned OR block. Furthermore, such waiting times slightly increase when it is allowed insertion of more than one emergency patient per OR block, with the insertion of the first emergency patient which causes a longer expected wait for the following ones.

Finally, we observe that the assumption about the insertion of at most one emergency surgery per OR block does not have a large impact on the quality of the solutions. The worsening of the objective function value of the allocation scheduling SMIP model \mathcal{B} (column o.f.) is increased by about 2.5 units, mainly due to over time costs, and it is quite constant among the different parameter configurations, without having a significant impact on the overall performance.

3.8 Conclusions

In this work we presented a two-step stochastic optimization framework for the surgical schedule problem of operating theaters from the perspective of elective patients, modeling and analyzing two particular classes of patients with different characteristics and preferences, that is inpatients and outpatients. Three different uncertainty factors were considered, namely the duration of the surgeries, the arrival of emergency patients, and no-shows. The aim of the new proposed approach is to provide a decision support tool for the operational context in which these two classes of patients shares ORs and to present a quantitative analysis in which research questions emerged in recent literature reviews of OR planning and scheduling.

A CCIP model has been proposed to solve allocation scheduling by setting parameters that determine the robustness of the schedule with respect to cancellations and the OR block patient mix. Starting from a solution obtained in the first phase, a SMIP model allows us to fix the sequencing and the timing of the surgical procedures. At the end, among all the parameter configurations, we choose the one that minimizes an objective function including costs associated with direct and indirect waiting time, cancellations, idle time, and overtime.

Monte Carlo sampling has been adopted to solve the two proposed mathematical programming models with a general-purpose solver. While it resulted to be a methodology capable of providing near-optimal advance scheduling solutions in reasonable times with respect to the application context, the classic SAA proved to be completely inadequate in solving the allocation scheduling SMIP model, due to its high computational complexity. Therefore, two alternative techniques have been proposed: the N -fold SAA as a variant of SAA that is effective on small-size instances and the BRKGA with a custom

Table 3.11: Performance with at most one emergency patient per OR block (Bernoulli) or more (Poisson): average values of instances with $|W| = 1100$, $c^g = 1/9$, and $c^h = 1/6$. Columns “emerg” indicate average waiting time (min) of emergency patients.

ID	Bernoulli						Poisson					
	o.f.	over	idle	canc	wait	emerg	o.f.	over	idle	canc	wait	emerg
A1	20.97	1.09	10.38	8.07	1.43	26.02	23.44	1.13	10.44	10.49	1.38	28.59
A2	21.57	1.08	10.30	8.84	1.36	25.74	23.81	1.11	10.39	11.03	1.33	28.59
A3	21.30	1.09	10.31	8.51	1.39	25.92	23.62	1.12	10.37	10.77	1.36	28.53
B1	28.41	2.18	7.59	16.74	1.98	29.00	30.67	2.21	7.62	18.97	1.87	31.61
B2	29.21	2.19	7.50	17.68	1.85	28.55	31.54	2.21	7.53	19.99	1.81	31.42
B3	28.58	2.20	7.51	16.96	1.91	29.01	31.01	2.24	7.55	19.36	1.87	31.59
C1	29.25	2.29	7.43	17.61	1.92	29.00	31.46	2.31	7.47	19.81	1.87	31.52
C2	30.83	2.36	7.29	19.31	1.87	29.57	33.10	2.39	7.32	21.58	1.82	32.30
C3	29.60	2.34	7.30	18.03	1.93	29.40	32.11	2.39	7.33	20.51	1.89	31.60

Table 3.12: Performance with at most one emergency patient per OR block (Bernoulli) or more (Poisson): average values of instances with $|W| = 1100$, $c^g = 1/3$, and $c^h = 1/3$. Columns “emerg” indicate average waiting time (min) of emergency patients.

ID	Bernoulli						Poisson					
	o.f.	over	idle	canc	wait	energ	o.f.	over	idle	canc	wait	energ
A1	42.61	2.12	31.05	7.89	1.55	26.10	45.15	2.22	31.22	10.21	1.50	28.64
A2	42.91	2.11	30.80	8.53	1.47	25.65	45.47	2.17	30.97	10.90	1.43	28.58
A3	42.82	2.11	30.84	8.33	1.54	25.83	45.33	2.19	31.02	10.61	1.50	28.41
B1	45.65	4.30	22.69	16.70	1.96	29.08	48.05	4.36	22.80	18.97	1.92	31.50
B2	46.23	4.33	22.42	17.55	1.93	28.67	48.69	4.38	22.53	19.89	1.89	31.48
B3	45.67	4.35	22.46	16.89	1.97	28.79	48.31	4.43	22.57	19.39	1.93	31.54
C1	46.26	4.54	22.23	17.51	1.97	28.93	48.54	4.58	22.34	19.70	1.93	31.39
C2	47.59	4.65	21.75	19.27	1.92	29.54	49.93	4.70	21.86	21.50	1.87	32.12
C3	46.38	4.65	21.84	17.92	1.97	29.26	49.09	4.73	21.93	20.50	1.93	31.68

encoding that outperforms the other approaches on medium- and large-size instances.

Several interesting results emerged from the quantitative analysis, based on real data from a Norwegian hospital, enriched with a realistic generation of missing but useful information for the purposes of our study. Although the construction of robust surgical schedules effectively guarantees a reduction in cancellation costs, this entails a worsening of the costs associated with indirect waiting time and idle time. This worsening is such as providing solutions that are inefficient and not preferable to those obtained by setting a low robustness level. Consequently, a real trade-off between direct and indirect waiting time seems to be almost absent, since the scheduling costs have an order of magnitude much higher than the waiting costs in common operational contexts. This suggests that when inpatients and outpatients share ORs, indirect and direct waiting times could be optimized in a hierarchical way, with the former at the upper level and the latter at the bottom level.

The analysis of the three different proxies proposed to promote the composition of different patient mixes within the OR blocks allowed us to evaluate the impact of balancing the patients with the highest cancellation and waiting costs between the OR blocks, or to balance their total expected duration. Once again, direct waiting time turned out to be the least driving criterion, since the best configuration for different instance sets always resulted in the enabling of proxies designed to minimize cancellations and balance idle time and overtime. However, in this case, the orders of magnitude were comparable, and therefore different operational contexts or preferences of the decision-maker could lead to other types of choices. Nonetheless, a general managerial insight concerns the composition of the schedules that have performed better, in which inpatients and outpatients are indifferently scheduled in the same OR blocks, with those of the former type sequenced at the beginning and those of the latter type scheduled at the end, with an effect very similar to a pooling strategy.

Future research developments can follow three different directions. First, the analysis for which the proposed approach was designed can be adapted to real case studies with sufficient information to identify real surgical procedures and their duration. Such information could be exploited by machine learning models for the prediction of the duration of the interventions and of the probability of no-shows of particular classes of patients. New proxies can be identified or the proposed ones can be combined (i.e. through different weights) to define different patient mixes within the OR blocks. Furthermore, the impact of optimizing the OR scheduling week-by-week over time [1, 6] with the proposed stochastic optimization framework deserves to be studied.

From a modeling point of view, a further generalization of the stochastic optimization framework could include the availability of downstream and upstream resources, whose allocation to inpatients and outpatients follows different needs. Furthermore, different overtime allocation policies could be

implemented, as well as the impact of other non-elective patient admission rules (e.g. see [47, 46, 56, 123]) could be investigated. From a methodological perspective, the proposed approach can be extended by simultaneously solving advance scheduling and allocation scheduling by designing metaheuristics able to address the high complexity of the problem.

ON THE EXACTNESS OF JABR-LIKE MODELS AND DISTRIBUTIONALLY ROBUST OPTIMAL POWER FLOW

4.1 Introduction

Carpentier introduced the Optimal Power Flow problem (OPF) as an extension of the problem of optimal economic dispatch of generation in electric power systems [38], with his key contribution being the inclusion of the electric power flow equations in the set of equality constraints. This defining feature remains central to the OPF problem today. A power flow study determines the voltages, currents, and real and reactive power flows in an electric network under given load conditions. The goal of an OPF problem is to minimize generation costs, losses, emissions, and constraint violations. The OPF problem, in its most realistic form, is complex: it is large-scale, non-smooth, non-convex, and nonlinear. It generally has multiple local minima and a corresponding feasibility problem that is strongly NP-hard [23].

The transportation of power from generating plants (generators) to users (loads) occurs by means of a network, called power grid, whose nodes and edges are called buses and lines, respectively. A bus may host at the same time both generators and loads. Electrical network may transport Direct Current (DC) or Alternating Current (AC). In direct current, the electric charge (current) only flows in one direction. Electric charge in alternating current, on the other hand, changes direction periodically. DC is generally used in small electronic devices while the transportation of power on any geographical area is typically based on AC. The complexity of the model makes it challenging to understand its relationship with the characteristics of its inherent graph structure of the network.

In addition, with emerging technologies like renewable generation, batteries, and electric vehicles [40, 126, 102] to tackle climate change, the problem further grows in complexity since OPF problems are needed to adapt to new stochastic variables in energy production and consumption. Future power networks will require more sophisticated methods for managing risks at both transmission and distribution levels. This new class of problems is called stochastic OPF. Various approaches have been developed to tackle this problem. Many formulations assume that uncertain forecast errors follow a prescribed probability distribution [22] however such assumptions can often fail

in weather prediction. In this work, we adopt a different approach, that is to formulate a stochastic OPF as a minimization problem over a Wasserstein ball in the infinite-dimensional space of multivariate probability distributions [61], which uses duality results in optimal transport developed in [99].

Another important aspects of OPF regards the topology of the network. Although strong relaxations are well-known for the radial case [70], the presence of cycles complicates the models and necessitates the inclusion of additional constraints. These constraints are usually formulated by introducing new branches in the network, in order to reduce the degree of the associated polynomial [81, 103]. On the other hand, these new branches add on the number of cycles, and so the number of constraints that must be added to the model. In this work, we discuss two possible methods to handle the cycles of the network directly, without the need of introducing new branches.

This work is divided into two parts, each of which tackles one of the problems discussed above: in Section 4.2 we describe the OPF model, and we derive an accurate model of a distributionally robust OPF. We validate this model with a standard OPF example, modified with the introduction of renewable energy source. In Section 4.3, we focus on a theoretical study of the topology of the network, describing additional constraints on Jabr-like relaxation models and deriving some linearizations based on multilinear relaxation approaches.

4.2 Distributionally robust OPF

For the OPF model construction we define the network as the directed graph (\mathbf{B}, \mathbf{L}) where \mathbf{B} is the set of buses and $\mathbf{L} \subset \mathbf{B} \times \mathbf{B}$ is the set of branches of the network and for each adjacent buses k, m both (k, m) and (m, k) are in \mathbf{L} . So the line l adjacent to k, m is modeled by two edges in the arc $\{(k, m), (m, k)\}$. L can be partitioned in L_0 and L_1 with $|L_0| = |L_1|$ where every line l , adjacent to the buses k, m and with a transformer at k , is oriented so that $(k, m) \in L_0$ and $(m, k) \in L_1$. We also consider a set \mathbf{G} of generators, partitioned into (possibly empty) subsets \mathbf{G}_k for every bus $k \in \mathbf{B}$. In Alternating Current Optimal Power Flow (ACOPF) all voltages and currents are sinusoids with fixed magnitude, frequency, and phase shift. Under these conditions the time domain differential equations governing the voltages and currents of the system can be transformed into a set of complex algebraic equations. The frequency f of a signal is fixed, therefore it is not considered in such transformation. The *phasor* of the sinusoidal function $c \sin(2\pi ft + \gamma)$ is the complex value $ce^{j\gamma}$. We note that the imaginary part of the phasor corresponds to the sinusoidal function at $t = 0$ and therefore the phasor univocally represents the sinusoidal function. We denote the phase current on the line l adjacent to the buses k, m by I_{km} , voltage at bus k by $V_k = |V_k|e^{j\delta_k}$, where $|V_k|$ is the *voltage magnitude* and δ_k is the *voltage angle* or *voltage phase* at bus k . Situations where the magnitudes are zero typically represent exceptional cases like faults or open circuits. In normal system analysis and operation, voltages

are assumed to be nonzero, allowing voltage angles to be well-defined. The power injected in l at k is denoted by $S_{km} = P_{km} + iQ_{km} \in \mathbb{C}$, where the real and imaginary parts P_{km} and Q_{km} are called *real power* and *reactive power* respectively. Considering the cartesian representation of the admittance matrix $\mathbf{Y} = G + iB$, we derive the classical convex Jabr relaxation [70] of the OPF problem:

$$\inf_{\substack{P_g^G, Q_g^G, c_{km}, \\ s_{km}, S_{km}}} \sum_{g \in \mathcal{G}} F_g(P_g^G) \quad (4.1a)$$

$$\text{s.t. } c_{km}^2 + s_{km}^2 = c_{kk}c_{mm}, \quad (4.1b)$$

$$P_{km} + P_{km} \geq 0, \quad (4.1c)$$

$$P_{km} = G_{kk}c_{kk} + G_{km}c_{km} + B_{km}s_{km}, \quad (4.1d)$$

$$Q_{km} = -B_{kk}c_{kk} - B_{km}c_{km} + G_{km}s_{km}, \quad (4.1e)$$

$$S_{km} = P_{km} + jQ_{km}, \quad (4.1f)$$

$$\sum_{km \in L} S_{km} + P_k^L + iQ_k^L = \sum_{g \in \mathcal{G}(k)} P_g^G + i \sum_{g \in \mathcal{G}(k)} Q_g^G, \quad (4.1g)$$

$$P_{km}^2 + S_{km}^2 \leq U_{km}, \quad (4.1h)$$

$$V_k^{\min^2} \leq |c_{kk}| \leq V_k^{\max^2}, \quad (4.1i)$$

$$P_g^{\min} \leq P_g^G \leq P_g^{\max}, \quad (4.1j)$$

$$Q_g^{\min} \leq Q_g^G \leq Q_g^{\max}, \quad (4.1k)$$

$$c_{kk} \geq 0, \quad (4.1l)$$

$$V_k^{\max} V_m^{\max} \geq c_{km} \geq 0, \quad (4.1m)$$

$$-V_k^{\max} V_m^{\max} \leq s_{km} \leq V_k^{\max} V_m^{\max}, \quad (4.1n)$$

$$c_{km} = c_{mk}, s_{km} = -s_{mk}. \quad (4.1o)$$

We denote with *Jabr equality relaxation* the model (4.1). We also define two other models,

$$(4.1) \text{ but with } c_{km}^2 + s_{km}^2 \leq c_{kk}c_{mm} \text{ instead of (4.1b)}, \quad (4.2)$$

$$(4.1) \text{ but with } c_{km}^2 + s_{km}^2 \geq c_{kk}c_{mm} \text{ instead of (4.1b)}, \quad (4.3)$$

and we denote (4.2) as *Jabr convex relaxation* and (4.3) as *Jabr concave relaxation*.

In day-ahead models, taking into account errors in the prediction of loads and of power production of renewable energy sources is crucial. The distribution of the prediction errors can vary greatly depending upon the time of the year. Consequently, when determining the distribution of predictive errors, it is practical to use data that is temporally proximate, often resulting in a limited dataset. Given this constraint, it is crucial for the adopted policy to

showcase robust out-of-sample performances. This is translated into trying to find the policy which minimizes the cost over the worst case of plausible prediction errors. We first formulate a general stochastic OPF problem as a distributionally robust stochastic optimal control problem. We then construct a version of this model starting from a relaxed version of the Jabr OPF model (4.1) and then apply finite reduction results. We conclude this section with the numerical results of our implementation applied to a variant of the IEEE 118-bus test system.

The Distributionally robust (DRO) OPF formulation we consider will be in the following form. Let $x_t \in \mathbb{R}^n$ denote a state vector at time t that includes the internal states of all devices in the network, and let $\xi_t \in \mathbb{R}^{N_\xi}$ denote a random vector in a probability space $(\Omega, \mathcal{F}, \mathbb{P}_t)$ that includes forecast errors of all uncertainties in the network

$$\inf_{\pi \in \Pi} \sup_{\mathbb{P} \in \mathcal{P}} \mathbb{E}^{\mathbb{P}} \sum_{t=0}^T h_t(x_t, \xi_t) \quad (4.4a)$$

$$\text{s.t. } x_t = \pi(x_0, \dots, x_{t-1}, \xi_t, \mathcal{D}_t), \quad (4.4b)$$

$$x_t \in \mathcal{X}_t. \quad (4.4c)$$

The set \mathcal{X}_t denotes the set of feasible state of the network while Π is the set of admissible policies. Some constraints may be modeled deterministically as constraints on set Π and others may be included as risk terms in the objective function. Note that the ambiguity set \mathcal{P} is crucial in this framework. Ideally, the ambiguity set should have a high confidence of containing the true data-generating distribution. At the same time, it should exclude pathological distributions that could lead to overly conservative decisions. Furthermore, it is desirable for the ambiguity set to allow for a tractable reformulation of the problem from a computational perspective, enabling efficient solution approaches. In this work we define the ambiguity set as a ball centered on the inferred from data probability $\hat{\mathbb{P}}$ in the space of probability distributions by using as the probability distance function the Wasserstein metric.

We now aim to build a model satisfying the *Convexity Assumption* [99, Assumption 4.1] in order to apply Kuhn's finite reduction result [99, Theorem 4.2]. Since many of the constraints in the polar OPF problem are not concave, we first consider the Jabr concave relaxation (4.3).

4.2.1 Parameters and decision variables

For each variable in the model, we introduce an additional variable for each time step: $c_{tkm}, s_{tkm}, P_{tk}^G, Q_{tk}^G, S_{tkm}$. We use the term “stochastic variables” to refer to the variables in the model that depend on the prediction errors ξ_t . These variables can be classified into two categories: those that directly depend on the prediction errors and those that depend indirectly through the policy π .

The first category includes variables like the power generation S_k^t at each renewable generator $k \in \mathcal{G}_r$, which are directly influenced by the uncertainty introduced by the prediction errors.

The second category comprises variables related to traditional generators and the values of c and s . These variables are influenced by the policy π , which acts as a correction based on the observed prediction errors. The policy adjusts the planned energy production of the network to account for the uncertainty introduced by the prediction errors.

By incorporating both direct and indirect dependencies on the prediction errors, the stochastic variables provide a comprehensive representation of the system's behavior under uncertainty.

1. *Renewable generators.* For each renewable generator $k \in \mathbb{R}$, the generated power is decomposed as $P_{tk}^G = P_{tk}^{Gn} + P_{tk}^{Ge}(\xi_t)$ and $Q_{tk}^G = Q_{tk}^{Gn} + Q_{tk}^{Ge}(\xi_t)$, where P_{tk}^{Gn}, Q_{tk}^{Gn} represent the predicted power generation at bus t and bus k , and $P_{tk}^{Ge}(\xi_t), Q_{tk}^{Ge}(\xi_t)$ are stochastic variables that correspond to the coordinates of the random vector ξ_t representing the error predictions at time t .
2. *Traditional generators.* Let $P_t^G, Q_t^G \in \mathbb{R}^{c\mathcal{G}}$ be the vectors representing the real and reactive power generation at time t for each traditional generator. We model the time interdependencies through affine policies as follows: $P_{tk}^G = P_{tk}^{Ge}\xi_t + P_t^{Gn}$, where $P_t^{Gn} \in \mathbb{R}^t$ can be interpreted as a nominal schedule and $P_t^{Ge}k \in \mathbb{R}^{N_\xi}$ represents the reserve policy.
3. *c and s variables.* The variables c_{tkm} are defined as follows. $c_{tkm} = c_{tkm}^e\xi_t + c_{tkm}^n$ for all $t = 1, \dots, T$ and $\{k, m\} \in \mathbf{L}$ if $k \neq m$ or $k = m \in \mathbf{B}$. $e_{tkm}^n \in \mathbb{R}$ represents the nominal schedule and $c_{tkm}^e \in \mathbb{R}^{N_\xi}$ represents the adjustment policy. Analogously we define s_{tkm} as $s_{tkm} := s_{tkm}^e\xi_t + s_{tkm}^n$.

The affine policy function is represented by the list:

$$\pi = (P_h^{Gn}, P_h^{Ge}, Q_h^{Gn}, c_{tkm}^e, c_{tkm}^n, s_{tkm}^e, s_{tkm}^n)$$

for all $t = 1, \dots, T$, $h \in \mathcal{G}_t$, and $k, m \in \mathbf{L}$ if $k \neq m$ or $k = m \in \mathbf{B}$.

4.2.2 Constraints definition

All the constraints in (4.3) must hold for each time $t = 1, \dots, T$. Thus, considering time dependency and combining (4.1d)–(4.1f), we have:

$$\begin{aligned} P_{tk}^{Gn} + P_{tk}^{Ge}\xi_t - P_{tk}^d &= \sum_{km \in \delta(k)} P_{tkm} = \\ &= \sum_{km \in \delta(k)} (G_{kk}c_{tkk} + G_{km}c_{tkm} + B_{km}s_{tkm}) = m_k^P \begin{pmatrix} c_t \\ s_t \end{pmatrix}, \end{aligned}$$

where $c_t \in \mathbb{R}^{\mathbf{B} \sqcup \mathbf{L}}$ is the vector with coordinates $(c_t)_k := c_{tkk} \forall k \in \mathbf{B}$ and $(c_t)_{kh} := c_{tkh} \forall kh \in \mathbf{L}$, $s_t \in \mathbb{R}^{\mathbf{L}}$ is the vector with coordinates $(s_t)_{kh} := s_{tkh} \forall kh \in \mathbf{L}$, and $m_k^P \in \mathbb{R}^{\mathbf{B} \times \mathbf{L} \times \mathbf{L}'}$, with $L' := L$, is the vector inducing the second equation with coordinates for all $h \in \mathbf{B}$, $(m_k^P)_h = G_{hh}$, for all $kh \in \delta(k) \subset \mathbf{L}$, $(m_k^P)_{kh} = G_{km}$, and for all $kh \in \delta(k) \subset \mathbf{L}'$, $(m_k^P)_{kh} = B_{kh}$. Since $c_t = c_t^n + c_t^e \xi_t$, where $c_t^e \in \mathbb{R}^{(\mathbf{B} \sqcup \mathbf{L}) \times N_\xi}$ is the matrix with rows $(c_t^e)_h := c_{th}^e \forall h \in \mathbf{B}$ and $(c_t^e)_{hk} = c_{thk}^e \forall kh \in \mathbf{L}$, and $s_t \in \mathbb{R}^{\mathbf{L} \times N_\xi}$ is analogously defined by stacking $s_{thk}^e \forall kh \in \mathbf{L}$. Thus we obtain:

$$P_{tk}^{G_n} + P_{tk}^{G_e} \xi_t - P_{tk}^d - m_k^P \begin{pmatrix} c_t^n \\ s_t^n \end{pmatrix} - m_k^P \begin{pmatrix} c_t^e \\ s_t^e \end{pmatrix} \xi_t = 0.$$

Since this constraint must hold for all possible realizations of ξ_t and therefore for all $\xi_t \in \Xi$, we obtain the following equivalent constraints for all $k \in \mathbf{B}$:

$$P_{tk}^{G_n} - P_{tk}^d - m_k^P \begin{pmatrix} c_t^n \\ s_t^n \end{pmatrix} = 0, \quad (4.5a)$$

$$P_{tk}^{G_e} - m_k^P \begin{pmatrix} c_t^e \\ s_t^e \end{pmatrix} = 0. \quad (4.5b)$$

Constraint (4.5a) means that the nominal values c_t^n must be a solution of (4.3) substituting the variables of renewable power generation with their prediction at time t , while constraint (4.5b) regards the components $P_{tk}^{G_e}$, c_t^e and s_t^e of the policy π . We apply the same procedure to the reactive power in the equation (4.1g) and thus obtain the following analogous constraints for all $k \in \mathbf{B}$:

$$Q_{tk}^{G_n} - Q_{tk}^d - m_k^Q \begin{pmatrix} c_t^n \\ s_t^n \end{pmatrix} = 0, \quad (4.6a)$$

$$Q_{tk}^{G_e} - m_k^Q \begin{pmatrix} c_t^e \\ s_t^e \end{pmatrix} = 0. \quad (4.6b)$$

4.2.3 Inequality affine constraints

In [62], bounds on the wind power forecast errors, denoted as ξ_τ , were not explicitly set. Specifically, the ambiguity set Ξ extends over the entirety of \mathbb{R}^{N_ξ} . However, in this section, we consider the potential advantages of viewing the set Ξ as a compact polytope. Given the inherent limits of wind energy production and the consequent bounds on its forecasting error, it is practical to see Ξ as finite. This perspective provides a distinct advantage: the ambiguity set $\mathcal{B}_\varepsilon(\mathbb{P}_N)$ excludes implausible extreme distributions. For instance, scenarios where forecast error surpasses the maximal potential output of wind generators are disregarded.

Additionally, when the polytope is defined as $\Xi := \{y \in \mathbb{R}^{N_\xi} \mid Hy \leq d\}$, has its symmetry point at $0 \in \mathbb{R}^{N_\xi}$, and when the matrix $H \in \mathbb{R}^{2N_\xi \times N_\xi}$ is full rank, the affine inequality constraints given in (4.3) can be expressed as

constraints on the $\|\cdot\|_1$ -norm of the policy π . To see this, consider an affine inequality constraint in the form:

$$A(\pi)\xi + b(\pi) \leq M, \quad \forall \xi \in \Xi, \quad (4.7)$$

with the condition that $m \leq b(\pi) \leq M$. If we rewrite (4.7) as

$$A(\pi)\xi \leq M - b(\pi), \quad \forall \xi \in \Xi,$$

and let $r := M - b(\pi)$, then (4.7) is satisfied if and only if $A(\pi)\xi \in B_r(0)$ for all ξ in Ξ , given the fact that 0 is a symmetry point of Ξ . If H is a $2N_\xi \times N_\xi$ full-rank matrix, a linear isomorphism $L : \mathbb{R}^{N_\xi} \rightarrow \mathbb{R}^{N_\xi}$ exists such that the $\|\cdot\|_\infty$ -norm ball maps to Ξ . Hence, $A(\pi)\xi$ belongs to $B_r(0) \forall \xi \in \Xi$ if and only if $A(\pi)Lx \in B_r(0) \forall x \in B_r^{\|\cdot\|_\infty}(0)$. This holds true only when the function $A(\pi)L$ from $(\mathbb{R}^{N_\xi}, \|\cdot\|_\infty)$ to \mathbb{R} has a continuity norm $\|\cdot\|_1$ no greater than r . Thus, constraint (4.7) reduces to:

$$\|A(\pi)L\|_1 \leq r = |M - b(\pi)|,$$

translating to a set of finite affine constraints on the policy π . Specifically, we apply this to constraints on voltage and power generation magnitudes. In our model we consider the ambiguity set to be in the form $\Xi = [-E_1, E_1] \times [-E_2, E_2] \times [-E_3, E_3]$. In this case L is the linear isomorphism associated to the 3×3 diagonal matrix having as entries E_1, E_2, E_3 . We consider the constraint on the power generation at the generator $g \in \mathcal{G}$:

$$P_g^{\min} \leq P_g^G = P_g^{Gn} + P_g^{Ge}\xi \leq P_g^{\max} \quad \forall \xi \in \Xi,$$

which is a constraint in the form (4.7) and thus we obtain the following equivalent constraints:

$$\|P_g^{Ge}L\|_1 \leq P_g^{Gn} - P_g^{\min}, \quad (4.8a)$$

$$\|P_g^{Ge}L\|_1 \leq P_g^{\max} - P_g^{Gn}. \quad (4.8b)$$

Similarly, for the voltage magnitude constraints in MW, where $\bar{b} \in \mathbf{B}$ is the reference bus, we have for each bus $b \in \mathbf{B}$:

$$V_b^{\min^2} c_{bb}^- \leq c_{bb} \leq V_b^{\max^2} c_{bb}^+,$$

thus

$$V_b^{\min^2} (c_{bb}^n + c_{bb}^e \xi) \leq c_{bb}^n + c_{bb}^e \xi \leq V_b^{\max^2} (c_{bb}^n + c_{bb}^e \xi) \quad \forall \xi \in \Xi,$$

which is equivalent to:

$$\|(c_{bb}^e - V_b^{\max^2} c_{bb}^e)L\|_1 \leq V_b^{\max^2} c_{bb}^n - c_{bb}^n, \quad (4.9a)$$

$$\|(c_{bb}^e - V_b^{\min^2} c_{bb}^e)L\|_1 \leq c_{bb}^n - V_b^{\max^2} c_{bb}^n. \quad (4.9b)$$

We observe from these equations that policies lean towards conservative values when b (the nominal value) approaches the constraint boundary, ensuring non-violation, while adopting a larger continuity norm otherwise. We note that a similar reasoning can be easily done with the $\|\cdot\|_p$ -norm of the policy, with $p \in [1, +\infty]$, instead of the $\|\cdot\|_1$ -norm.

4.2.4 Conditional Value at Risk

We will consider constraint violations of nonlinear constraints in terms of a loss function $C(\pi, \xi)$. We can treat $C(\pi, \xi)$ as a random variable. If we designate α as an acceptable loss, $\mathbb{P}(C(\pi, \xi) > \alpha)$ represents the probability of violating the acceptable loss α with the policy π . Viceversa we can also consider, given a risk level $\beta \in [0, 1]$, the Value at Risk (VaR) $\alpha_\beta(\pi) = \inf\{\alpha \in \mathbb{R} \mid \mathbb{P}(C(\pi, \xi) > \alpha) \leq \beta\}$ which corresponds to the minimum loss α with a probability smaller than β of being exceeded. The Conditional Value at Risk (CVaR) with risk level β is defined $\phi_\beta(\pi) = \frac{1}{\beta} \int_{C(\pi, \xi) \geq \alpha_\beta(\pi)} f(\pi, \xi) d\mathbb{P}$ and represents the average loss exceeding $\alpha_\beta(\pi)$. By [112, Theorem 1], the CVaR with risk level β can be determined from the formula:

$$\Phi_\beta = \inf_{\tau \in \mathbb{R}} \mathbb{E}^\mathbb{P}[\tau + \frac{1}{\alpha} [f(\pi, \xi) - \tau]^+].$$

4.2.5 Objective Function Definition

The objective function of the DRO OPF model (4.10) comprises a weighted sum of an operational cost function and a constraint violation risk function:

$$h_t = J_{\text{Cost}}^t + \rho J_{\text{Risk}}^t,$$

where $\rho \in \mathbb{R}_+$ is a weight that quantifies the network operator's risk aversion. The operational cost is assumed to be linear or convex quadratic:

$$J_{\text{Cost}}^t = \sum_{g \in \mathcal{G}} c_{1,g} (P_g^G)^2 + c_{2,g} P_g^G + c_{3,g},$$

which captures nominal and reserve costs of responding to wind energy forecast errors.

Affine inequality constraints such as (4.1c) and concave constraints such as (4.1b) cannot be expressed as constraints on the policy π as done for the affine equality constraints of the model. We group each of these constraints in the set \mathcal{C}_t . We model the constraints in \mathcal{C}_t as components of the constraint violation risk function using the CVaR. Each constraint c in \mathcal{C}_t can be written in the form $C_{tc}(\pi, \xi_t) \leq 0$ with C_{tc} affine (concave) with respect to ξ_t if the corresponding constraint c is affine (concave). The CVaR with risk level β of each individual constraint in the set \mathcal{C}_t is given by

$$\inf_{\sigma_{tc}} \frac{1}{\sigma_{tp}} \mathbb{E}_{\xi_t} \{[C_{tc}(\pi, \xi_t) + \sigma_{tc}]_+ - \sigma_{tp} \beta\}.$$

Thus, we obtain the following distributionally robust total CVaR objective:

$$\hat{J}_{\text{Risk}}^t = \inf_{\sigma_{tp}} \sum_{t=0}^T \sum_{c \in \mathcal{C}_t} \sup_{\mathcal{Q}_t \in \hat{\mathcal{P}}_t^{N_s}} \frac{1}{\sigma_{tp}} \mathbb{E}^{\mathcal{Q}_t} [\{[C_{tc}(\pi, \xi_t) + \sigma_{tc}]_+ - \sigma_{tp} \beta\}].$$

We observe that the functions inside the expectations can be expressed as the maximum of affine or concave functions respect to ξ_t :

$$[C_{tc}(\pi, \xi_t) + \sigma_{tc}]_+ - \sigma_{tc} \beta = \max\{C_{tc}(\pi, \xi_t) + (1 - \beta)\sigma_{tc}, -\sigma_{tc}\beta\}$$

for all $c \in \mathcal{C}_t$.

4.2.6 Pre-reduction model

We obtain the following model with optimization variables the components defining the affine function $\pi = (P_h^{G_n}, P_h^{G_e}, Q_h^{G_n}, c_{tkm}^e, c_{tkm}^n, s_{tkm}^e, s_{tkm}^n)$ for all $t = 1, \dots, T$, $h \in \mathcal{G}_t$, and $k, m \in \mathbf{L}$ if $k \neq m$ and for all $k = m \in \mathbf{B}$:

$$\begin{aligned} \inf_{\pi \in \Pi_{\text{aff}}, \sigma_c^t} \sum_{t=0}^T \left\{ \mathbb{E} \left[\hat{J}_{\text{Cost}}^t \right] \right. \\ \left. + \rho \sum_{c \in \mathcal{C}_t} \sup_{\mathbb{Q}_t \in \hat{\mathcal{P}}_t^{N_s}} \mathbb{E}^{\mathbb{Q}_t} \left[\max\{C_{tc}(\pi, \xi_t) \right. \right. \\ \left. \left. + (1 - \beta)\sigma_{tc}, -\sigma_{tc}\beta\} \right] \right\} \end{aligned} \quad (4.10a)$$

s.t. $\forall k \in \mathbf{B}, t = 1, \dots, T$:

$$P_{tk}^{G_n} - P_{tk}^d - m_k^P \begin{pmatrix} c_t^n \\ s_t^n \end{pmatrix} = 0, \quad (4.10b)$$

$$P_{tk}^{G_e} - m_k^P \begin{pmatrix} c_t^e \\ s_t^e \end{pmatrix} = 0, \quad (4.10c)$$

$$Q_{tk}^{G_n} - Q_{tk}^d - m_k^Q \begin{pmatrix} c_t^n \\ s_t^n \end{pmatrix} = 0, \quad (4.10d)$$

$$Q_{tk}^{G_e} - m_k^Q \begin{pmatrix} c_t^e \\ s_t^e \end{pmatrix} = 0, \quad (4.10e)$$

$$\|P_{tg}^{G_e} L\|_\infty \leq P_{tg}^{G_n} - P_g^{\min}, \quad (4.10f)$$

$$\|P_{tg}^{G_e} L\|_\infty \leq P_g^{\max} - P_{tg}^{G_n}, \quad (4.10g)$$

$$\|(c_{tbb}^e - V_b^{\max^2} c_{tbb}^e) L\|_\infty \leq V_b^{\max^2} c_{tbb}^n - c_{tbb}^n, \quad (4.10h)$$

$$\|(c_{tbb}^e - V_b^{\min^2} c_{tbb}^e) L\|_\infty \leq c_{tbb}^n - V_b^{\min^2} c_{tbb}^n, \quad (4.10i)$$

where $\mathbb{E}[\hat{J}_{\text{Cost}}^t]$ represents the mean cost over the empirical distribution \hat{P}^{N_s} .

4.2.7 Finite reduction

Problem (4.10) is an optimization problem over an infinite dimensional vector field $\hat{\mathcal{P}}_t^{N_s}$. To make make the problem tractable, we use results from Kuhn et al. [99] to reformulate

$$\sup_{\mathbb{Q}_t \in \hat{\mathcal{P}}_t^{N_s}} \mathbb{E}^{\mathbb{Q}_t} \left[\rho \max\{C_{tc}(\pi, \xi_t) + (1 - \beta)\sigma_{tc}, -\sigma_{tc}\beta\} \right] \quad (4.11)$$

as a finite dimensional convex program. Whenever $C_{tc}(\pi, \xi_t)$ is affine, we can utilize the reduction result for piecewise affine functions [99, Corollary 5.1], and whenever $C_{tc}(\pi, \xi_t)$ is concave with respect to ξ_t , we will employ [99, Theorem 4.2].

Let us consider the first case. Let $c \in \mathcal{C}_t$ be an affine constraint then we can write $C_{tc}(\pi, \xi_t) = A_{tc}(\pi)\xi_t + b_{tc}(\pi)$ for some $A_{tc}(\pi) \in \mathbb{R}^{N_\xi}$ and $b_{tc}(\pi) \in \mathbb{R}$ both depending linearly on π . Thus applying [99, Corollary 5.1] we obtain that (4.11) is equal to the infimum:

$$\inf_{\lambda_t^c, s_{ti}^c, \gamma_{tik}^c} \lambda_t^c + \frac{1}{N_s} \sum_{i=1}^{N_s} s_{ti}^c, \quad (4.12a)$$

$$\text{s.t. } \rho(b_{tc}(\pi) + (1 - \beta)\sigma_{tc} + \langle A_{tc}, \hat{\xi}_t \rangle + \langle \gamma_{ti1}^c, d - H\hat{\xi}_i \rangle) \leq s_{ti}^c, \quad (4.12b)$$

$$\rho(-\sigma_{tc}\beta + \langle \gamma_{ti2}^c, d - H\hat{\xi}_i \rangle) \leq s_{ti}^c, \quad (4.12c)$$

$$\|H^T \gamma_{ti1}^c - \rho A_{tc}(\pi)\|_\infty \leq \lambda_t^c, \quad (4.12d)$$

$$\|H^T \gamma_{ti1}^c\|_\infty \leq \lambda_t^c, \quad (4.12e)$$

$$\gamma_{tik}^c \geq 0. \quad (4.12f)$$

Let us now consider the case where $C_{tc}(\pi, \xi_t)$, of a constraint $c \in \mathcal{C}_t$, is concave with respect to ξ_t and π , then the maximum in (4.11) is the maximum of a concave and an affine function. Therefore, using [99, Theorem 4.2], problem (4.11) is equal to the following infimum:

$$\inf_{\lambda_t^c, s_{ti}^c, \gamma_{tik}^c} \lambda_t^c + \frac{1}{N_s} \sum_{i=1}^{N_s} s_{ti}^c \quad (4.13a)$$

$$\begin{aligned} \text{s.t. } \rho \Big([-C_{tc}(\pi, \xi_t) + (\beta - 1)\sigma_{tc}]^* (z_{i1}^c - \nu_{ti1}^c) \\ + \sigma_\Xi(\nu_{ti1}) - \langle z_{ti1}^c, \hat{\xi}_i \rangle \Big) \leq s_{ti}^c, \end{aligned} \quad (4.13b)$$

$$\rho \Big([-\delta_{tc}\beta]^* (z_{i2}^c - \nu_{ti2}^c) + \sigma_\Xi(\nu_{ti1}) - \langle z_{ti1}^c, \hat{\xi}_i \rangle \Big) \leq s_{ti}^c, \quad (4.13c)$$

$$\|z_{ik}^c\|_\infty \leq \lambda_t^c. \quad (4.13d)$$

We observe that the infimum corresponds to the optimal value of a convex problem since $[-\ell_{tk}]^*$ is convex. Finally, by substituting (4.12) and (4.13) into (4.10), we obtain the equivalent finite dimensional convex reformulation of Problem (4.10)

$$\inf_{\substack{\pi \in \Pi_{\text{aff}}, \\ \sigma_t^c, \lambda_t^c, \\ s_{ti}^c, \gamma_{tik}^c}} \sum_{t=0}^T \left\{ \mathbb{E} \left[\hat{J}_{\text{Cost}}^t \right] + \rho \sum_{c \in \mathcal{C}_t} \lambda_t^c + \frac{1}{N_s} \sum_{i=1}^{N_s} s_{ti}^c \right\} \quad (4.14a)$$

$$\text{s.t. } \forall k \in \mathbf{B}, t = 1, \dots, T :$$

$$P_{tk}^{G_n} - P_{tk}^d - m_k^P \begin{pmatrix} c_t^n \\ s_t^n \end{pmatrix} = 0, \quad (4.14b)$$

$$P_{tk}^{G_e} - m_k^P \begin{pmatrix} c_t^e \\ s_t^e \end{pmatrix} = 0, \quad (4.14c)$$

$$Q_{tk}^{G_n} - Q_{tk}^d - m_k^Q \begin{pmatrix} c_t^n \\ s_t^n \end{pmatrix} = 0, \quad (4.14d)$$

$$Q_{tk}^{G_e} - m_k^Q \begin{pmatrix} c_t^e \\ s_t^e \end{pmatrix} = 0, \quad (4.14e)$$

$$\|P_{tg}^{G_e} L\|_\infty \leq P_{tg}^{G_n} - P_g^{\min}, \quad (4.14f)$$

$$\|P_{tg}^{G_e} L\|_\infty \leq P_g^{\max} - P_{tg}^{G_n}, \quad (4.14g)$$

$$\|(c_{tbb}^e - V_b^{\max^2} c_{tbb}^e) L\|_\infty \leq V_b^{\max^2} c_{tbb}^n - c_{tbb}^n, \quad (4.14h)$$

$$\|(c_{tbb}^e - V_b^{\min^2} c_{tbb}^e) L\|_\infty \leq c_{tbb}^n - V_b^{\max^2} c_{tbb}^n, \quad (4.14i)$$

$\forall c \in \mathcal{C}_t$ concave constraints:

$$\begin{aligned} \text{s.t. } \rho \Big([-C_{tc}(\pi, \xi_t) + (\beta - 1)\sigma_{tc}]^* (z_{i1}^c - \nu_{ti1}^c) \\ + \sigma_\Xi(\nu_{ti1}) - \langle z_{ti1}^c, \hat{\xi}_i \rangle \Big) \leq s_{ti}^c, \end{aligned} \quad (4.14j)$$

$$\rho \Big([-\delta_{tc}\beta]^* (z_{i2}^c - \nu_{ti2}^c) + \sigma_\Xi(\nu_{ti1}) - \langle z_{ti1}^c, \hat{\xi}_i \rangle \Big) \leq s_{ti}^c, \quad (4.14k)$$

$$\|z_{ik}^c\|_\infty \leq \lambda_t^c, \quad (4.14l)$$

$\forall c \in \mathcal{C}_t$ affine constraints:

$$\rho(b_{tc}(\pi) + (1 - \beta)\sigma_{tc} + \langle A_{tc}, \hat{\xi}_t \rangle + \langle \gamma_{ti1}^c, d - H\hat{\xi}_i \rangle) \leq s_{ti}^c, \quad (4.14m)$$

$$\rho(-\sigma_{tc}\beta + \langle \gamma_{ti2}^c, d - H\hat{\xi}_i \rangle) \leq s_{ti}^c, \quad (4.14n)$$

$$\|H^T \gamma_{ti1} - \rho A_{tc}(\pi)\|_\infty \leq \lambda_t^c, \quad (4.14o)$$

$$\|H^T \gamma_{ti1}\|_\infty \leq \lambda_t^c, \quad (4.14p)$$

$$\gamma_{tik} \geq 0. \quad (4.14q)$$

Example 4. As a concrete example, we illustrate the modeling of the power loss inequality constraint (4.1c) within our DRO OPF framework. This involves expanding the expressions of P_{tkh} and P_{thk} as follows

$$\begin{aligned} 0 &\geq -P_{tkh} - P_{thk} = \\ &= -G_{kk}c_{kk} - G_{kh}c_{tkh} - B_{kh}s_{tkh} - G_{hh}c_{thh} - G_{hk}c_{thk} - B_{hk}s_{thk} = \\ &= -(G_{hk} + G_{kh})c_{tkh} - (B_{kh} - B_{hk})s_{tkh} - G_{kk}c_{tkk} - G_{hh}c_{thh} \\ &= -(G_{hk} + G_{kh})(c_{tkh}^n + c_{tkh}^e \xi_t) - (B_{kh} - B_{hk})(s_{tkh}^n + s_{tkh}^e \xi_t) + \\ &\quad - G_{kk}(c_{tkk}^n + c_{tkk}^e \xi_t) - G_{hh}(c_{thh}^n + c_{thh}^e \xi_t) = C_{tc}(\pi, \xi_t). \end{aligned}$$

Thus the power loss inequality constraint can be written as $C_{tc}(\pi, \xi_t) =$

$A_{tc}(\pi)\xi_t + b_{tc}(\pi) \leq 0$ with $A_{tc}(\pi)$ and $b_{tc}(\pi)$ defined as follows:

$$\begin{aligned} A_{tc}(\pi) &= -[(G_{hk} + G_{kh})c_{tkh}^e + (B_{kh} - B_{hk})s_{tkh}^e - G_{kk}c_{tkk}^e - G_{hh}c_{thh}^e], \\ b_{tc}(\pi) &= -[(G_{hk} + G_{kh})c_{tkh}^n + (B_{kh} - B_{hk})s_{tkh}^n - G_{kk}c_{tkk}^n - G_{hh}c_{thh}^n]. \end{aligned}$$

Therefore, by following the passages at the beginning of this section, the CVaR of violating the power loss constraint, can be modeled in the form (4.12).

4.2.8 Numerical results

In this subsection, we apply the model developed above to a variant of the IEEE 118-bus test system, as referenced in [62] and [150]. This modified version integrates three wind farms connected to buses #1, #9, and #26. Consequently, the conventional generators originally connected to bus #1 and bus #26 have been removed. The implementation details discussed in this chapter, along with the codes for models (4.1) and (4.3), are available for download at [111].

The wind power forecast errors were obtained from a data source provided by CESI. The process involves fitting a Weibull distribution to represent the probability distribution of wind power production [29, 139]. The predicted power output is then set as the mean of this distribution. Error samples are generated by drawing random samples from the fitted distribution and then computing the discrepancy with the predicted power generation.

To align our findings with existing literature [62], we adjusted our forecasting errors to have a zero mean. Additionally, standard deviations were set at 200MW for wind farms #1 and #9, and 300MW for wind farm #26. We focused on the single-period stochastic optimization problem following the approach in [62].

The dataset comprises 152 observations and is divided into 13 samples to train the model and 139 test samples for measuring the out-of-sample performance of the solution. Therefore $N_s = 13$ and $N_{\Xi} = 3$ since there are three windfarms whose power generation is predicted. The optimization task was completed in 100 seconds for each loop on a laptop equipped with an Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz and 8GB of RAM.

Figure 4.1 illustrates the variation in the average operational cost based on different values of the risk aversion parameter ρ and the radius ε of the Wasserstein metric. An evident trend is the rise in average nominal cost as ρ increases. This indicates a greater focus on avoiding constraint violations across all potential realizations of ξ . While this strategy ensures higher security, it does come at the cost of increased operational expenses.

Figure 4.2 showcases the nominal power loss in some key lines of the network. In general, there is a decreasing constraint violation as ρ increases, as in line 86. Line 15 is an exception. This happens because a overall lower cost is achieved by allowing higher risk of violating the power loss constraint in this line.

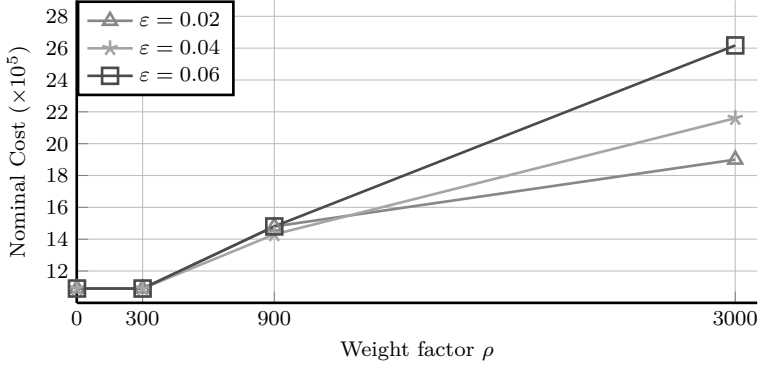


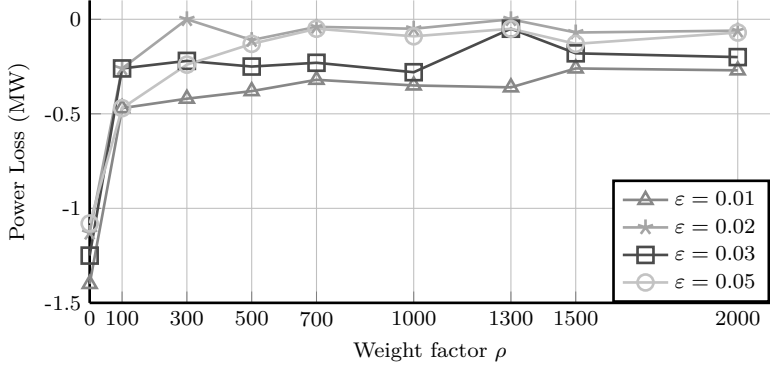
Figure 4.1: Comparison of the average operational cost for various values of risk aversion ρ and Wasserstein metric radius ϵ .

In Figure 4.3, we observe how out of sample performance varies with ϵ , by showing the average number of power loss constraint violations of the test samples. We observe that large ϵ ensures fewer constraints violations. Inequality affine constraints on power generation and voltage magnitudes, implemented as policy constraints, are never violated for the tests samples.

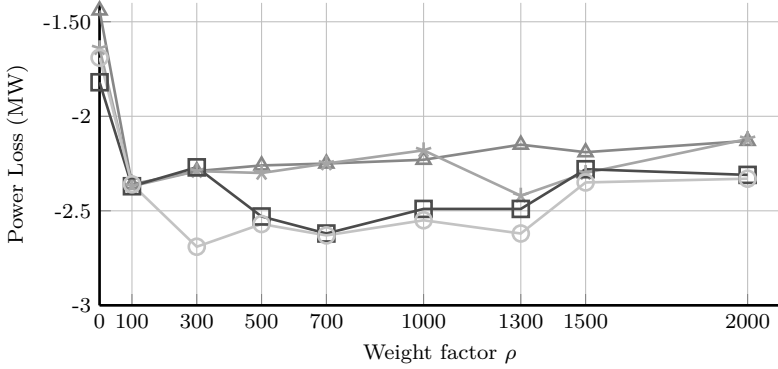
Lastly, Figure 4.4 shows the relationship between nominal real power output and the risk aversion coefficient ρ . A discernible pattern is the increasing conservatism in policy as ρ elevates. This ensures respect for magnitude constraints, especially when confronted with extreme realizations of the predictions error ξ . However the change in power generation in some buses, as in bus 15, is not monotonic, due to the nonlinearities of the OPF model, which make the behavior of the solution with respect to the changes of various parameters somewhat unpredictable.

4.2.9 Conclusions

We have showcased the efficacy and adaptability of our proposed data-driven, distributionally robust stochastic OPF methodology. Unlike many existing models, the one we propose directly integrates error training datasets, eliminating the need to presuppose a forecast error distribution. This approach allows us to use distributionally robust optimization techniques, leading to enhanced out-of-sample performance. Importantly, our underlying deterministic model operates with fewer assumptions compared to the conventional DC OPF. This ensures resilience, especially when faced with extreme weather events or fluctuating demand scenarios in the network. We introduced a novel method of modeling affine inequality constraints as continuity constraints on the policy, allowing us to impose fewer constraints using Conditional Value at Risk (CVaR). The key distinction between these two approaches is that



(a) Line 86.



(b) Line 15.

Figure 4.2: Comparison of the power loss in six lines of the network for various values of risk aversion parameter ρ and the Wasserstein metric radius ϵ .

the former is hard, enforcing the constraint for every possible realization of error predictions but is only applicable to affine constraints, while the latter can be applied more generally to concave constraints but it does not guarantee compliance for every possible realization of error predictions. However, it is worth noting that our model tends to be more computationally demanding than the DCOPF-based stochastic optimal power flow model presented in [62], due to the increased number of variables and constraints derived from the AC OPF formulation. To reduce the number of variables and constraints, one approach is to identify the critical lines in the network, by running a relaxation of the model without the line constraints and by identifying where these constraints are violated, and then apply the CVaR constraints exclusively to these lines. From simulations results, we observed that larger ϵ may

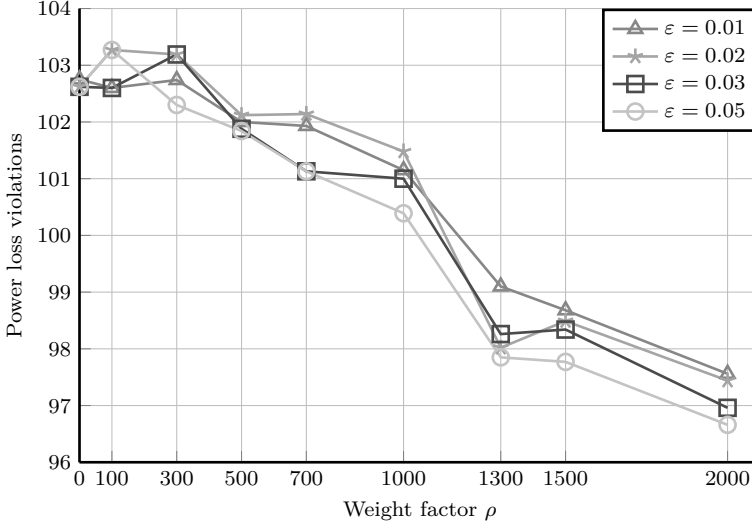
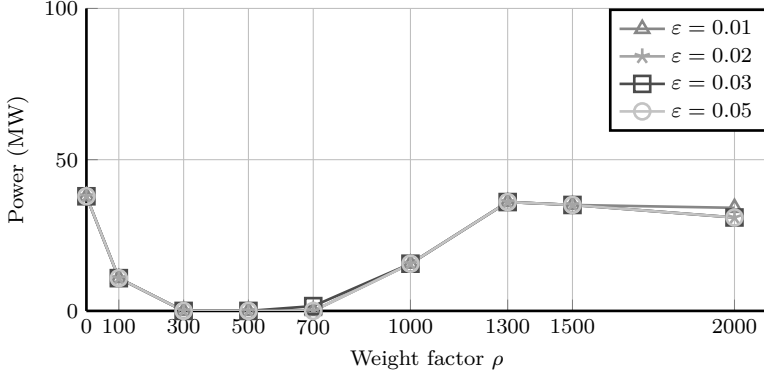


Figure 4.3: Comparison of the number of out of samples constraints violation for various values of risk aversion parameter ρ and the Wasserstein metric radius ϵ .

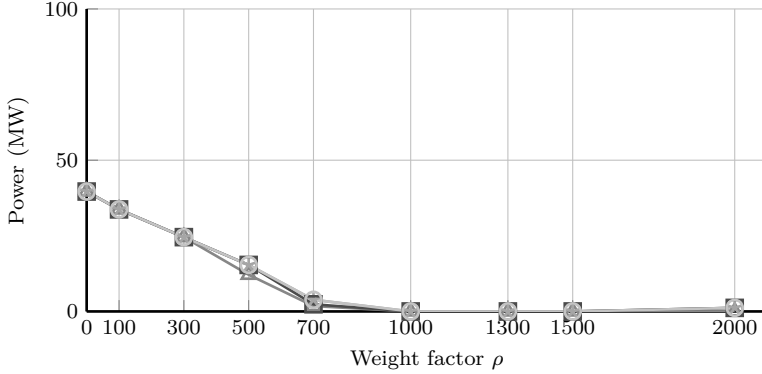
lead to large constraint violation when the sum of the CVaRs has a lower overall risk by allowing higher risk of violation of specific constraints. This may be addressed by considering the risk of joint constraint violation instead of modeling each constraint individually. In our model we considered Ξ to be in the form $[-E_1, E_1] \times [-E_2, E_2] \times [-E_3, E_3]$. Although this would work well if the components of ξ were independent, this is rarely the case for wind power prediction, and this results in overly conservative policies. Future works can include considering the best polytope fitting the considered samples. Moreover, limiting our consideration to affine policies in a nonlinear problem like the OPF overlooks numerous admissible policies that are nonlinear. A potential avenue for future research could involve exploring piecewise affine policies instead.

Acknowledgment

We would like to thank CESI – Centro Elettrotecnico Sperimentale Italiano for providing the data used in this study and for their valuable suggestions during the research process.



(a) Bus 15.



(b) Bus 50.

Figure 4.4: Comparison of output real powers of selected generators for different risk levels ρ and Wasserstein metric values ε . Note that 100MW is the upper bound for the power generation of both buses.

4.3 A multilinear approach on cycles

We now discuss the relaxation (4.1) and a possible improvement. A famous result states that if (\mathbf{B}, \mathbf{L}) is a multisource radial network, then the Jabr equality ACOPF relaxation is exact [70]. If the network is not radial, that is, it contains cycles, we can recover exactness thanks to the following observation.

Observation 9. Model (4.1) with the additional *cycle constraint* [81]

$$\sum_{k=0}^{\lfloor n/2 \rfloor} \sum_{\substack{A \subset [n] \\ |A|=2k}} (-1)^k \prod_{h \in A} s_{k_h, k_{h+1}} \prod_{h \in A^c} c_{k_h, k_{h+1}} = \prod_{k=1}^n c_{k_i, k_i}. \quad (4.15)$$

for every cycle A in a cycle basis of (\mathbf{B}, \mathbf{L}) is exact.

In [81], auxiliary branches were added to the network, dividing each cycle in smaller cycles, to decrease the degree of the polynomials defining the cycle constraint. McCormick linearization is then applied. The problem with this approach is that one auxiliary branch is added for every branch in the cycle. We now discuss how to deal with (4.15) directly.

4.3.1 Standard multilinear relaxation

We now aim at studying constraint (4.15) from a multilinear optimization point of view. Consider a set of multilinear constraints:

$$\sum_{I \in \mathcal{I}_j} c_I^j \prod_{v \in I} x_v \leq b \quad \forall j \in \{1, 2, \dots, m\}, \quad (4.16a)$$

$$x_v \in [l_v, u_v] \quad \forall v \in V, \quad (4.16b)$$

where V denotes the set of the indices of the variables and \mathcal{I}_j is a set of subsets of V . A straightforward simplification is to introduce a variable z_I for every subset I of variables appearing in the constraints. Thus we obtain the following equivalent problem

$$\sum_{I \in \mathcal{I}_j} c_I^j z_I \leq b_j \quad \forall j \in \{1, 2, \dots, m\}, \quad (4.17a)$$

$$z_I = \prod_{i \in I} x_i \quad \forall I \in \mathcal{I} := \cup_{j=1}^m \mathcal{I}_j, \quad (4.17b)$$

$$x_v \in [l_v, u_v] \quad \forall v \in V. \quad (4.17c)$$

By affine transformations, we can assume the variables x_v to be in the interval $[0, 1]$. Note that such affine transformations need to be handled with care, we will cover this in Section 4.3.2. Since constraint (4.17a) is now linear, we are interested in the linearization of the following set $Pr := \{(x, z) \in [0, 1]^V \times [0, 1]^{\mathcal{I}} \mid z_I = \prod_{i \in I} x_i \forall I \in \mathcal{I}\}$. We now define the standard linear relaxation of Pr .

Definition 5 (Standard form relaxation). Let the polyhedral PrR be defined by the following linear constraints:

$$z_I \leq x_v, \quad \forall v \in I, \forall I \in \mathcal{I}, \quad (4.18a)$$

$$z_I + \sum_{v \in I} (1 - x_v) \geq 1, \quad \forall I \in \mathcal{I}, \quad (4.18b)$$

$$z_I \geq 0, \quad \forall I \in \mathcal{I}, \quad (4.18c)$$

$$x_v \in [0, 1], \quad \forall v \in V. \quad (4.18d)$$

Note that this relaxation can be strengthened, for example with recursive McCormick linearization or flower inequalities [118].

4.3.2 Handling affine transformations

We assumed $x_v \in [0, 1]$ because affine transformation of homogeneous constraints remain homogeneous. But it must be noted that for each non linear affine transformation, that is when the lower bound of the corresponding variable is not zero, the number of binomials increases. More precisely, given a monomial defined by $I \in \mathcal{I}$, let $I' \subset I$ be the subset of variables in I for which a nonlinear transformation is applied. Then the monomial I is split into $2^{|I'|+1}$ new monomials. When the size of such I' is large this greatly increases the number of auxiliary variables z_J which must be introduced.

Thus applying many non linear affine transformation can be very costly and complicates the handling of the constraints. For this reason, instead of applying non linear affine transformation, for each variables $x_v \in V$ such that $x_v \in [l_v, u_v]$ and $l_v * u_v < 0$, we split the problem in two new subproblems having $x_v \in [l_v, 0]$ and $x_v \in [0, u_v]$ respectively. This way, linear transformations can be applied in the subproblems. This creates many subproblems, and so we rewrite the subproblems as a unique mixed integer programming problem.

Let $C = \{k_1, \dots, k_n\} \subset \mathbf{B}$ be a cycle. The variables s_h are in the form $s_h \in [-u_{s_h}, u_{s_h}]$ where $h = (k_i, k_{i+1})$ for all $i = 1, \dots, n$. We can then substitute s_h with $u_h s'_h = s_h$ where $s'_h \in [-1, 1]$. We then define the sign variables $\sigma_h \in \{0, 1\}$ for each $h \in C$, where $\sigma_h = 0$ if s_h is negative and 1 if it is positive. We can now rewrite the cycle constraint as:

$$\sum_{k=0}^{\lfloor n/2 \rfloor} \sum_{\substack{A \subset [n] \\ |A|=2k}} (-1)^k \left(\prod_{h \in A} (2\sigma_h - 1) u_h \right) z_A = z'_C$$

Where we substitute the monomial $\prod_{h \in A} s_{k_h k_{h+1}}$ with z_A and the product $\prod_{k=1}^n c_{k_i, k_i}$ with z'_C . For each even subset $A \subset [n]$ we introduce the binary variable $\lambda_A \in \{0, 1\}$ which is 0 if $\prod_{h \in A} (2\sigma_h - 1)$ is -1 and $\lambda_A = 1$ otherwise. The cycle constraint becomes:

$$\sum_{k=0}^{\lfloor n/2 \rfloor} \sum_{\substack{A \subset [n] \\ |A|=2k}} (-1)^k (2\lambda_A - 1) U_A z_A = z'_C$$

The product $\lambda_A z_A$ can easily be linearized. To enforce the relation $2\lambda_A - 1 = \prod_{h \in A} (2\delta_h - 1)$, simply note that $\lambda_A = 0$ if and only if there is an odd number

of δ_h equal to 0, that is, there exists $m_A \in \mathbb{Z}$ such that:

$$\lambda_A + 2m_A = \sum_{h \in A} \delta_h.$$

4.3.3 Other cuts

We now discuss a different approach of handling multilinear terms. Let I be a set of indices and $x_v, v \in I$, continuous variables such that $x_v \in [l_v, u_v] \subset [-1, 1]$. Let also the set I be partitioned in two set such that $I = J \oplus K$ and

$$l_v > 0, \quad u_v = 1, \quad \forall v \in J, \quad (4.19a)$$

$$l_v = -u_v, \quad u_v < 1, \quad \forall v \in K. \quad (4.19b)$$

In practice, $x_v, v \in J$ represent cosine variables and $x_v, v \in K$ represent sine variables. If we define $z_I := \prod_{v \in I} x_v$ and $I' := I \setminus \{v\}$, then the following lower and upper bounds trivially hold:

$$z_I \leq x_v \prod_{v' \in I'} u_{v'}, \quad \forall v \in J, \quad (4.20a)$$

$$z_I \leq |x_v| \prod_{v' \in I'} u_{v'}, \quad \forall v \in K, \quad (4.20b)$$

$$z_I \geq -|x_v| \prod_{v' \in I'} u_{v'}, \quad \forall v \in K, \quad (4.20c)$$

$$z_I \geq x_v \prod_{v' \in I'} l_{v'}, \quad \forall v \in J, I' = J' \oplus K' : K' = \emptyset, \quad (4.20d)$$

$$z_I \geq -x_v \prod_{v' \in I'} u_{v'}, \quad \forall v \in J, I' = J' \oplus K' : K' \neq \emptyset. \quad (4.20e)$$

We now discuss a non-trivial bound, that generalizes inequality (4.18b).

Lemma 12. *The following inequality holds:*

$$z_I + \sum_{v \in I} c_v(u_v - x_v) \geq \prod_{v \in I} u_v, \quad c_v := \prod_{v' \in I \setminus \{v\}} u_{v'}. \quad (4.21)$$

Proof. Because we are dealing with a multilinear inequality, it is sufficient to verify that it holds for every vertex of the multidimensional rectangular cuboid

$$\mathfrak{C} := \prod_{v \in I} [l_v, u_v] \subset [-1, 1]^{|I|}.$$

For such a vertex x , we have either $x_v = l_v$ or $x_v = u_v$ for every $v \in I$. Define $I_1 := \{v \in I \mid x_v = l_v\}$ and $I_2 := \{v \in I \mid x_v = u_v\}$, and J_1, J_2, K_1, K_2 analogously. By defining $k_1 := |K_1|$, we then have

$$z_I + \sum_{v \in I} c_v(u_v - x_v) = \prod_{v \in I_1} l_v \prod_{v \in I_2} u_v + \sum_{v \in I_1} c_v(u_v - x_v) + \sum_{v \in I_2} c_v(u_v - x_v) =$$

$$\begin{aligned}
&= \prod_{v \in I_1} l_v \prod_{v \in I_2} u_v + \sum_{v \in I_1} c_v(u_v - l_v) = \\
&= (-1)^{k_1} \prod_{v \in K_1 \cup I_2} u_v \prod_{v \in J_1} l_v + \sum_{v \in J_1} c_v(1 - l_v) + 2 \sum_{v \in K_1} c_v u_v = \\
&= A + B + C.
\end{aligned}$$

Now, we rewrite each of the three terms. First,

$$\begin{aligned}
A &= (-1)^{k_1} \prod_{v \in K_1 \cup I_2} u_v \prod_{v \in J_1} l_v = (-1)^{k_1} \prod_{v \in K_1 \cup I_2} u_v \prod_{v \in J_1} u_v \prod_{v \in J_1} l_v = \\
&= (-1)^{k_1} \prod_{v \in I} u_v \prod_{v \in J_1} l_v,
\end{aligned}$$

where the second equality holds because $u_v = 1, \forall v \in J_1$. Second,

$$\begin{aligned}
B &= \sum_{v \in J_1} c_v(1 - l_v) = \sum_{v \in J_1} \left(\prod_{v' \in I \setminus \{v\}} u_{v'} \right) (1 - l_v) = \\
&= \sum_{v \in J_1} \left(\prod_{v' \in I} u_{v'} \right) (1 - l_v) = \\
&= \left(\prod_{v \in I} u_v \right) \sum_{v \in J_1} (1 - l_v),
\end{aligned}$$

where the third equality holds because $u_v = 1 \forall v \in J_1$. Finally,

$$\begin{aligned}
C &= 2 \sum_{v \in K_1} c_v u_v = 2 \sum_{v \in K_1} \left(\prod_{v' \in I \setminus \{v\}} u_{v'} \right) u_v = 2 \sum_{v \in K_1} \left(\prod_{v' \in I} u_{v'} \right) = \\
&= 2k_1 \left(\prod_{v \in I} u_v \right).
\end{aligned}$$

We then have

$$\begin{aligned}
z_I + \sum_{v \in I} c_v(u_v - x_v) &= A + B + C = \\
&= \left(\prod_{v \in I} u_v \right) ((-1)^{k_1} \prod_{v \in J_1} l_v + \sum_{v \in J_1} (1 - l_v) + 2k_1).
\end{aligned}$$

To conclude, we just need to prove that

$$(-1)^{k_1} \prod_{v \in J_1} l_v + \sum_{v \in J_1} (1 - l_v) + 2k_1 \geq 1,$$

but this is true because

$$(-1)^{k_1} \prod_{v \in J_1} l_v + \sum_{v \in J_1} (1 - l_v) + 2k_1 =$$

$$\begin{aligned}
&= ((-1)^{k_1} - 1) \prod_{v \in J_1} l_v + 2k_1 + \prod_{v \in J_1} l_v + \sum_{v \in J_1} (1 - l_v) = \\
&\geq ((-1)^{k_1} - 1) \prod_{v \in J_1} l_v + 2k_1 + 1 \geq (-1)^{k_1} + 2k_1 \geq 1,
\end{aligned}$$

where the first inequality holds because $\prod_{v \in J_1} l_v + \sum_{v \in J_1} (1 - l_v) \geq 1$ because of the standard multilinear relaxation (4.18b). The second inequality holds because $0 < l_v < 1$ and so $0 \leq \prod_{v \in J_1} l_v < 1$, and the last inequality holds because $k_1 \geq 0$ integer. \square

Note that inequality (4.21) is tight in $v := (u_v)_{v \in I}$, in every vertex v' adjacent to v , and in every point of the segment connecting v and v' .

4.3.4 A possible generalization

We now consider when, given a vertex a of the cuboid \mathfrak{C} , we can define a separating hyperplane π for Pr , that is, when $\pi : \mathbb{R}^n \rightarrow \mathbb{R}$ such that either $\pi(x) \leq \prod_{v \in I} x_v$ for all x in \mathfrak{C} for all x in the vertex or $\pi(x) \geq \prod_{v \in I} x_v$ for all x in \mathfrak{C} . Furthermore, we want a good cut, so we look for hyperplanes such that $\pi(a) = \prod_{v \in I} a_v$ and $\pi(y) = \prod_{v \in I} y_v$ for all vertices y in \mathfrak{C} , adjacent to a . First we observe that since in \mathbb{R}^{n+1} fixed $n+1$ affinely independent points there exists a unique hyperplane containing all these points, then there exists a unique hyperplane such that $\pi(y) = \prod_{v \in I} y_v$ for all vertices y in \mathfrak{C} , adjacent to a and for $y = a$.

Observation 10. The hyperplane $\pi(x) := \prod_{v \in I} a_v + \sum_{v \in I} c_v a_v (x_v - a_v)$, where $c_v := \prod_{v' \in I \setminus \{v\}} a_{v'}$, is the only hyperplane such that $\pi(y) = \prod_{v \in I} y_v$ for all vertices y in \mathfrak{C} , adjacent to a , and such that $\pi(a) = \prod_{v \in I} a_v$.

We now look for conditions for when this uniquely defined hyperplane is also a separating hyperplane for Pr . Note in particular that (4.21) is of the type described above with $a = v$. Let us define two functions

$$\begin{aligned}
f_a(x) &= \prod_{v \in I} x_v + \sum_{v \in I} c_v (a_v - x_v) - \prod_{v \in I} a_v, \\
g_a(x) &= \prod_{v \in I} x_v - \sum_{v \in I} c_v x_v.
\end{aligned}$$

We are looking for $a \in \mathfrak{C}$ such that either

$$\begin{aligned}
f_a(x) &\leq 0 & \forall x \in \mathfrak{C}, \text{ or} \\
f_a(x) &\geq 0 & \forall x \in \mathfrak{C},
\end{aligned}$$

that is

$$\begin{aligned}
g_a(x) &\leq g_a(a), & \forall x \in \mathfrak{C}, \text{ or} \\
g_a(x) &\geq g_a(a), & \forall x \in \mathfrak{C}.
\end{aligned}$$

So we are looking for a such that the function g_a attains its maximum or minimum at a . Note that $g_a(a) = (1 - |I|) \prod_{v \in I} a_v$. We now make a simplification on hypothesis (4.19). Let I be partitioned in two set such that $I = J \oplus K$ and let $0 < l < u < 1$ such that

$$l_v = l > 0, \quad u_v = 1, \quad \forall v \in J, \quad (4.25a)$$

$$l_v = -u_v = -u, \quad u_v = u < 1, \quad \forall v \in K. \quad (4.25b)$$

Lemma 13. *Let a be a vertex of \mathfrak{C} such that, for all $v \in I$, either $a_v = 1$ or $a_v = u$ or $a_v = -u$ and let $p := |\{v \in I : a_v = -u\}|$. Then we have that*

$$g_a(x) \leq g_a(a), \quad \forall x \in \mathfrak{C}, \text{ if } p \text{ odd}, \quad (4.26)$$

$$g_a(x) \geq g_a(a), \quad \forall x \in \mathfrak{C}, \text{ if } p \text{ even}. \quad (4.27)$$

Proof. If we derive the function g_a we obtain

$$\frac{\partial g_a(x)}{\partial x_w} = \prod_{v \in I \setminus \{w\}} x_v - \prod_{v \in I \setminus \{w\}} a_v.$$

Let p be odd. If w is such that $a_w = -u$, then $\frac{\partial g_a(x)}{\partial x_w} \leq 0$ and so the maximum is obtained by choosing $x_w = -u = a_w$, if w is such that $a_w \neq -u$, that is $a_w = 1$ or $a_w = u$, then $\frac{\partial g_a(x)}{\partial x_w} \geq 0$ and so the maximum is obtained by choosing $x_w = 1$ or $x_w = u$, that is $x_w = a_w$. By doing the same reasoning for every partial derivative, we obtain that a is in fact the maximum of g_a . The same reasoning can be done for when p is even. \square

4.3.5 Conclusions and future work

In this section, we studied different techniques of handling multilinear terms, starting from the standard relaxation and developing tailored inequalities that can be added to the Jabr model in order to obtain higher lower bounds and better solutions, that can in principle make Newton–Raphson method converge faster if used as starting points. In future works, we plan to generalize some of the theoretical results we presented and to test them numerically in comparison with other techniques used in the literature.

CONCLUSIONS

In this thesis, the author has studied various combinatorial optimization methods and, through tailored approaches, has applied them to different combinatorial problems. These problems, while interesting in themselves and leading to noteworthy results, primarily serve as applications of the studied methods. In particular, it is worth noting how these methods and approaches can lead to interesting generalizations whose applications could be highly beneficial for solving numerous combinatorial problems.

In Chapter 1, the approach of the integrality gap studied for the traveling salesman problem is generalized for the Steiner tree problem, stressing the generalizability of this methodology for the study of the integrality gap. Moreover, the simplification techniques applied to the bidirected cut formulation not only provided theoretical insights on the properties of optimal vertices but also reduced the number of suboptimal ones, suggesting that the study of combinatorial polytopes could greatly benefit from these types of techniques. In addition, similar approaches could find a use in the direct solution of combinatorial problems.

In Chapter 2, the exploitation of the multi-objective approach to training a neural network suggests the advantages of training a network that is both robust and lightweight. In addition, several generalizations can be derived from the ensemble approach, stressing its broad applicability to different multiclassification tasks. Finally, the study of the weight distribution, underlining the polarized behavior of their values, could have an impact on the study of the weights of different networks, depending both on their architecture and the nature of the training data.

In Chapter 3, different stochastic optimization techniques are applied to a two-stage problem studying the operating theater where patients with different characteristics are present. In particular, various sample average approximation approaches are applied, and the multi-objective approach used in the first stage shows the importance of parametrizing different characteristics one looks for in a solution to obtain better solutions in the second stage of the optimization. This approach has the potential of being generalized for various two-staged stochastic optimization problems.

In Chapter 4, a distributionally robust optimization technique is applied to an optimal power flow problem, showing good out-of-sample performances. This method is very general, and different stochastic problems could benefit from its application, given the flexible yet well-parameterized definition of the ambiguity set in which the adversarial distribution can lie. In addition, the multilinear study conducted on a particular class of constraints suggests the possibility of obtaining better linear relaxation with fewer additional variables by leveraging the particular bounds given on the original variables. This method can be easily adapted for different types of multilinear problems.

In conclusion, while each technique proposed in this thesis has been presented within a specific combinatorial optimization problem, the methods devised in tackling the examined problems could be easily generalized to a broad range of different applications and raise interesting theoretical questions.

APPENDICES

A Steiner Tree Problem: further details

A.a Enumerating vertices with Polymake and weakness of the BCR formulation

As discussed in the previous section, we aim to solve the Gap problem on every vertex. Hence, we need an exhaustive list of vertices of the polytope $P_{BCR}(n, t)$ for each $n \geq 3$, for each $3 \leq t \leq n - 1$. We use the software Polymake [53], designed for managing polytope and polyhedron. We implement the Gap function in Python, using the commercial solver Gurobi 11.0 [63] to model and solve the Gap ILP model.

On every vertex, we solve the Gap problem to get the maximum possible value of the integrality gap associated with that vertex. Table A.1 reports our computational results. From these results, we can draw several conclusions. First, Polymake can only exhaustively generate vertices for n up to 5. Second, for all these cases, the value of the integrality gap is exactly 1. For larger values of n , the enumeration becomes computationally untractable. Furthermore, by running the Gap problem on many vertices of the BCR formulation, we observe that the problem turns out to be infeasible. By analyzing the minimum infeasibility set, we observe that many vertices of the BCR formulation are incompatible with the triangle inequality of the cost vector c nor with its non-negativity. We tackle both issues in the following sections by (a) introducing a novel formulation tailored for the metric case and (b) designing two heuristic algorithms for enumerating nontrivial vertices.

Table A.1: Number of feasible and optimal vertices for P_{BCR} and P_{CM} . The column “time” reports the time of the generation in seconds, while the column “gap” reports the maximum gap obtained for the optimal vertices. While the BCR polytope has several (feasible) vertices that cannot be optimal for any metric cost, the CM polytope does not suffer this issue (and it implicitly reduces the number of isomorphic vertices).

n	t	time	P_{BCR}			time	P_{CM}		
			feas.	opt.	gap		feas.	opt.	gap
4	3	0.04	256	70	1.00	0.73	4	4	1.00
5	3	4563.57	28 345	3 655	1.00	44.62	5	5	1.00
5	4	2798.17	24 297	3 645	1.00	37.01	44	44	1.00

A.b Enumerating vertices with Polymake for the CM formulation

We enumerate all the vertices of the CM formulation using the software Polymake [8]. Recalling what we have done in Appendix A.a, we compute the gap value for each vertex by implementing the model (1.23) using Gurobi 11.0.0 [63].

Table A.1 reports our computational results. First, we can observe that the number of vertices generated is smaller than the number of vertices of the BCR formulation, and all of them are feasible. As expected, the integrality gap is 1 (Note that it must be a lower bound w.r.t the one of the BCR formulation, which was 1). Note also that, even in this case, Polymake cannot generate vertices for $n \geq 6$. These limited results motivate the design of the two heuristic algorithms to generate nontrivial vertices introduced in the next section.

A.c Pure one-quarter algorithm

Algorithm 3 Pure one-quarter vertices search

```

1: procedure POQ( $n, t$ )
2:  $\mathbb{G} = \{G = (V, E) \mid G \text{ connected, } \deg(i) \geq 3 \forall i \in V, |V| = n, |E| = n + 3t - 4\}$ 
3:  $\text{di}\mathbb{G} = \emptyset$ 
4: for  $G = (V, E) \in \mathbb{G}$  do
5:   if  $|\{i \in V \mid \deg(i) = 3\}| \leq n - t$  then
6:     add to  $\text{di}\mathbb{G}$  every non-isomorphic orientation of  $G$  s.t.
7:       · every edge can be oriented in only one way
8:       · every node has a maximum indegree of 4
9:    $\mathcal{V} = \emptyset$ 
10:  for  $\text{di}G = (V, A) \in \text{di}\mathbb{G}$  do
11:    if  $|\{i \in V \mid \text{indeg}(i) = 0\}| = 1$  then
12:      if  $|\{i \in V \mid \text{indeg}(i) = 1\}| = n - t$  then
13:        if  $|\{i \in V \mid \text{indeg}(i) = 4\}| = t - 1$  then
14:           $x_{ij} = 1/4$  iff  $(i, j) \in A$  is a solution of  $P_{CM}(n, t)$  with
15:            ·  $\{r\} = \{i \in V \mid \text{indeg}(i) = 0\}$ 
16:            ·  $V \setminus T = \{i \in V \mid \text{indeg}(i) = 1\}$ 
17:            ·  $T \setminus \{r\} = \{i \in V \mid \text{indeg}(i) = 4\}$ 
18:          if  $x$  is a feasible vertex of  $P_{CM}(n, t)$  then
19:            add  $x$  to  $\mathcal{V}$ 
20: return  $\mathcal{V}$ 

```

We defined Algorithm 3: a modified version of the PHI algorithm to find POQ vertices exploiting the properties described in Section 1.5.3.

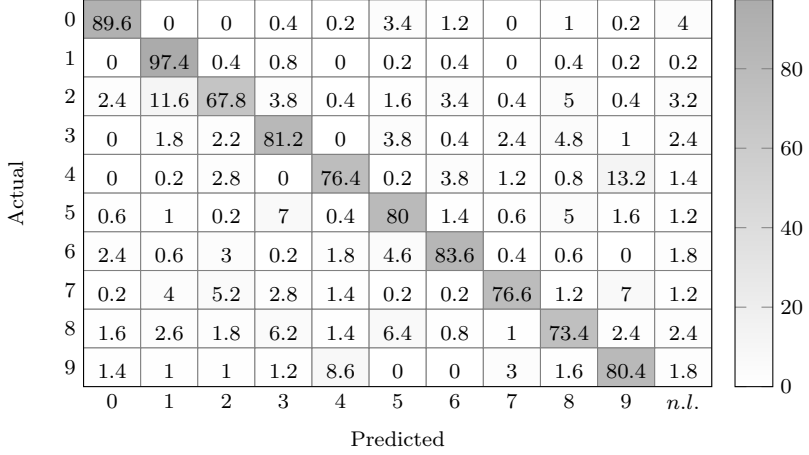


Figure B.5: Confusion matrix of the BeMi ensemble trained with 40 images per digits with the architecture [784, 4, 4, 1]. We reported the percentages for each couple. Note that the *n.l.* indicate the unclassified images.

B Further Neural Network Experiments and Generalizations

B.a Complement to Neural Network Experiments

We first performed an experiment with the *BeMi* ensemble trained with 40 images per digit and we reported the results in a confusion matrix, depicted in Figure B.5. This gave us an idea of what digits were easy or difficult to distinguish between. We then replicate Experiment 1 with a different couple of digits, namely 1 and 8, which are easier to distinguish than the digits 4 and 9. Results are depicted in Figure B.6. We can draw the same conclusions of Experiment 1.

B.b Ensemble Generalization

The definition of the ensemble introduced in Section 2.4.2 can be generalize as follows. Define $\mathcal{P}(\mathcal{I})_m$ as the set of all the subsets of the set \mathcal{I} that have cardinality m , where \mathcal{I} is the set of the classes of the classification problem. Then our structured ensemble is constructed in the following way:

1. We set a parameter $1 < m \leq n = |\mathcal{I}|$.
2. We train a INN denoted by $\mathcal{N}_{\mathcal{J}}$ for every $\mathcal{J} \in \mathcal{P}(\mathcal{I})_m$.
3. When testing a data point, we feed it to our list of trained INNs, namely $(\mathcal{N}_{\mathcal{J}})_{\mathcal{J} \in \mathcal{P}(\mathcal{I})_m}$, obtaining a list of predicted labels $(\mathbf{c}_{\mathcal{J}})_{\mathcal{J} \in \mathcal{P}(\mathcal{I})_m}$.

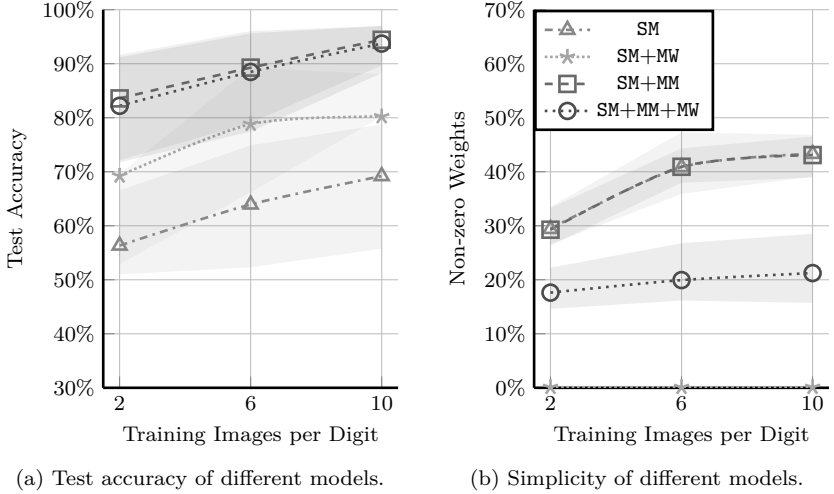


Figure B.6: Ablation study with a different couple of digits, namely 1 and 8, which are easier to distinguish.

4. We then apply a majority voting system.

Note that we set $m > 1$, otherwise our structured ensemble would have been meaningless. Whenever $m = n$, our ensemble is made of one single INN. When $m = 2$, we are using the one-versus-one scheme presented in the paper.

The idea behind this structured ensemble is that, given an input \mathbf{x}^k labelled l ($= y^k$), the input is fed into $\binom{n}{m}$ networks where $\binom{n-1}{m-1}$ of them are trained to recognize an input with label l . If all of the networks correctly classify the input \mathbf{x}^k , then at most $\binom{n-1}{m-1} - \binom{n-2}{m-2}$ other networks can classify the input with a different label. With this approach, if we plan to use $r \in \mathbb{N}$ inputs for each label, we are feeding our INNs a total of $m \cdot r$ inputs instead of feeding $n \cdot r$ inputs to a single large INN. When $m \ll n$, it is much easier to train our structured ensemble of INNs rather than training one large INN.

Majority voting system. After the training, we feed one input \mathbf{x}^k to our list of INNs, and we need to elaborate on the set of outputs.

Definition 6 (Dominant label). For every $b \in \mathcal{I}$, we define

$$C_b = \{\mathcal{J} \in \mathcal{P}(\mathcal{I})_m \mid \mathbf{c}_{\mathcal{J}} = b\},$$

and we say that a label b is a *dominant label* if $|C_b| \geq |C_l|$ for every $l \in \mathcal{I}$. We then define the set of dominant labels

$$\mathcal{D} := \{b \in \mathcal{I} \mid b \text{ is a dominant label}\}.$$

Using this definition, we can have three possible outcomes:

- (a) There exists a label $b \in \mathcal{I}$ such that $\mathcal{D} = \{b\} \implies$ our input is labelled as b .
- (b) There exist $b_1, \dots, b_m \in \mathcal{I}$, $b_i \neq b_j$ for all $i \neq j$ such that $\mathcal{D} = \{b_1, \dots, b_m\}$, so $\mathcal{D} \in \mathcal{P}(\mathcal{I})_m \implies$ our input is labelled as $\mathbf{e}_{\{b_1, \dots, b_m\}} = \mathbf{e}_{\mathcal{D}}$.
- (c) $|\mathcal{D}| \neq 1 \wedge |\mathcal{D}| \neq m \implies$ our input is labelled as $z \notin \mathcal{I}$.

While case (a) is straightforward, we can label our input even when we do not have a clear winner, that is, when we have trained a INN on the set of labels that are the most frequent (i.e., case (b)). Note that the proposed structured ensemble alongside its voting scheme can also be exploited for regular NNs.

Definition 7 (Label statuses). In our labelling system, when testing an input seven different cases, herein called *label statuses*, can arise. The statuses names are of the form “number of the dominant labels + fairness of the prediction”. The first parameter can be 1, m , or o , where o means ‘other cases’. The fairness of the prediction is C when it is correct, or I when it is incorrect. The subscripts related to I' and I'' only distinguish between different cases. These cases are described through the tree diagram depicted in Figure B.7.

The reason behind this generalisation is the following. Suppose to have trained a NN to distinguish between 3 different classes, c_1, c_2, c_3 . If one class is added to the problem, namely c_4 , instead of discarding the trained NN, one could train 3 other NNs, namely the one that distinguishes between c_1, c_2, c_4 , another one for c_1, c_3, c_4 , and the last one for c_2, c_3, c_4 , and use the majority voting scheme. Note that the training of the smaller networks is linked to smaller MILPs: in fact, if we plan to use 10 input data for each class, we need 40 input data in total, but every network is fed with only 30 of them. Moreover, the training of the three additional NNs can be done in parallel, saving computational time.

C Linear formulation of the SMIP model $\mathcal{B}_{ij}^{II}(\omega)$

In this section we report linearization formulation of non-linear constraints in the SMIP model $\mathcal{B}_{ij}^{II}(\omega)$. Big-M and a small constant $\epsilon > 0$ are used in several constraints, while auxiliary variables are defined for each specific group. Linearization of constraints (3.3p) is trivial, then we omit it. Constraints (3.3b) are linearized by introducing the following constraints, in which auxiliary variables $\gamma_i, \hat{\gamma}_i \in \{0, 1\}$ are used:

$$\begin{aligned}
 c_i &\geq t_{i'} - M(1 - \gamma_i) - M(1 - o_{ii'}), & i, i', i \neq i' \in I_{jk}, \\
 c_i &\leq t_{i'} + M\gamma_i - \epsilon(1 - \gamma_i) + M(1 - o_{ii'}), & i, i', i \neq i' \in I_{jk}, \\
 q_{i'} &\geq t_{i'} - M\gamma_i - M(1 - o_{ii'}), & i, i', i \neq i' \in I_{jk},
 \end{aligned}$$

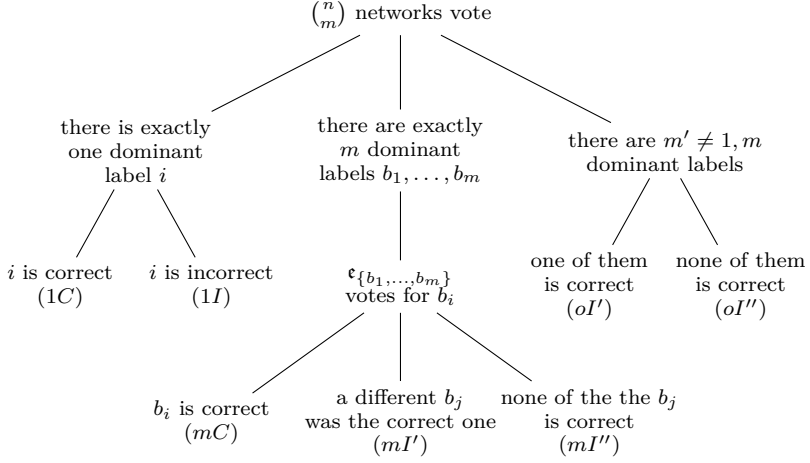


Figure B.7: Tree diagram representing the label statuses for the generalized ensemble.

$$\begin{aligned}
 q_{i'} &\leq t_{i'} + M\gamma_i + M(1 - o_{ii'}), & i, i', i \neq i' \in I_{jk}, \\
 q_{i'} &\geq c_i - M(1 - \gamma_i) - M(1 - o_{ii'}), & i, i', i \neq i' \in I_{jk}, \\
 q_{i'} &\leq c_i + M(1 - \gamma_i) + M(1 - o_{ii'}), & i, i', i \neq i' \in I_{jk}, \\
 \hat{c}_i &\geq t_{i'} - M(1 - \hat{\gamma}_i) - M(1 - o_{ii'}), & i, i', i \neq i' \in I_{jk}, \\
 \hat{c}_i &\leq t_{i'} + M\hat{\gamma}_i - \epsilon(1 - \hat{\gamma}_i) + M(1 - o_{ii'}), & i, i', i \neq i' \in I_{jk}, \\
 \hat{q}_{i'} &\geq t_{i'} - M\hat{\gamma}_i - M(1 - o_{ii'}), & i, i', i \neq i' \in I_{jk}, \\
 \hat{q}_{i'} &\leq t_{i'} + M\hat{\gamma}_i + M(1 - o_{ii'}), & i, i', i \neq i' \in I_{jk}, \\
 \hat{q}_{i'} &\geq \hat{c}_i - M(1 - \hat{\gamma}_i) - M(1 - o_{ii'}), & i, i', i \neq i' \in I_{jk}, \\
 \hat{q}_{i'} &\leq \hat{c}_i + M(1 - \hat{\gamma}_i) + M(1 - o_{ii'}), & i, i', i \neq i' \in I_{jk}.
 \end{aligned}$$

Constraints (3.3j) are linearized by introducing the following constraints, in which the auxiliary variables $\Lambda_i^{(1)}, \Lambda_i^{(2)} \in \{0, 1\}$ are used:

$$\begin{aligned}
 \tau_{jk}(\omega) - \hat{q}_{i'} &\geq \epsilon - M\Lambda_i^{(2)} + \epsilon(1 - \Lambda_i^{(2)}) - M(1 - o_{ii'}), & i, i', i \neq i' \in I_{jk}, \\
 \tau_{jk}(\omega) - \hat{q}_{i'} &\leq \epsilon + M(1 - \Lambda_i^{(2)}) + M(1 - o_{ii'}), & i, i', i \neq i' \in I_{jk}, \\
 \tau_{jk}(\omega) - \hat{q}_{i'} &\geq -M(1 - \Lambda_i^{(1)}), & i \in I_{jk}, \\
 \tau_{jk}(\omega) - \hat{q}_{i'} &\leq M\Lambda_i^{(1)} - \epsilon(1 - \Lambda_i^{(1)}), & i \in I_{jk}, \\
 \Lambda_i^{(1)} + \Lambda_i^{(2)} &\leq 1 + e_i, & i \in I_{jk}, \\
 e_i &\leq \Lambda_i^{(\ell)}, & i \in I_{jk}, \ell = 1, 2.
 \end{aligned}$$

Constraints (3.3k) are linearized by introducing the following constraints, in which auxiliary variables $\lambda_i^{(e)}, \lambda_i^{(\tau)} \in \{0, 1\}$ are used:

$$\begin{aligned}
\tau_{jk}(\omega) &\leq q_i + \rho_i(\omega)\theta_i(\omega)y_i + M\lambda_i^{(\tau)} - \epsilon(1 - \lambda_i^{(\tau)}), & i \in I_{jk}, \\
\tau_{jk}(\omega) &\geq q_i + \rho_i(\omega)\theta_i(\omega)y_i - M(1 - \lambda_i^{(\tau)}), & i \in I_{jk}, \\
z_i &\leq \tau_{jk}(\omega) - (q_i + \rho_i(\omega)\theta_i(\omega)y_i) + M(1 - \lambda_i^{(e)}), & i \in I_{jk}, \\
z_i &\geq \tau_{jk}(\omega) - (q_i + \rho_i(\omega)\theta_i(\omega)y_i) - M(1 - \lambda_i^{(e)}), & i \in I_{jk}, \\
e_i + \lambda_i^{(\tau)} &\leq 1 + \lambda_i^{(e)}, & i \in I_{jk}, \\
z_i &\leq M\lambda_i^{(\tau)}, & i \in I_{jk}.
\end{aligned}$$

Finally, constraints (3.3l) are linearized by introducing the following constraints:

$$\begin{aligned}
\theta_i(\omega)(q_i + \mu_i) &\leq L_{jk} + H + M(1 - y_i), & i \in I_{jk}, \\
\theta_i(\omega)(q_i + \mu_i) &\geq L_{jk} + H - My_i + \epsilon(1 - y_i), & i \in I_{jk}.
\end{aligned}$$

REFERENCES

- [1] Bernardetta Addis, Giuliana Carello, Andrea Grosso, and Elena Tànfani. “Operating room scheduling and rescheduling: a rolling horizon approach”. In: *Flexible Services and Manufacturing Journal* 28 (2016), pp. 206–232 (cit. on p. 118).
- [2] V. Agrawal, Y. Zhang, and P. S. Sundararaghavan. “Multi-criteria surgery scheduling optimization using modeling, heuristics, and simulation”. In: *Healthcare Analytics* 2 (2022), p. 100034 (cit. on pp. 68, 71, 73).
- [3] Ross Anderson, Joey Huchette, Will Ma, Christian Tjandraatmadja, and Juan Pablo Vielma. “Strong mixed-integer programming formulations for trained neural networks”. In: *Mathematical Programming* 183.1 (2020), pp. 3–39 (cit. on pp. 42, 43).
- [4] Roberto Aringhieri and Davide Duma. “Patient-Centred Objectives as an Alternative to Maximum Utilisation: Comparing Surgical Case Solutions”. In: *Springer Proceedings in Mathematics and Statistics* 217 (2017), pp. 105–112 (cit. on p. 68).
- [5] Roberto Aringhieri and Davide Duma. “The optimization of a surgical clinical pathway”. In: *Advances in Intelligent Systems and Computing* 402 (2015), pp. 313–331 (cit. on p. 74).
- [6] Roberto Aringhieri, Davide Duma, and Enrico Faccio. “Ex post evaluation of an operating theatre”. In: *Electronic Notes in Discrete Mathematics* 69 (2018), pp. 157–164 (cit. on p. 118).
- [7] Roberto Aringhieri, Davide Duma, Paolo Landa, and Simona Mancini. “Combining workload balance and patient priority maximisation in operating room planning through hierarchical multi-objective optimisation”. In: *European Journal of Operational Research* 298.2 (2022), pp. 627–643 (cit. on p. 68).
- [8] Benjamin Assarf, Ewgenij Gawrilow, Katrin Herr, Michael Joswig, Benjamin Lorenz, Andreas Paffenholz, and Thomas Rehn. “Computing convex hulls and counting integer points with Polymake”. In: *Mathematical Programming Computation* 9 (2017), pp. 1–38 (cit. on p. 146).
- [9] Macarena Azar, Rodrigo A. Carrasco, and Susana Mondschein. “Dealing with uncertain surgery times in operating room scheduling”. In: *European Journal of Operational Research* 299.1 (2022), pp. 377–394 (cit. on pp. 73, 93).

- [10] Ron Banner, Itay Hubara, Elad Hoffer, and Daniel Soudry. “Scalable methods for 8-bit training of neural networks”. In: *Advances in neural information processing systems (NeurIPS)* 31 (2018), pp. 5145–5153 (cit. on p. 48).
- [11] Sakine Batun, Brian T. Denton, Todd R. Huschka, and Andrew J. Schaefer. “Operating room pooling and parallel surgery processing under uncertainty”. In: *INFORMS Journal on Computing* 23.2 (2011), pp. 220–237 (cit. on pp. 68, 71, 72, 76, 93).
- [12] Tatiana Benaglia, Didier Chauveau, David R. Hunter, and Derek S. Young. “mixtools: An R Package for Analyzing Mixture Models”. In: *Journal of Statistical Software* 32.6 (2009), pp. 1–29. URL: <https://www.jstatsoft.org/index.php/jss/article/view/v032i06> (cit. on p. 94).
- [13] Yoshua Bengio, Andrea Lodi, and Antoine Prouvost. “Machine learning for combinatorial optimization: a methodological tour d’horizon”. In: *European Journal of Operational Research* 290.2 (2021), pp. 405–421 (cit. on p. 43).
- [14] Genevieve Benoit and Sylvia Boyd. “Finding the exact integrality gap for small traveling salesman problems”. In: *Mathematics of Operations Research* 33.4 (2008), pp. 921–931 (cit. on pp. 10–12, 28, 38).
- [15] David Bergman, Teng Huang, Philip Brooks, Andrea Lodi, and Arvind U. Raghunathan. “Janos: an integrated predictive and prescriptive modeling framework”. In: *INFORMS Journal on Computing* 34.2 (2022), pp. 807–816 (cit. on p. 43).
- [16] Ambrogio Maria Bernardelli, Lorenzo Bonasera, Davide Duma, and Eleonora Vercesi. “Multi-objective stochastic scheduling of inpatient and outpatient surgeries”. In: *Flexible Services and Manufacturing Journal* (2024), pp. 1–55 (cit. on p. 4).
- [17] Ambrogio Maria Bernardelli, Stefano Gualandi, Hoong Chuin Lau, and Simone Milanesi. “The BeMi stardust: a structured ensemble of binarized neural networks”. In: *International Conference on Learning and Intelligent Optimization (LION)*. Springer. 2023, pp. 443–458 (cit. on pp. 3, 42).
- [18] Ambrogio Maria Bernardelli, Stefano Gualandi, Simone Milanesi, Hoong Chuin Lau, and Neil Yorke-Smith. *Multi-Objective Linear Ensembles for Robust and Sparse Training of Few-Bit Neural Networks*. 2024. DOI: 10.1287/ijoc.2023.0281.cd. URL: <https://github.com/INFORMSJoc/2023.0281> (cit. on p. 56).
- [19] Ambrogio Maria Bernardelli, Stefano Gualandi, Simone Milanesi, Hoong Chuin Lau, and Neil Yorke-Smith. “Multiobjective Linear Ensembles for Robust and Sparse Training of Few-Bit Neural Networks”. In: *INFORMS Journal on Computing* (2024) (cit. on p. 3).

- [20] Ambrogio Maria Bernardelli, Eleonora Vercesi, Stefano Gualandi, Monaldo Mastrolilli, and Luca Maria Gambardella. “On the integrality gap of the Complete Metric Steiner Tree Problem via a novel formulation”. In: *arXiv preprint arXiv:2405.13773* (2024) (cit. on p. 3).
- [21] Leonora Bianchi, Marco Dorigo, Luca Maria Gambardella, and Walter J. Gutjahr. “A survey on metaheuristics for stochastic combinatorial optimization”. In: *Natural Computing* 8 (2008), pp. 239–287 (cit. on p. 73).
- [22] Daniel Bienstock, Michael Chertkov, and Sean Harnett. “Chance-constrained optimal power flow: Risk-aware network control under uncertainty”. In: *Siam Review* 56.3 (2014), pp. 461–495 (cit. on p. 121).
- [23] Daniel Bienstock and Abhinav Verma. “Strong NP-hardness of AC power flows feasibility”. In: *Operations Research Letters* 47.6 (2019), pp. 494–501 (cit. on p. 121).
- [24] Cristopher M. Bishop and Nasser M. Nasrabadi. *Pattern Recognition and Machine Learning*. Springer, 2006 (cit. on p. 44).
- [25] Davis W. Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John V. Guttag. “What is the state of neural network pruning?” In: *Machine Learning and Systems (MLSYS)*. Vol. 2. 2020, pp. 129–146 (cit. on p. 63).
- [26] Julian Blank and Kalyanmoy Deb. “Pymoo: Multi-Objective Optimization in Python”. In: *IEEE Access* 8 (2020), pp. 89497–89509 (cit. on p. 97).
- [27] M. Blott, L. Halder, M. Leeser, and L. Doyle. “Qutibench: Benchmarking neural networks on heterogeneous hardware”. In: *ACM Journal on Emerging Technologies in Computing Systems (JETC)* 15.4 (2019), pp. 1–38 (cit. on p. 41).
- [28] Elena Botoeva, Panagiotis Kouvaros, Jan Kronqvist, Alessio Lomuscio, and Ruth Misener. “Efficient verification of relu-based neural networks via dependency analysis”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 2020, pp. 3291–3299 (cit. on p. 43).
- [29] G.J. Bowden, P.R. Barker, V.O. Shestopal, and J.W. Twidell. “The Weibull distribution function and wind power statistics”. In: *Wind Engineering* (1983), pp. 85–98 (cit. on p. 132).
- [30] Sylvia Boyd and Paul Elliott–Magwood. “Computing the integrality gap of the asymmetric travelling salesman problem”. In: *Electronic Notes in Discrete Mathematics* 19 (2005), pp. 241–247 (cit. on pp. 9, 38).

- [31] Sylvia Boyd and Paul Elliott–Magwood. “Structure of the extreme points of the subtour elimination polytope of the STSP”. In: *Combinatorial Optimization and Discrete Algorithms* 23 (2007), pp. 33–47 (cit. on pp. 9, 12, 28).
- [32] Lorenzo Brigato, Björn Barz, Luca Iocchi, and Joachim Denzler. “Image classification with small datasets: overview and benchmark”. In: *IEEE Access* 10 (2022), pp. 49233–49250 (cit. on p. 65).
- [33] Jarosław Byrka, Fabrizio Grandoni, Thomas Rothvoß, and Laura Sanità. “Steiner tree approximation via iterative randomized rounding”. In: *Journal of the ACM (JACM)* 60.1 (2013), pp. 1–33 (cit. on pp. 7, 8).
- [34] Jarosław Byrka, Fabrizio Grandoni, and Vera Traub. *The Bidirected Cut Relaxation for Steiner Tree has Integrality Gap Smaller than 2*. 2024. arXiv: 2407.19905 [cs.DS]. URL: <https://arxiv.org/abs/2407.19905> (cit. on p. 8).
- [35] Junyang Cai, Khai-Nguyen Nguyen, Nishant Shrestha, Aidan Good, Ruisen Tu, Xin Yu, Shandian Zhe, and Thiago Serra. “Getting away with more network pruning: From sparsity to geometry and linear regions”. In: *International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR)*. Springer. 2023, pp. 200–218 (cit. on pp. 42, 43).
- [36] Quentin Cappart, Didier Chételat, Elias B. Khalil, Andrea Lodi, Christopher Morris, and Petar Veličković. “Combinatorial optimization and reasoning with graph neural networks”. In: *Journal of Machine Learning Research* 24.130 (2023), pp. 1–61 (cit. on p. 43).
- [37] Brecht Cardoen, Erik Demeulemeester, and Jeroen Beliën. “Optimizing a multiple objective surgical case sequencing problem”. In: *International Journal of Production Economics* 119.2 (2009), pp. 354–366 (cit. on pp. 68, 93).
- [38] Jacques Carpentier. “Optimal power flows”. In: *International Journal of Electrical Power & Energy Systems* 1.1 (1979), pp. 3–15 (cit. on p. 121).
- [39] Batuhan Çelik, Serhat Gul, and Melih Çelik. “A stochastic programming approach to surgery scheduling under parallel processing principle”. In: *Omega (United Kingdom)* 115 (2023) (cit. on p. 70).
- [40] Anurag Chauhan and R.P. Saini. “A review on Integrated Renewable Energy System based power generation for stand-alone applications: Configurations, storage options, sizing methodologies and control”. In: *Renewable and Sustainable Energy Reviews* 38 (2014), pp. 99–120 (cit. on p. 121).
- [41] Sunil Chopra and Chih-Yang Tsai. “Polyhedral approaches for the Steiner tree problem on graphs”. In: *Steiner trees in industry*. Springer, 2001, pp. 175–201 (cit. on pp. 14, 15).

- [42] Stefan Creemers, Jeroen Beliën, and Marc Lambrecht. “The optimal allocation of server time slots over different classes of patients”. In: *European Journal of Operational Research* 219.3 (2012), pp. 508–521 (cit. on p. 76).
- [43] Brian Denton, James Viapiano, and Andrea Vogl. “Optimization of surgery sequencing and scheduling decisions under uncertainty”. In: *Health Care Management Science* 10.1 (2007), pp. 13–24 (cit. on p. 70).
- [44] Edsger W. Dijkstra. “A note on two problems in connexion with graphs”. In: *Numerische Mathematik* 1 (1959), pp. 269–271. URL: <https://api.semanticscholar.org/CorpusID:123284777> (cit. on p. 11).
- [45] Davide Duma and Roberto Aringhieri. “An online optimization approach for the Real Time Management of operating rooms”. In: *Operations Research for Health Care* 7 (2015), pp. 40–51 (cit. on pp. 71, 72).
- [46] Davide Duma and Roberto Aringhieri. “The management of non-elective patients: shared vs. dedicated policies”. In: *Omega (United Kingdom)* 83 (2019), pp. 199–212 (cit. on pp. 71, 72, 74, 75, 77, 119).
- [47] Davide Duma and Roberto Aringhieri. “The real time management of operating rooms”. In: *International Series in Operations Research and Management Science* 262 (2018), pp. 55–79 (cit. on p. 119).
- [48] Jack Edmonds et al. “Optimum branchings”. In: *Journal of Research of the National Bureau of Standards B* 71.4 (1967), pp. 233–240 (cit. on pp. 7, 11).
- [49] Mostafa ElAraby, Guy Wolf, and Margarida Carvalho. “OAMIP: Optimizing ANN Architectures Using Mixed-Integer Programming”. In: *International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR)*. Springer. 2023, pp. 219–237 (cit. on pp. 42, 43).
- [50] Paul Elliott–Magwood. “The integrality gap of the asymmetric travelling salesman problem”. PhD thesis. University of Ottawa (Canada), 2008 (cit. on pp. 11, 12, 28).
- [51] Matteo Fischetti and Jason Jo. “Deep neural networks and mixed integer linear optimization”. In: *Constraints* 23.3 (2018), pp. 296–309 (cit. on pp. 42, 43).
- [52] Gerald Gamrath, Thorsten Koch, Stephen J. Maher, Daniel Rehfeldt, and Yuji Shinano. “SCIP-Jack—a solver for STP and variants with parallelization extensions”. In: *Mathematical Programming Computation* 9 (2017), pp. 231–296 (cit. on pp. 8, 13).

- [53] Ewgenij Gawrilow and Michael Joswig. “Polymake: a framework for analyzing convex polytopes”. In: *Polytopes—combinatorics and computation*. Springer, 2000, pp. 43–73 (cit. on p. 145).
- [54] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. “A survey of quantization methods for efficient neural network inference”. In: *Low-Power Computer Vision*. Chapman and Hall/CRC, 2022, pp. 291–326 (cit. on pp. 47, 48).
- [55] Michel Xavier Goemans and Young-Soo Myung. “A catalog of Steiner tree formulations”. In: *Networks* 23.1 (1993), pp. 19–28 (cit. on pp. 7, 8, 11).
- [56] E. Gökalp, N. Gülpınar, and Xuan Vinh Doan. “Dynamic Surgery Management under Uncertainty”. In: *European Journal of Operational Research* (2023) (cit. on p. 119).
- [57] José Fernando Gonçalves and Mauricio G. C. Resende. “Biased random-key genetic algorithms for combinatorial optimization”. In: *Journal of Heuristics* 17.5 (2011), pp. 487–525 (cit. on pp. 73, 90).
- [58] Aidan Good, Jiaqi Lin, Xin Yu, Hannah Sieg, Mikey Fergusson, Shandian Zhe, Jerzy Wiecek, and Thiago Serra. “Recall distortion in neural network pruning and the undecayed pruning algorithm”. In: *Advances in Neural Information Processing Systems (NeurIPS)* 35 (2022), pp. 32762–32776 (cit. on p. 43).
- [59] Martin Grötschel, Alexander Martin, and Robert Weismantel. “The Steiner tree packing problem in VLSI design”. In: *Mathematical Programming* 78 (1997), pp. 265–281 (cit. on p. 8).
- [60] Serhat Gul, Brian T. Denton, John W. Fowler, and Todd Huschka. “Bi-criteria scheduling of surgical services for an outpatient procedure center”. In: *Production and Operation Management* 20.3 (2011), pp. 406–417 (cit. on p. 93).
- [61] Yi Guo, Kyri Baker, Emiliano Dall’Anese, Zechun Hu, and Tyler Holt Summers. “Data-based distributionally robust stochastic optimal power flow—Part I: Methodologies”. In: *IEEE Transactions on Power Systems* 34.2 (2018), pp. 1483–1492 (cit. on p. 122).
- [62] Yi Guo, Kyri Baker, Emiliano Dall’Anese, Zechun Hu, and Tyler Holt Summers. “Data-based distributionally robust stochastic optimal power flow—Part II: Case studies”. In: *IEEE Transactions on Power Systems* 34.2 (2018), pp. 1493–1503 (cit. on pp. 126, 132, 134).
- [63] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*. 2023. URL: <https://www.gurobi.com> (cit. on pp. 33, 56, 145, 146).

- [64] Song Han, Huizi Mao, and William J. Dally. “Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding”. In: *arXiv preprint arXiv:1510.00149* (2015) (cit. on p. 63).
- [65] Sean Harris and David Claudio. “Current Trends in Operating Room Scheduling 2015 to 2020: a Literature Review”. In: *Operations Research Forum* 3.1 (2022) (cit. on p. 67).
- [66] Stefan Hougardy, Jannik Silvanus, and Jens Vygen. “Dijkstra meets Steiner: a fast exact goal-oriented Steiner tree algorithm”. In: *Mathematical Programming Computation* 9.2 (2017), pp. 135–202 (cit. on p. 8).
- [67] Taoan Huang, Aaron M. Ferber, Yuandong Tian, Bistra Dilkina, and Benoit Steiner. “Searching Large Neighborhoods for Integer Linear Programs with Contrastive Learning”. In: *International Conference on Machine Learning (ICML)*. Vol. 202. Proceedings of Machine Learning Research. PMLR, 2023, pp. 13869–13890. URL: <https://proceedings.mlr.press/v202/huang23g.html> (cit. on p. 41).
- [68] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. “Binarized neural networks”. In: *Advances in Neural Information Processing Systems (NeurIPS)* 29 (2016), pp. 4107–4115 (cit. on pp. 41, 45, 47, 58).
- [69] Joey Huchette, Gonzalo Muñoz, Thiago Serra, and Calvin Tsay. “When deep learning meets polyhedral theory: A survey”. In: *arXiv preprint arXiv:2305.00241* (2023) (cit. on p. 43).
- [70] Rabih A. Jabr. “Radial distribution load flow using conic programming”. In: *IEEE transactions on power systems* 21.3 (2006), pp. 1458–1459 (cit. on pp. 122, 123, 136).
- [71] A. Janosi, W. Steinbrunn, M. Pfisterer, and R. Detrano. *Heart Disease Dataset*. 1988. URL: <http://archive.ics.uci.edu/ml/datasets/Heart+Disease> (cit. on pp. 55, 64).
- [72] Aida Jebali, Atidel B. Hadj Alouane, and Pierre Ladet. “Operating rooms scheduling”. In: *International Journal of Production Economics* 99.1-2 (2006), pp. 52–56 (cit. on p. 70).
- [73] Yiding Jiang, Dilip Krishnan, Hossein Mobahi, and Samy Bengio. “Predicting the Generalization Gap in Deep Networks with Margin Distributions”. In: *International Conference on Learning Representations (ICLR)*. 2019 (cit. on p. 46).
- [74] Richard M. Karp. *Reducibility among combinatorial problems*. Springer, 2010 (cit. on p. 7).

- [75] Kenji Kawaguchi, Yoshua Bengio, and Leslie Pack Kaelbling. “Generalization in Deep Learning”. In: *Mathematical Aspects of Deep Learning*. Cambridge University Press, Dec. 2022, pp. 112–148. DOI: 10.1017/9781009025096.003. URL: <http://dx.doi.org/10.1017/9781009025096.003> (cit. on p. 46).
- [76] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. “On large-batch training for deep learning: Generalization gap and sharp minima”. In: *International Conference on Learning Representations (ICLR)*. Vol. 5. 2017 (cit. on p. 46).
- [77] Elias B. Khalil, Amrita Gupta, and Bistra Dilkina. “Combinatorial Attacks on Binarized Neural Networks”. In: *International Conference on Learning Representations (ICLR)*. 2019 (cit. on p. 42).
- [78] Anton J. Kleywegt, Alexander Shapiro, and Tito Homem-de-Mello. “The Sample Average Approximation Method for Stochastic Discrete Optimization”. In: *SIAM Journal on Optimization* 12.2 (2002), pp. 479–502 (cit. on p. 89).
- [79] Thorsten Koch and Alexander Martin. “Solving Steiner tree problems in graphs to optimality”. In: *Networks: An International Journal* 32.3 (1998), pp. 207–232 (cit. on pp. 7, 8, 13).
- [80] Thorsten Koch, Alexander Martin, and Stefan Voß. “SteinLib: An updated library on Steiner Tree problems in graphs”. In: *Steiner Trees in Industry*. Ed. by Xiu Zhen Cheng and Ding-Zhu Du. Boston, MA: Springer US, 2001, pp. 285–325. ISBN: 978-1-4613-0255-1. DOI: 10.1007/978-1-4613-0255-1_9. URL: https://doi.org/10.1007/978-1-4613-0255-1_9 (cit. on p. 16).
- [81] Burak Kocuk, Santanu S Dey, and X Andy Sun. “Strong SOCP relaxations for the optimal power flow problem”. In: *Operations Research* 64.6 (2016), pp. 1177–1196 (cit. on pp. 122, 137).
- [82] Jochen Könemann, David Pritchard, and Kunlun Tan. “A partition-based relaxation for Steiner trees”. en. In: *Mathematical Programming* 127.2 (Apr. 2011), pp. 345–370. ISSN: 1436-4646. DOI: 10.1007/s10107-009-0289-2. URL: <https://doi.org/10.1007/s10107-009-0289-2> (visited on 05/14/2024) (cit. on pp. 9, 10, 38).
- [83] Jannis Kurtz and Bubacarr Bah. “Efficient and robust mixed-integer optimization methods for training binarized deep neural networks”. In: *arXiv preprint arXiv:2110.11382* (2021) (cit. on p. 41).
- [84] Paolo Landa, Roberto Aringhieri, Patrick Soriano, Elena Tànfani, and Angela Testi. “A hybrid optimization algorithm for surgeries scheduling”. In: *Operations Research for Health Care* 8 (2016), pp. 103–114 (cit. on pp. 71, 72, 75, 76).

- [85] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *Nature* 521.7553 (2015) (cit. on p. 41).
- [86] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. *The MNIST database of handwritten digits*. 1998. URL: <http://yann.lecun.com/exdb/mnist> (cit. on p. 55).
- [87] Sangbok Lee and Yuehwern Yih. “Reducing patient-flow delays in surgical suites through determining start-times of surgical cases”. In: *European Journal of Operational Research* 238.2 (2014), pp. 620–629 (cit. on p. 70).
- [88] Xiaocheng Li, Chunlin Sun, and Yinyu Ye. “Simple and Fast Algorithm for Binary Integer and Online Linear Programming”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2020, pp. 9412–9421. URL: <https://proceedings.neurips.cc/paper/2020/hash/6abba5d8ab1f4f32243e174beb754661-Abstract.html> (cit. on p. 41).
- [89] Xiaofan Lin, Cong Zhao, and Wei Pan. “Towards accurate binary convolutional neural network”. In: *Advances in Neural Information Processing Systems (NeurIPS)* 30 (2017), pp. 345–353 (cit. on p. 42).
- [90] Ivana Ljubić. “Solving Steiner trees: Recent advances, challenges, and perspectives”. In: *Networks* 77.2 (2021), pp. 177–204 (cit. on pp. 7, 8).
- [91] Michele Lombardi and Michela Milano. “Boosting combinatorial problem modeling with machine learning”. In: *The 27th International Joint Conference on Artificial Intelligence (IJCAI-ECAI’18)*. 2018, pp. 5472–5478 (cit. on p. 43).
- [92] Daniel P. Loucks. “Chance Constrained and Monte Carlo Modeling”. In: *International Series in Operations Research & Management Science* 318 (2022), pp. 177–185 (cit. on p. 73).
- [93] Camilo Mancilla and Robert Storer. “A sample average approximation approach to stochastic appointment sequencing and scheduling”. In: *IIE Transactions (Institute of Industrial Engineers)* 44.8 (2012), pp. 655–670 (cit. on p. 70).
- [94] Carlo Mannino, Eivind J. Nilssen, and Tomas E. Nordlander. *SINTEF ICT: MSS-adjusts surgery data*. 2010. URL: <https://www.sintef.no/Projectweb/Health-care-optimization/Testbed/> (cit. on pp. 92, 93, 97).
- [95] Inês Marques, M. Eugénia Captivo, and Margarida Vaz Pato. “Scheduling elective surgeries in a Portuguese hospital using a genetic heuristic”. In: *Operations Research for Health Care* 3.2 (2014), pp. 59–72 (cit. on pp. 70, 76).

- [96] Jerrold H. May, William E. Spangler, David P. Strum, and Luis G. Vargas. “The Surgical Scheduling Problem: Current Research and Future Opportunities”. In: *Production and Operations Management* 20 (Jan. 2011), pp. 392–405 (cit. on p. 67).
- [97] Brendan D. McKay and Adolfo Piperno. “Practical graph isomorphism, II”. In: *Journal of Symbolic Computation* 60 (2014), pp. 94–112. DOI: 10.1016/j.jsc.2013.09.003. URL: <https://doi.org/10.1016/j.jsc.2013.09.003> (cit. on p. 31).
- [98] Miten Mistry, Dimitrios Letsios, Gerhard Krennrich, Robert M. Lee, and Ruth Misener. “Mixed-integer convex nonlinear optimization with gradient-boosted trees embedded”. In: *INFORMS Journal on Computing* 33.3 (2021), pp. 1103–1119 (cit. on p. 43).
- [99] Peyman Mohajerin Esfahani and Daniel Kuhn. “Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations”. In: *Mathematical Programming* 171.1 (2018), pp. 115–166 (cit. on pp. 122, 124, 129, 130).
- [100] John Moody. “The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems”. In: *Advances in Neural Information Processing Systems (NeurIPS)* 4 (1991), pp. 847–854 (cit. on p. 46).
- [101] Ari Morcos, Haonan Yu, Michela Paganini, and Yuandong Tian. “One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers”. In: *Advances in Neural Information Processing Systems (NeurIPS)* 32 (2019), pp. 4932–4942 (cit. on p. 63).
- [102] Francis Mwasilu, Jackson John Justo, Eun-Kyung Kim, Ton Duc Do, and Jin-Woo Jung. “Electric vehicles and smart grid interaction: A review on vehicle to grid and renewable energy sources integration”. In: *Renewable and sustainable energy reviews* 34 (2014), pp. 501–516 (cit. on p. 121).
- [103] Mohammad Rasoul Narimani, Daniel K Molzahn, Harsha Nagarajan, and Mariesa L Crow. “Comparison of various trilinear monomial envelopes for convex relaxations of optimal power flow problems”. In: *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE. 2018, pp. 865–869 (cit. on p. 122).
- [104] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. “Exploring generalization in deep learning”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 30. 2017, pp. 5947–5956 (cit. on p. 46).
- [105] E. Omiling, A. Jarnheimer, J. Rose, J. Björk, J. G. Meara, and L. Hagander. “Population-based incidence rate of inpatient and outpatient surgical procedures in a high-income country”. In: *British Journal of Surgery* 105.1 (2018), pp. 86–95 (cit. on p. 96).

- [106] J. J. Pandit and A. Carey. “Estimating the duration of common elective operations: Implications for operating list management”. In: *Anaesthesia* 61.8 (2006), pp. 768–776 (cit. on pp. 74, 93).
- [107] Vrishabh Patil and Yonatan Mintz. “A Mixed-Integer Programming Approach to Training Dense Neural Networks”. In: *arXiv preprint arXiv:2201.00723* (2022) (cit. on p. 41).
- [108] Sven Peyer. “Shortest paths and Steiner trees in VLSI routing”. PhD thesis. Universitäts-und Landesbibliothek Bonn, 2007 (cit. on p. 8).
- [109] Robert Clay Prim. “Shortest connection networks and some generalizations”. In: *The Bell System Technical Journal* 36.6 (1957), pp. 1389–1401 (cit. on p. 11).
- [110] Daniel J. Quemby and Mary E. Stocker. “Day surgery development and practice: key factors for a successful pathway”. In: *Continuing Education in Anaesthesia, Critical Care & Pain* 14 (2014), pp. 256–261 (cit. on p. 67).
- [111] Gabor Riccardi. *Jabr OPF*. <https://github.com/frulcino/Jabr-OPF-models>. 2023 (cit. on p. 132).
- [112] R. Tyrrell Rockafellar and Stanislav Uryasev. “Optimization of conditional value-at-risk”. In: *Journal of risk* 2 (2000), pp. 21–42 (cit. on p. 128).
- [113] Carles Roger, Riera Molina, Camilo Rey, Thiago Serra, Eloi Puer-tas, and Oriol Pujol. “Training Thinner and Deeper Neural Networks: Jumpstart Regularization”. In: *Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR)*. Vol. 13292. Springer, 2022, pp. 345–357 (cit. on p. 46).
- [114] Vahid Roshanaei, Kyle E. C. Booth, Dionne M. Aleman, David R. Urbach, and J. Christopher Beck. “Branch-and-check methods for multi-level operating room planning and scheduling”. In: *International Journal of Production Economics* 220 (2020) (cit. on p. 70).
- [115] Hadhemi Saadouli, Badreddine Jerbi, Abdelaziz Dammak, Lotfi Mas-moudi, and Abir Bouaziz. “A stochastic optimization and simulation approach for scheduling operating rooms and recovery beds in an ortho-pedic surgery department”. In: *Computers and Industrial Engineering* 80 (2015), pp. 72–79 (cit. on p. 68).
- [116] Charbel Sakr, Jungwook Choi, Zhuo Wang, Kailash Gopalakrishnan, and Naresh Shanbhag. “True gradient-based training of deep binary activated neural networks via continuous binarization”. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2018, pp. 2346–2350 (cit. on p. 42).

- [117] Michael Samudra, Carla Van Riet, Erik Demeulemeester, Brecht Cardoen, Nancy Vansteenkiste, and Frank Rademakers. “Scheduling operating rooms: achievements, challenges and pitfalls”. In: *Journal of Scheduling* 19.5 (2016), pp. 493–525 (cit. on p. 68).
- [118] Emily Schutte and Matthias Walter. “Relaxation strength for multi-linear optimization: McCormick strikes back”. In: *International Conference on Integer Programming and Combinatorial Optimization*. Springer. 2024, pp. 393–404 (cit. on p. 138).
- [119] Thiago Serra, Abhinav Kumar, and Srikumar Ramalingam. “Loss-less Compression of Deep Neural Networks”. In: *Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR)*. Vol. 17. 2020, pp. 417–430 (cit. on p. 42, 43).
- [120] Thiago Serra, Xin Yu, Abhinav Kumar, and Srikumar Ramalingam. “Scaling Up Exact Neural Network Compression by ReLU Stability”. In: *Advances in Neural Information Processing Systems (NeurIPS)* (2021), pp. 27081–27093 (cit. on p. 46).
- [121] Karmel S. Shehadeh. “Data-Driven Distributionally Robust Surgery Planning in Flexible Operating Rooms Over a Wasserstein Ambiguity”. In: *Computers & Operations Research* 146 (2022) (cit. on p. 73, 97).
- [122] Karmel S. Shehadeh and Luis F. Zuluaga. *14th AIMMS-MOPTA Optimization Modeling Competition. Surgery Scheduling in Flexible Operating Rooms Under Uncertainty*. 2022. URL: <https://iccopt2022.lehigh.edu/competition-and-prizes/aimms-mopta-competition/> (cit. on p. 67).
- [123] Belinda Spratt and Erhan Kozan. “A real-time reactive framework for the surgical case sequencing problem”. In: *Flexible Services and Manufacturing Journal* 33 (2021) (cit. on p. 119).
- [124] David P. Strum, Jerrold H. May, Allan R. Sampson, Luis G. Vargas, and William E. Spangler. “Estimating times of surgeries with two component procedures comparison of the lognormal and normal models”. In: *Anesthesiology* 98.1 (2003), pp. 232–240 (cit. on p. 93).
- [125] Aidan L. Tan, Calvin J. Chiew, Sijia Wang, Hairil Rizal Abdullah, Sean S. W. Lam, Marcus E. H. Ong, Hiang Khoon Tan, and Ting Hway Wong. “Risk factors and reasons for cancellation within 24h of scheduled elective surgery in an academic medical centre: A cohort study”. In: *International Journal of Surgery* 66 (2019), pp. 72–78 (cit. on p. 74).

- [126] Kang Miao Tan, Thanikanti Sudhakar Babu, Vigna K. Ramachandaramurthy, Padmanathan Kasinathan, Sunil G. Solanki, and Shangari K. Raveendran. “Empowering smart grid: A comprehensive review of energy storage technology and application with renewable energy integration”. In: *Journal of Energy Storage* 39 (2021), p. 102591 (cit. on p. 121).
- [127] Wei Tang, Gang Hua, and Liang Wang. “How to train a compact binary neural network with high accuracy?” In: *Thirty-First AAAI Conference on Artificial Intelligence*. 2017, pp. 2625–2631 (cit. on p. 42).
- [128] Angela Testi, Elena Tànfani, and Giancarlo Torre. “A three-phase approach for operating theatre schedules”. In: *Health Care Management Science* 10.2 (2007), pp. 163–172 (cit. on pp. 67, 71, 72, 76, 93).
- [129] Tómas Thorbjarnarson and Neil Yorke-Smith. “Optimal training of integer-valued neural networks with mixed integer programming”. In: *PLOS ONE* 18.2 (2023), pp. 1–17 (cit. on pp. 42–48, 58, 59).
- [130] Christian Tjandraatmadja, Ross Anderson, Joey Huchette, Will Ma, Krunal Kishor Patel, and Juan Pablo Vielma. “The convex relaxation barrier, revisited: Tightened single-neuron relaxations for neural network verification”. In: *Advances in Neural Information Processing Systems (NeurIPS)* 33 (2020), pp. 21675–21686 (cit. on p. 43).
- [131] Vincent Tjeng, Kai Y. Xiao, and Russ Tedrake. “Evaluating Robustness of Neural Networks with Mixed Integer Programming”. In: *International Conference on Learning Representations (ICLR)*. 2018 (cit. on p. 43).
- [132] Rodrigo Toro Icarte, León Illanes, Margarita P. Castro, Andre A. Cire, Sheila A. McIlraith, and J. Christopher Beck. “Training binarized neural networks using MIP and CP”. In: *International Conference on Principles and Practice of Constraint Programming*. Vol. 11802. Springer. 2019, pp. 401–417 (cit. on pp. 41, 43–48, 56, 58, 59).
- [133] Calvin Tsay, Jan Kronqvist, Alexander Thebelt, and Ruth Misener. “Partition-Based Formulations for Mixed-Integer Optimization of Trained ReLU Neural Networks”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2021, pp. 3068–3080 (cit. on p. 43).
- [134] Roberto Valente, Angela Testi, Elena Tànfani, Marco Fato, Ivan Porro, Maurizio Santo, Gregorio Santori, Giancarlo Torre, and Gianluca Ansaldo. “A model to prioritize access to elective surgery on the basis of clinical urgency and waiting time”. In: *BMC Health Services Research* 9 (2009) (cit. on pp. 74, 93).
- [135] Carla Van Riet and Erik Demeulemeester. “Trade-offs in operating room planning for electives and emergencies: A review”. In: *Operations Research for Health Care* 7 (2015), pp. 52–69 (cit. on pp. 68, 74, 75).
- [136] Joaquin Vanschoren. “Meta-learning”. In: *Automated machine learning*. Springer, 2019, pp. 35–61 (cit. on p. 41).

- [137] Sriram Venkataraman, Lawrence D. Fredendall, Kevin M. Taaffe, Nathan Huynh, and Gilbert Ritchie. “An empirical examination of surgeon experience, surgeon rating, and costs in perioperative services”. In: *Journal of Operations Management* 61 (2018), pp. 68–81 (cit. on p. 68).
- [138] Robert Vicari. *Simplex based Steiner tree instances yield large integrality gaps for the bidirected cut relaxation*. 2020. arXiv: 2002.07912 [cs.DS]. URL: <https://arxiv.org/abs/2002.07912> (cit. on pp. 8, 35).
- [139] Piotr Wais. “A review of Weibull functions in wind sector”. In: *Renewable and Sustainable Energy Reviews* 70 (2017), pp. 1099–1107 (cit. on p. 132).
- [140] K. Wang, L. Lozano, C. Cardonha, and D. Bergman. “Optimizing over an ensemble of trained neural networks”. In: *INFORMS Journal on Computing* 35.3 (2023), pp. 652–674 (cit. on p. 44).
- [141] Lien Wang, Erik Demeulemeester, Nancy Vansteenkiste, and Frank Rademakers. “On the use of partitioning for scheduling of surgeries in the inpatient surgical department”. In: *Health Care Management Science* 25(4) (2022), pp. 526–550 (cit. on pp. 68, 71, 72, 113).
- [142] Lien Wang, Erik Demeulemeester, Nancy Vansteenkiste, and Frank Rademakers. “Operating room planning and scheduling for outpatients and inpatients: A review and future research”. In: *Operations Research for Health Care* 31 (2021) (cit. on pp. 68, 69, 74, 94, 96).
- [143] H. Paul Williams. *Model building in mathematical programming*. John Wiley & Sons, 2013 (cit. on p. 50).
- [144] Han Xiao, Kashif Rasul, and Roland Vollgraf. “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms”. In: *arXiv preprint arXiv:1708.07747* (2017) (cit. on p. 55).
- [145] Yao Xiao and Reena Yoogalingam. “A simulation optimization approach for planning and scheduling in operating rooms for elective and urgent surgeries”. In: *Operations Research for Health Care* 35 (2022) (cit. on p. 70).
- [146] Huigen Ye, Hua Xu, Hongyan Wang, Chengming Wang, and Yu Jiang. “GNN&GBDT-Guided Fast Optimizing Framework for Large-scale Integer Programming”. In: *International Conference on Machine Learning (ICML)*. Vol. 202. Proceedings of Machine Learning Research. PMLR, 2023, pp. 39864–39878. URL: <https://proceedings.mlr.press/v202/ye23e.html> (cit. on p. 41).
- [147] H. Peyton Young. “Condorcet’s theory of voting”. In: *American Political Science Review* 82.4 (1988), pp. 1231–1244 (cit. on p. 44).

- [148] Xin Yu, Thiago Serra, Srikumar Ramalingam, and Shandian Zhe. “The combinatorial brain surgeon: Pruning weights that cancel one another in neural networks”. In: *International Conference on Machine Learning (ICML)*. 2022, pp. 25668–25683 (cit. on pp. 42, 43).
- [149] Shuwan Zhu, Wenjuan Fan, Shanlin Yang, Jun Pei, and Panos M. Pardalos. “Operating room planning and surgical case scheduling: a review of literature”. In: *Journal of Combinatorial Optimization* 37.3 (2019), pp. 757–805 (cit. on pp. 68, 76).
- [150] Ray Daniel Zimmerman, Carlos Edmundo Murillo-Sánchez, and Robert John Thomas. “MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education”. In: *IEEE Transactions on power systems* 26.1 (2010), pp. 12–19 (cit. on p. 132).

RINGRAZIAMENTI

Il primo immenso grazie va a chi sta leggendo queste righe dopo essersi sparato tutte le pagine precedenti: eroe vero. A te che stai leggendo solo i ringraziamenti, dopo aver letto il titolo e sfogliato qualche pagina in cerca di figure, dico: non ti biasimo, ho fatto la stessa cosa innumerevoli volte, e non smetterò di certo ora.

La restante parte dei ringraziamenti verrà effettuata in rigoroso ordine alfabetico, di certo come non si confà in simile occasione; in quanto autore, tuttavia, sento di potermi prendere tale scellerata libertà. Nota bene che i seguenti insieme di persone sono non vuoti e non necessariamente con intersezione vuota.

A tutti quelli che non ricadono in nessuna delle altre categorie.

L'ordine alfabetico si è rivelato fallimentare già alla prima riga. Pazienza. Ho creato questa categoria non solo perché chi legge potrebbe non riconoscersi pienamente in nessuna delle altre, ma anche e soprattutto perché sono un pavido e uno smemorato e ho paura di dimenticarmi dei pezzi. E dunque a te, collaboratore della mia vita, dico grazie.

Al mio supervisore. Puntare su di me, per lui, penso sia stato simile a puntare sul rosso in una corsa di cavalli. E dopo aver letto tutto ciò che ho scritto e ascoltato gran parte dei miei deliri, sarà lui a dire se la scommessa è risultata vincente o meno. Le persone assennate sanno però che, quando si scommette, è più per divertimento che per brama di vittoria, quindi spero almeno che si sia divertito. Per quanto mi riguarda, io non avrei potuto chiedere di meglio.

Alla mia famiglia. Assolutamente non banale avermi come figlio, fratello, nipote, cugino, e devo ringraziare chi questo fardello se lo è sempre sobbarcato. Spero di dimostrare a queste persone l'ottimo lavoro che stanno facendo, che hanno sempre fatto, e che, ne sono certo, continueranno a fare.

Alle persone con cui ho condiviso delle conferenze. Tra ventilconvettori con aria condizionata fissa sui 16°C, terremoti ferroviari, cene sociali alle 17, truffe alberghiere, addetti al ricevimento discutibili, pause caffè dai risvolti imprevedibili, piste "ciclabili", teli mare, mostri e gattini, troppe papas, locali jazz, presentazioni con persone che corrono, e noci di cocco si è anche trovato

il tempo di seguire delle conferenze. E niente sarebbe stato anche solo lontanamente divertente come lo è stato grazie alle persone con cui le ho condivise. Avete colorato i miei viaggi con tinte indelebili.

Ai miei amici di Lugano. Se mi avessero chiesto cosa avrei associato di più a Lugano tra teoria della complessità e calciobalilla post quiz di cultura generale, probabilmente avrei sbagliato risposta. E non solo per colpa mia, ci tengo a precisare. Oltre a ciò, devo dire che ho trovato molto più di quanto avrei mai sperato di trovare, e spero di aver lasciato alle incredibili persone che mi hanno accolto in quel luogo inospitale anche solo una minima parte di quello che loro hanno lasciato a me.

Ai miei amici di Pavia. A chi mi segue dalle medie fino a chi ho incontrato poche settimane fa, la stima che mi avete dimostrato mi ha sostenuto come poche altre cose hanno fatto in questi anni, ed è stata molto più importante di quanto possiate pensare. Un ringraziamento speciale per chi ha bazzicato il dimat con me. Qui mi è stato insegnato che le cose sono molto più di quello per cui si mostrano: quasi mai la pausa acqua serviva a riempire la borraccia così come la pausa pranzo a rifocillarsi, e non credo di ricordare una sola discussione di lavoro che abbia trattato solo di matematica. Sarebbe stato tanto triste fare la vita “di chi crede che la realtà sia quella che si vede”, e per fortuna le persone che ho incontrato sono state sufficientemente straordinarie da risparmiarmi questo supplizio.

Ai miei colleghi. A tutti quelli che hanno scritto, ragionato, o anche solo chiacchierato di matematica con me, dico grazie. Avete smontato in fretta le idee assurde che mi sono venute in mente, risparmiandomi un sacco di fatica nel farlo da solo; me ne avete suggerite altre, a dir poco brillanti; mi avete aiutato a finalizzare le poche sensate che ogni tanto ho partorito. Non potrei mai dire che preferisco mille volte lavorare con qualcuno anziché da solo, se quel qualcuno non fosse uno di voi.



UNIONE EUROPEA
Fondo Sociale Europeo



La borsa di dottorato è stata cofinanziata con risorse del
Programma Operativo Nazionale Ricerca e Innovazione 2014-2020, risorse FSE REACT-EU
Azione IV.4 “Dottorati e contratti di ricerca su tematiche dell’innovazione”
e Azione IV.5 “Dottorati su tematiche Green”.



CompOpt
UNIPV