Model-driven solution approaches for patient scheduling in admission and surgery management

Haichao Liu

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

université angers

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE ANGERS
COMUE UNIVERSITÉ BRETAGNE LOIRE

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Informatique*

Par
Haichao LIU

Model-driven solution approaches for patient scheduling in admission and surgery management

Thèse présentée et soutenue à Angers, le 7 Octobre 2024
Unité de recherche : Laboratoire d'Étude et de Recherche en Informatique d'Angers (LERIA)
Thèse N° :

Rapporteurs avant soutenance :

M. André ROSSI	Professeur à Université Paris Dauphine
Mme. Jingwen ZHANG	Professeur à Northwestern Polytechnical University

Composition du Jury :

Président :	M. Eric MONFROY	Professeur à Université d'Angers
Examinateurs :	M. Manuel CLERGUE	Professeur à École supérieure d'informatique électronique automatique
	M. Jin-Kao HAO	Professeur à Université d'Angers
	M. André ROSSI	Professeur à Université Paris Dauphine
	Mme. Yang WANG	Professeur à Northwestern Polytechnical University
	M. Qinghua WU	Professeur à Huazhong University of Science and Technology
	Mme. Jingwen ZHANG	Professeur à Northwestern Polytechnical University
Directeur de thèse :	M. Jin-Kao HAO	Professeur à Université d'Angers
Co-dir. de thèse :	Mme. Yang WANG	Professeur à Northwestern Polytechnical University

Acknowledgement

The past two years studying at the Université d'Angers have been an immensely enriching and fulfilling period in my academic journey. Foremost, I would like to express my sincere gratitude to my advisor Professor Jin-Kao Hao and co-advisor Professor Yang Wang for their guidance, encouragement, and support throughout my study. Their patience, vision, motivation, and immense knowledge have deeply inspired me. Working with them has been a great pleasure and a valuable experience. They have taught me the methodology to carry out the research, the way to think critically, and the way to write a scientific paper.

I would like to thank Professor Manuel Clergue and Professor Qinghua Wu in my thesis committee for providing me help during my study. I would also like to thank Professor André Rossi and Professor Jingwen Zhang who reviewed my thesis. I would like to thank Professor Eric Monfroy for being the president of my thesis committee.

My sincere thanks also go to the assistance of our technicians Jérôme Chalain, Jean-Mathieu Chantrein, Aurélien Simon, our nice secretary Emmanuelle Baudouin and all other lab members. Thanks to them, I have been able to focus on my research work and have a good working environment in the LERIA lab.

Last but not the least, I am extremely grateful to my family: my parents Yajuan Wang and Fuan Liu, my sister Rui Liu for their support and encouragement. Moreover, I am grateful to my friends Qing Du, Mingjie Li, Yuji Zou, Zhenyu Lei, Wei Yang, Lina Zang, Yuqi Zhao, He Zheng, Pengfei He for their friendship and help during my study in France.

This research has been financially supported by the National Natural Science Foundation of China (NSFC, No. 71971172, No. 72371200), the National Key Research and Development Project of China (No. 2023YFE0206200), the China Scholarship Council (CSC, Grant No. 202206290116) and the Innovation Foundation for Doctor Dissertation of Northwestern Polytechnical University (No. CX2022057).

Table of Contents

General Introduction

Context

Hospitals are under pressure from an increase in demand for inpatient care and high-quality patient care. However, it is challenging to meet patients' needs under limited hospital resources [1]. Among the healthcare services, hospital admission is generally the first step in receiving treatment, as emergency and surgical care is highly dependent on bed availability [2]. One key issue is to optimize the use of bed resources as much as possible, given that beds are a critical and limited resource in a hospital [3]. Following admission, surgical procedures are the main way to treat patients with various diseases. Efficient operating room (OR) scheduling is essential in a hospital since OR is the most resource-intensive cost and productive unit generating more than 40% of total revenues and expenses [4, 5]. Therefore, it is crucial to develop scheduling approaches that simultaneously improve OR efficiency and patient satisfaction. In this thesis, we focus on two critical problems in healthcare management, including the patient admission scheduling problem and the surgical case scheduling problem.

The patient admission scheduling (PAS) problem [6] consists of assigning patients to beds in specific departments on each day of their hospitalization while satisfying a number of hard constraints and as many soft constraints as possible. Over the past few years, this problem has received increasing attention in the literature. The PAS problem studied in the literature can be divided into static or dynamic versions. In the static PAS problem, only elective patients are considered, and all patient admission and discharge requirements are deterministic [7, 8, 9, 10, 11]. The dynamic PAS problem, also known as PASU (U for uncertainty), extends the static PAS problem by considering several real-world features, such as the presence of urgent and emergency patients whose arrival dates are uncertain, the possibility of delayed admissions and the uncertainty of length of stay (LOS) [12].

The surgical case scheduling (SCS) problem [13], which involves assigning surgeries to dates and ORs in a given time horizon, and determining the start time of each surgery, while taking into account various constraints, such as stochastic surgery duration, resource capacity, and other relevant factors. The SCS problem considers two types of surgeries:

elective surgeries, which are planned in advance and can be scheduled in a flexible manner, and emergency surgeries, which are unpredictable and need to be performed immediately. The stochastic surgery duration and the requirement of having ORs immediately for the treatment of unpredictable emergency surgeries, which are the main focus in the literature as noted in [13, 14], make the SCS problem a real challenge. Moreover, the critical issue in addressing the SCS problem is how to balance allocating OR resources between emergency surgeries and elective surgeries.

Objectives

This thesis focuses on building mathematical optimization models and proposing solution methods for the PAS problem and the SCS problem. The main objectives of this thesis are as follows.

— Reducing the size of the classical integer programming models of the standard PAS problem to improve the computational efficiency.

— Providing methodologies to help hospitals make better patient admissions and cope with uncertainty in length of stay.

— Balancing allocating OR resources between emergency and elective surgeries by integrated proactive and reactive strategies.

— Evaluating the performance of the proposed models and solution methods on benchmark instances in comparison with the state-of-the-art methods.

— Analyzing the ingredients of the proposed methods to get useful insights for elective patient scheduling problems.

Contributions

The main contributions of this thesis are summarized as follows.

— For the standard PAS problem, we present a two-stage optimization approach to solve the problem. Specifically, in the first stage, we propose two aggregated gender policy constraints and one aggregated patient transfer constraint to reduce the size of the classical integer programming model. Experimental results on the 13 benchmark instances in the literature indicate that our method can obtain new improved

solutions (new upper bounds) for 6 instances, including one proven optimal solution. This work has been published in *European Journal of Operational Research* [7].

— We propose a new stochastic PAS (SPAS) problem and build two-stage stochastic programming models, including a scenario-based model $SPAS_{SB}$ and its equivalent state-variable model $SPAS_{SV}$. Compared to the former, the latter model is significantly reduced and has a pseudo-polynomial number of variables and constraints. To solve the state-variable model efficiently, we propose a hybrid sample average approximation and state-variable (SAA-SV) approach. The *SAA-SV* method is capable of finding solutions with an average optimality gap of 1.73% for large instances reaching 500 patients and 3.3×10^{150} scenarios in 1 hour. This work has been submitted to *European Journal of Operational Research*.

— We study a surgical case scheduling problem in flexible operating rooms under uncertainty (SSFU). To solve the problem, we apply the proactive/reactive strategy, which decomposes the problem into two sub-problems: a proactive SSFU problem and a reactive SSFU problem. We build a two-stage stochastic programming model and a mixed integer programming model. Moreover, we implement three mechanisms — reserving capacity, Break-In-Moment, and buffer — to improve the robustness of the plan. Extensive experiments show the effectiveness of the proposed proactive/reactive strategy. This work has been submitted to *Production and Operations Management*.

— We propose an innovative three-phase simulation-optimization (TPSO) approach to solve the proactive SSFU problem to obtain a high-quality solution, where the result of dynamic rescheduling in the planning horizon is considered. Specifically, the problem is further decomposed into a surgery assignment problem and multiple surgery sequencing problems. Moreover, a discrete-event simulation algorithm is proposed to evaluate the quality of the solution. We also propose a set of feedback constraints to guide the search process. Extensive experiments show the effectiveness of the proposed TPSO approach. One of the related works has been submitted to *Production and Operations Management*, and one has been accepted to *Journal of Systems Engineering*.

Organization

The thesis is organized in the following way.

— In the first chapter, we introduce the healthcare scheduling problems first. Then we present the related works on the patient scheduling problem, including the static variants and the dynamic variants. Furthermore, we review the literature based on the proactive/reactive/integrated proactive and reactive strategies adopted to solve the surgical case scheduling problem. Finally, the related approaches are introduced.

— In the second chapter, we present a study on the standard PAS problem. The two-stage optimization approach, integer programming models, and aggregation constraints are presented in detail. Extensive experiments on well-known benchmark instances show that the method competes favorably with the state-of-the-art methods in terms of solution quality.

— In the third chapter, we present a study on the SPAS problem. The scenario-based model $SPAS_{SB}$ and the state-variable model $SPAS_{SV}$ are introduced in detail. Then, the general SAA-SV approach is presented to solve the state-variable model. Extensive experiments indicate the effectiveness of the proposed models and SAA-SV approach.

— In the fourth chapter, we present a study on the SSFU problem. We present the proactive SSFU model and the reactive SSFU model, which are built based on the proactive/reactive strategy. Extensive experiments show the effectiveness of the proposed proactive/reactive strategy.

— In the fifth chapter, we present the TPSO approach to solve the proactive SSFU problem. Extensive experiments show the effectiveness of the proposed TPSO approach.

— In the last chapter, we summarize the contributions of this thesis and provide some perspectives for future research.

During my PhD, apart from the works mentioned above, I also worked on a multi-day task assignment problem, which introduced several features of practical relevance to the widely-studied generalized assignment problem. To solve this problem, an innovative three-phase matheuristic algorithm was proposed, which first employs a construction phase to quickly produce a reasonable quality solution and then alternates between an intensification phase to reach local optima and a diversification phase to drive the search into new regions. For the above phases, a new decomposition-based construction heuristic and solver-based heuristic strategies are developed. This work has been published in *Computers & Operations Research* [15].

Part I

Introduction

Chapter 1

Introduction

In this chapter, we present a brief overview of the healthcare scheduling problems. We present the related works on the patient admission scheduling (PAS) problem, including the static variants and the dynamic variants. In addition, the related heuristic and exact algorithms for the PAS problem are summarized. Moreover, we review the surgical case scheduling problem, including the proactive and reactive strategies, and the integrated proactive and reactive strategies. Finally, we discuss the related approaches used in this thesis, including constraint aggregation techniques, modeling decision problems under uncertainty, and simulation optimization approaches.

1.1 Healthcare scheduling problems

A healthcare system is a structured setup that provides preventive measures, medical services, and treatment to patients [16]. The demand for high-quality health services continues to increase year after year, while hospitals encounter more and more difficulties in terms of limited medical resources [17]. As per the World Health Statistics for 2023, the rise in life expectancy is contributing to a growing elderly population. The limited resources and high cost have captured the attention of many researchers. Healthcare scheduling is a complex and challenging task due to the significant constraints, the preference constraints of staffs and patients, the dynamic and unpredictable nature of the healthcare system. Efficient healthcare scheduling ensures the timely allocation of resources and treatment, leading to improved resource utilization and patient satisfaction [16]. Therefore, the advancement of patient scheduling techniques plays a vital role in the improvement of healthcare services.

In the literature, many healthcare scheduling problems have been studied, specifically emphasizing on patient admission scheduling [6], surgical case scheduling [13], and nurse rostering [18]. The patient admission scheduling (PAS) problem, first introduced by [6], consists of assigning patients to beds in specific departments on each day of their hospitalization while satisfying a number of hard constraints and as many soft constraints as possible. The surgical case scheduling (SCS) problem [13], also known as the surgery scheduling problem and the operating room scheduling problem, consists of assigning surgeries to dates and ORs in a given time horizon, and determining the start time of each surgery, while taking into account various constraints, such as stochastic surgery duration, resource capacity, and so forth. The nurse rostering problem (NRP) [19] is a type of staff scheduling problem that is set through the allocation of a group of different skilled nurses to various types of shifts over a predefined scheduling time. Apart from the above problems, there are many problems receiving less attention from researchers, such as scheduling physicians [20], home healthcare scheduling [21], telemedicine scheduling [22]. For more details on healthcare scheduling problems, readers are referred to the survey by Abdalkareem et al. [23]. Most of these problems have been proven to be NP-hard, which means that they are computationally challenging. The importance and growth in using optimization methods revealed very effective results when applied to these problems. However, it is still possible to improve the outcomes generated by present studies.

1.2 Patient admission scheduling problem

The PAS problem has undergone multiple extensions over the years and can be classified into static and dynamic variants. The PAS problem is known to be NP-hard [24]. As a result, solving the problem is computationally challenging. In the following sections, we provide a comprehensive review of the solution approaches proposed in the literature for both the static and dynamic PAS problems, including heuristic and exact methods.

1.2.1 Static patient admission scheduling problem

In the static variants, only elective patients are considered, and all patient admission and discharge requirements are assumed to be deterministic. Additionally, patient admissions in these static variants are scheduled in advance. The primary difference in studies that focus on static variants lies in their treatment of soft constraints that can be violated at the cost of incurring a penalty. These constraints include gender policy, age policy, mandatory equipment, single-room requirements, and patient transfer. In the original PAS problem, only the patient transfer constraint was described as a soft constraint, while the first 4 constraints were described as hard constraints, which are not allowed to be violated. However, Demmeester et al. [25] also consider these 4 hard constraints as soft. As a result, only Range et al. [10], Hammouri & Alweshah [26], and Guido et al. [9] treated the first 4 constraints or part of them as hard constraints, while others considered them as soft constraints or ignored some of them. Moreover, except for Ceschia & Schaerf [27], Range et al. [10], Turhan & Bilgen [28], and Bastos et al. [8], most studies treated the patient transfer constraint as a hard constraint. Considering patient transfer as a hard constraint can simplify the problem by narrowing the search space, but it may also result in only finding sub-optimal solutions for those variants that consider patient transfer as a soft constraint.

Heuristic algorithms for solving the PAS problem aim to find good enough solutions in a reasonable time. Existing heuristic algorithms are based either on single-trajectory search or population-based search. Among the single-trajectory search, Demmeester et al. [25] proposed an IP model to assign patients to rooms while allowing violations of some soft constraints. They applied a Hybrid Tabu Search (H-TS) algorithm blended with a token-ring and a variable neighborhood descent procedure. They generated and made publicly available a set of 13 realistic benchmark instances for the PAS problem, which were largely adopted in the literature. Ceschia & Schaerf [27] proposed an IP model which considers

all soft constraints to compute various lower bounds. Moreover, a simulated annealing (SA) algorithm was developed, which significantly improved the previous upper bounds.

Bilgin et al. [29] proposed a hyper-heuristic (H-H) involving multiple heuristic selection and move acceptance criteria. Range et al. [10] proposed a column generation-based (CG) heuristic, which decomposes the PAS problem into a set-partitioning problem as the master problem and a set of room scheduling problems as the pricing problem. Kifah & Abdullah [30] proposed an adaptive non-linear great deluge (ANLGD) algorithm, which accepts worse solutions of satisfying a given threshold. Turhan & Bilgen [28] utilized the IP model developed by Ceschia & Schaerf [27] and proposed two mixed integer programming-based heuristics, namely Fix-and-Relax (F&R) and Fix-and-Optimize (F&O), to obtain solutions with optimality gaps of 5-15% in less than three minutes. Bolaji et al. [31] introduced a late acceptance hill climbing (LAHC) algorithm, which first generates an initial feasible solution and then iteratively improves the solution by applying a local search procedure. Guido et al. [9] developed three IP models and proposed a matheuristic FiNeMath, combining the F&O heuristic, neighborhood search, and IP solvers. Their method produced good results for all benchmark instances presented in [25].

For population-based methods, Hammouri & Alrifal [32] first reported the biogeography based optimization (BBO) algorithm for the static PAS problem, which failed to improve the state-of-the-art. Later on, to improve the performance of the algorithm, the authors proposed a BBO algorithm with guided bed selection mechanism (BBO-GBS) [26], and a modified BBO algorithm with guided bed selection mechanism (MBBO-GBS) [33]. Moreover, several researchers have attempted to improve the performance of population-based algorithms for tackling the static PAS problem, including Harmony search (HS) algorithm [34], artificial bee colony (ABC) algorithm [35], discrete flower pollination (DFP) algorithm [36]. However, these algorithms could not produce competitive results on the benchmark instances.

In addition to the heuristic algorithms reviewed previously, Bastos et al. [8] studied an exact method to solve the static PAS problem. To the best of our knowledge, this is the only existing exact algorithm for the static PAS problem. The method was based on a new mathematical model, which incorporated all restrictions from the original model of Demeester et al. [25], and applied a warm start (WS) approach to solve it with the maximum running time set to 24 hours. They reported new best upper bounds for 9 out of the 13 benchmark instances introduced in [25]. Note that while Range et al. [10], Turhan & Bilgen [28], and Guido et al. [9] incorporated MIP formulations into heuristic

methods, these methods only improved the bounds of the optimal solution and failed to find optimal solutions.

1.2.2 Dynamic patient admission scheduling problem

The dynamic variants (DPAS), also known as the PASU (U for uncertainty), extend the static PAS problem by considering several real-world features, such as the presence of urgent and emergency patients whose arrival dates are uncertain, uncertainty in the length of stay, and the possibility of delayed admissions [37, 38, 39]. The above uncertainty information is gradually revealed on a day-to-day basis. Thus, the DPAS problem is solved through the use of daily rescheduling. Similar to the static variants, the main difference among the dynamic variants is that some soft constraints (e.g., age policy, mandatory equipment, department specialism) are considered as hard constraints. Moreover, in order to make the problem suitable for practical applications, some studies considered more realistic constraints, such as constraints related to operating room scheduling [37, 38].

Only a few studies addressed the dynamic PAS problem. Ceschia & Schaerf [27] first introduced a dynamic case of the PAS problem in which admission and discharge dates are uncertain. They adapted their SA algorithm to solve this problem. Later, Ceschia & Schaerf [40] formally introduced the dynamic PAS problem to account for uncertain length of stay, admission delays, and non-elective patients. This variant was solved using the SA algorithm and subsequently extended to incorporate operating room resources [37]. Lusby et al. [41] developed an adaptive large neighborhood search (ALNS) procedure combined with the SA framework. Their method showed superior results compared to the method suggested by Ceschia & Schaerf [40] in most cases. Recently, Guido et al. [39] proposed an optimization model that plans patient admissions and patient stays considering fluctuations and does not allow overcrowded rooms, as typically required in real-world cases. They proposed a matheurisitic FiNeMath-PASU, which is based on the FiNeMath [9].

For the exact methods, Vancroonenburge et al. [42] developed two IP models and considered the impact of emergency patients and patient length of stay estimates. Zhu et al. [38] studied the compatibility of short-term and long-term objectives in the dynamic PAS problem and developed multiple MIP formulations, which were solved by MIP solver. Their approach was shown to be significantly better than the available results for 26 out of 30 benchmark instances introduced in [37]. Table 1.1 provides a summary of the existing research on the PAS problem along with the problem type, problem constraints, and

solution approach. We indicate both types of PAS problems—static or dynamic— and specify which constraints are considered hard or soft in each study. The symbols "✓" and "-" are used to respectively indicate the problem type and absence of the optimization model.

Table 1.1 – Summary of the PAS research

Reviewed Literature	Problem type		Problem constraints		Solution approach	
	Static	Dynamic	Hard	Soft	Optimization model	Algorithm
Heuristic methods						
Demeester et al. (2010) [25]	✓		1-8	(5-8)*, 9-13	IP	H-TS
Ceschia & Schaerf (2011) [27]	✓	✓	1-4, 14	5-13	IP	SA
Ceschia & Schaerf (2012) [40]		✓	1, 2-3, 6-7, 10, 14	5, 9, 12-14	IP	SA
Bilgin et al. (2012) [29]	✓		1-4, 13**	5, 7-9, 11-12	MINLP	H-H
Hammouri & Alrifal (2014) [32]	✓		1-4, 13**	5-12	-	BBO
Range et al. (2014) [10]	✓		1-7	9-13	IP	CG
Kifah & Abdullah (2015) [30]	✓		1-4, 13**	5-12	MINLP	ANLGD
Ceschia & Schaerf (2016) [37]		✓	1, 3-4, 14	5-7, 9-14	-	SA
Lusby et al. (2016) [41]		✓	1, 3-4, 6-7, 10, 14	5, 9, 12-14	MIP	ALNS
Hammouri & Alweshah (2017) [26]	✓		1-5, 13**	7-12	-	BBO-GBS
Turhan & Bilgen (2017) [28]	✓		1-4	5-13	IP	F&R, F&O
Abu Doush et al. (2018) [34]	✓		1-4, 13	5-7, 9-12	MINLP	HS
Guido et al. (2018) [9]	✓		1-7, 13	5(partly soft), 6-7, 9-12	MIP	FiNeMath
Bolaji et al. (2018) [31]	✓		1-4, 13**	5, 7-12	MINLP	LAHC
Bolaji et al. (2022) [35]	✓		1-4, 13**	5, 7-12	MINLP	ABC
Hammouri (2022) [33]	✓		1-4, 13	5-7, 9-10, 12	MINLP	MBBO-GBS
Abdalkareem et al. (2022) [36]	✓		1-4, 13**	5-7, 9-10, 12	MINLP	DFP
Guido (2023) [39]		✓	1, 3-4, 6-7, 14	5, 9-10, 12-14	MIP	FiNeMath-PASU
Exact methods						
Vancroonenburg et al. (2016) [42]		✓	1, 2-3, 6-7, 10, 14	5, 9, 12-14	IP	MIP solver
Bastos et al. (2019) [8]	✓		1-4	5-13	MIP	WS
Zhu et al. (2019) [38]		✓	1, 3-4, 14	5-7, 9-14	MIP	MIP solver
This study	✓		1-4	5-13	IP	WS, CA

Problem constraints: 1 - complete assignment; 2 - unchangeable date; 3 - continuous schedule; 4 - non-overlapping allocation; 5 - gender policy; 6 - age policy; 7 - mandatory equipment; 8 - single room requirement; 9 - room type preference; 10 - departmental specialism; 11 - room specialism priority; 12 - perferred room properties; 13 - patient transfer; 14 - others
* The constraints 5-8 are also incorporated into the objective function as penalties in their H-TS algorithm.
** Although the author described the patient transfer constraint as 'soft', however, they do not provide mechanisms for transferring patients in their method. Thus, the patient transfer constraint would never be violated in their method.

1.3 surgical case scheduling problem

We briefly review studies that address the SCS problem considering the emergency surgeries and adopt flexible policy, which are most related to our work. These studies

can be classified into two main categories based on their ideas of addressing uncertainty: proactive strategy and reactive strategy [43, 44]. Both strategies for advance scheduling and allocation scheduling are related to our work. Thus, we will review these related studies based on the strategies they adopt in the following section. For a complete review of the SCS problem, readers are referred to [45, 46, 13, 47, 48, 49].

1.3.1 Proactive scheduling problem

The proactive strategy aims at proactively making some ex-ante preparations (i.e., reserving capacity, BIMs, buffers) in initial schedules to tackle the challenges posed by stochastic surgery durations and emergency arrivals. Reserving capacity is a common mechanism used to tackle the uncertainty of emergency arrival in advance scheduling problems. Thus, the uncertainties of elective surgery durations and emergency demands are the two primary factors of uncertainty in advance scheduling problems. The majority of studies develop stochastic or robust programming models, representing the uncertainty of elective surgery durations and emergency demands as stochastic variables, and propose different methods to solve them. Lamiri et al. [50] assumed that the elective surgery durations are deterministic, and the distribution of emergency demands is known. They developed a stochastic programming model and used the sample average approximation (SAA) method to solve it. Molina-Pariente et al. [51] considered that both the elective surgery durations and emergency demands are stochastic and assumed their distributions are known. To solve this problem, they developed a stochastic programming model and proposed a monte carlo optimization method, which combines the iterative greedy local search method with SAA. Also, Tang and Wang [52] assumed that only the lower and upper bound of elective surgery durations and emergency demands are known. They built a robust optimization model and proposed an implementer-adversary algorithm to solve it. Recently, Miao and Wang [53] addressed the need for distributed surgical scheduling across multiple hospitals and multiple days. They only considered the advance scheduling and stochastic surgery durations and emergency demands. They formulated the problem as a two-stage stochastic programming model and proposed an integer decomposition algorithm.

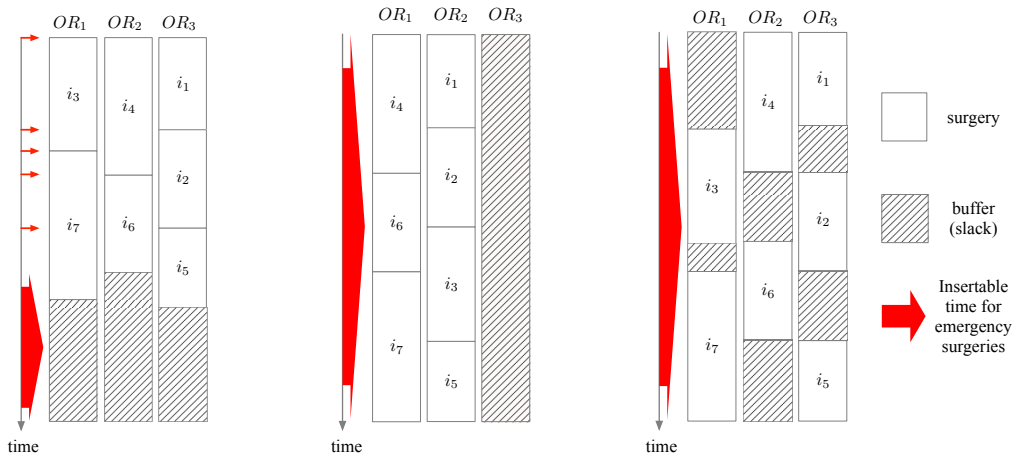
In the allocation scheduling problem, a fundamental principle is that it is typically not possible to interrupt an ongoing surgery to accommodate an emergency surgery. Thus, researchers proposed BIM and buffer mechanisms by optimizing the surgeries and the slack time distribution in one day to reduce waiting time for randomly arrived emergency

surgical insertions. Figure 1.1 presents the three combinations of BIM and buffer proposed in the literature, including BIMs only, BIMs + extra OR, and BIMs + buffers. The example shows seven elective surgeries scheduled in three ORs. Points or time intervals, when emergency surgeries can enter an OR, are marked with a red arrow. As the figure shows, the difference between the three combinations is the distribution of slack time. Firstly, BIMs only is the one [54, 55, 56, 57, 58] aim at. They schedule all elective surgeries without any buffer between but in all ORs such that the slack time is distributed over all ORs at the end of the day/block. Emergency surgeries can enter the schedule at the end of any elective surgery. Secondly, BIMs + extra OR is the one proposed in [59]. All surgeries are scheduled in the first two ORs such that the break-in-intervals (BIIs), i.e., the time between two consecutive BIMs, are minimal, and no buffer is included. The slack time is distributed over the third OR. Actually, this type is a special case of hybrid policy, where one OR is reserved for emergency surgeries, and the other ORs are available for both elective and emergency surgeries. Thirdly, BIMs + buffers is the one [59, 60, 43, 61] aim at. They not only introduced BIMs at the end of any scheduled job, which can be used by emergency surgeries to access resource capacity, but further considered the scheduling of buffers as a possible time interval for emergencies that seek to enter the schedule. In addition, Xiao and Yoogalingam [62, 63], Aissaoui et al. [44] only considered the scheduling of buffers without BIMs to handle the uncertainty of surgery durations. In order to select the best combination, we developed a novel programming model that can realize all the above types of combinations by setting the suitable weights of the objective function.

Despite the above studies, only some studies investigate the integration of advance scheduling and allocation scheduling problems, simultaneously. Tsai et al. [64] investigated the integration of advance scheduling and allocation scheduling problem considering a single OR. They applied the reserving capacity mechanism to tackle the uncertainty of emergency demands, developed a two-stage mixed integer model, and proposed a simulated optimization algorithm to solve it.

1.3.2 Reactive scheduling problem

The reactive strategy involves adjusting schedules in real-time based on actual surgery durations and emergency arrivals. Possible decisions include delaying, canceling, or rescheduling elective surgeries while inserting emergency surgeries into the schedule. Erdem et al. [65] proposed a genetic algorithm to reschedule elective and emergency surgeries with



(a) BIMs only **(b)** BIMs+ex. **(c)** BIMs+buffers

Figure 1.1 – Three combinations of BIMs and buffers (slack)

an option where an emergency surgery can be turned away. They developed dynamic reactive scheduling methods to reschedule elective and arrived emergency surgeries, considering the future impact of decisions. Bargetto et al. [66] divided a day into multiple decision stages and proposed a dynamic model to determine whether to insert a remaining elective surgery or an arrived emergency surgery at each decision stage. Later, Silva and De Souza [67] considered more resource constraints and proposed an approximate algorithm to dynamically reschedule elective and emergency surgeries at each decision stage. Recently, Wang et al. [68] considered that such decisions in the reactive strategy can significantly reduce the satisfaction of scheduled patients. They studied the surgery rescheduling problem considering the preferences of three involved participants. They built a multi-objective optimization model and proposed a hybrid particle swarm optimization (HPSO) algorithm to solve it.

1.3.3 Integrated proactive and reactive scheduling problem

In recent years, integrated approaches of proactive and reactive strategy are proposed for coping with surgical case scheduling problem with uncertainty. Jung et al. [61] studied both advance scheduling and allocation scheduling problems while not considering the uncertainty of surgery durations and emergency demands. They adopt the BIMs+buffers mechanism, propose a mixed integer programming (MIP) model, and develop two heuristics to generate initial schedules. Moreover, they proposed a heuristic to reschedule elective

and emergency surgeries in real time. Different from Jung et al. [61], Miao and Wang [53] only studied the allocation scheduling problem under emergency demands and arrival uncertainty. For reactive strategy, they also proposed a MIP model to update the schedule on surgery day. Later, Wang et al. [60] considered more resource constraints, but did not consider the uncertainty of surgery duration and emergency demand. Eshghali et al. [69] also considered more downstream resources and the uncertainty of surgery durations and emergency demands. They developed hierarchical weekly, daily, and rescheduling models. They used the machine learning method to predict the surgery durations and emergency arrivals, and proposed genetic algorithm and particle swarm optimization to solve the models. Moreover, Duma and Aringhieri [54] investigated both advance scheduling and allocation scheduling problems while did not consider the surgery duration uncertainty. In Table 1.2, we summarize the main characteristics of the related papers, i.e., resources, uncertainty, scheduling, mechanisms, objective, and methodology. We detail if there is a solved advance scheduling or allocation scheduling in the scheduling column. We also detail if there is used reserving capacity, BIMs, or buffers in the mechanisms column. The symbols "✓" are used to respectively indicate the scheduling type and mechanisms.

1.4 Related approaches

In this section, we briefly review the related approaches that we use in this thesis, including constraint aggregation techniques, modeling decision problems under uncertainty, and simulation-based approaches.

1.4.1 Constraint aggregation

Using constraint aggregation (CA) can reduce the number of constraints of the optimization model, thereby simplifying its formulation and reducing its computational complexity. Specifically, CA involves replacing original constraints with a set of aggregated constraints, which are linear combinations of the original constraints by multipliers [72]. Note that the aggregated constraints are a relaxation of the original constraints, which means that the solution space of the original constraints is a subset of the solution space of the aggregated constraints. Choices of the multipliers directly affect the strength of the aggregated constraints. In addition, CA can suffer from poor performance when the aggregated constraints have very large coefficients, either in scale [73] or in numerical

Table 1.2 – A summary of papers related to the SCS problem.

Reviewed literature	Resources	Uncertainty	Scheduling		Mechanisms			Objective	Methodology
			Advance	Allocation	Reserving capacity	BIMs	Buffers		
Proactive strategy									
Lamiri et al. (2008) [50]	1	2	✓		✓			2,6	1,4
Tang and Wang (2015) [52]	1	2	✓		✓			6,11	1
Molina-Pariente et al. (2018) [51]	1,2	1,2	✓		✓			2,11	1,3,4
van Essen et al. (2012) [58]	1	3		✓		✓		9	1,2,3
Wang et al. (2014) [70]	1	2		✓	✓			1,2	1,2
Freeman et al. (2016) [71]	1	1,3		✓		✓		2,6,7	1,2
Latorre-Núñez et al. (2016) [57]	1,4	3		✓		✓		4	1,3
Vandenberghe et al. (2019) [56]	1	1,3		✓		✓		7	1,2
Xiao and Yoogalingam (2021) [63]	1	3		✓	✓		✓	2,3,7	1,4
Tsai et al. (2021) [64]	1	1,2,3	✓	✓	✓			2,6,7	1,4
Xiao and Yoogalingam (2022) [62]	1	3		✓			✓	2,3,7	1,2,4
Schulz and Fliedner (2023) [59]	1	3		✓		✓	✓	3,9,11	1,2
Miao and Wang (2023) [53]	1	1,2	✓		✓			1,6	1,4
Reactive strategy									
Erdem et al. (2012) [65]	1,2,4			✓				2,6,11	1,3
Bargetto et al. (2019) [66]	1	1,2,3		✓				2,11	1,2
Silva and De Souza (2020) [67]	1,2	1,2,3		✓				6,8	1,2
Wang et al. (2021) [68]	1,2,4,6,8,9			✓				11	1,3
Integrated proactive and reactive strategies									
Miao and Wang (2023) [53]	1	2		✓		✓	✓	2,6,7	1,2,4
Jung et al. (2019) [61]	1	3	✓	✓		✓	✓	1,2,3	1,2
Duma and Aringhieri (2019) [54]	1	3	✓	✓	✓	✓		9	2
Wang et al. (2023) [60]	1,6	3	✓	✓		✓	✓	2,3,6	1,3
Eshghali et al. (2024) [69]	1,3,4,10,11	2,3	✓	✓	✓			2,6	1,3
This study	1	1,2,3	✓	✓	✓	✓	✓	2,3,6,7,8	1,2,4

Resources: 1-ORs; 2-surgerons; 3-beds; 4-PACU; 5-Equipments; 6-anaesthesia; 7-surgery day; 8-nurses;9-PHU;10-ICU;11-CCU
Uncertainty: 1-elective surgery duration; 2-emergency surgery duration (demand); 3-arrival time of emergency; 4-Length of stay in ward; 5-recovery durations
Objective: OR-related: 1-fixed OR cost; 2-overtime cost; 3-idle time cost; 4-makespan;5-others;
Patient-related: 6-cost of performing or postponing; 7-waiting time; 8-cancellation; 9-the intervals between BIMs; 10-rescheduling impact on elective surgeries;11-others
Methodology: 1-mathematical programming; 2-heuristics; 3-metaheuristics; 4-simulation

values [74]. Thus, a crucial question of CA is how to determine the multipliers of the aggregated constraints to produce the strongest possible constraints. In this regard, researchers have proposed multiple methods such as aggregation of diophantine equations, irrational multipliers method, maximum entropy method, P-norm method, etc [75].

It is worth noting that the above CA technique, also known as static constraint aggregation (SCA) [76, 77, 78, 72], aggregates constraints before the optimization process. In contrast to SCA, dynamic constraint aggregation (DCA) [79, 10, 80, 81], in which aggregated constraints contain a subset of solutions to the original constraints, dynamically aggregates constraints during the solution process to obtain the optimal solutions. Moreover, DCA is always implemented within a column generation algorithm to solve a large set partitioning problem. CA has been applied with success to a variety of optimization problems, including multicommodity transportation [82], wing aero-structural optimization [83], integrated airline crew scheduling [84], and set partitioning [85, 81]. For a comprehensive survey of CA in optimization, see [75, 86, 78].

1.4.2 Modeling decision problems under uncertainty

For handling problems under uncertainty, stochastic programming (SP) [87, 88, 89, 90], and distributionally robust optimization (DRO) [91, 92, 93], are representative modeling approaches. Scenario-based stochastic programming uses a set of scenarios to characterize the LOS, and each scenario represents a possible realization. For this approach, the number of scenarios increases exponentially with the number of patients, given that the length of stay (LOS) of all patients is mutually independent. It is quite difficult to solve such a scenario-based model, given that each scenario corresponds to specific variables and constraints. Sample average approximation (SAA) [94, 95, 96, 97] is a widely-used approach for dealing with a large number of scenarios, which uses a subset of scenarios to avoid creating a large number of variables and constraints. However, SAA cannot guarantee the optimality of the obtained solution because it uses only a limited number of samples rather than all the scenarios. Therefore, how to deal with an exponential number of scenarios to obtain the optimal solution is quite challenging when solving the stochastic programming problem.

Recently, Doulabi et al. [98] proposed a state-variable modeling approach for a class of two-stage stochastic programming problems, which is capable of modeling problems with an exponential number of scenarios without any need for sampling. They verified the state-variable modeling approach on project scheduling and operating room allocation problems. Subsequently, different from assigning surgeries to operating rooms available on a single day, Hashemi Doulabi and Khalilpourazari [99] assigned surgeries to operating rooms over a week and considered the deadlines of surgeries. They proposed a state-variable model and enhanced it by reducing the number of variables and constraints. The above studies demonstrated the effectiveness of the state-variable modeling approach.

1.4.3 Simulation optimization

Simulation optimization (SO) refers to the optimization of an objective function subject to constraints, both of which can be evaluated through a stochastic simulation [100]. As opposed to mathematical programming, SO does not assume that a closed form expression is available, and one needs to repeatedly estimate the objective function via simulation. Simulation optimization, like stochastic programming, also attempts to optimize under uncertainty. However, stochastic programming differs in that it makes heavy use of the model structure itself [101]. From the method-driven perspective of simulation

optimization, there are five modes of the integration approach: (1) simulation-supported optimization, (2) simulation-supported optimization, (3) simulation optimization iteration, (4) simulation-based optimization, and (5) optimization-embedded simulation. The detailed description of these five modes can be found in the work of Zhou et al. [102]. In this thesis, we focus on the simulation optimization iteration, where the simulation and optimization from a loop and are executed iteratively to update the parameters of each other.

It is worth noting that the simulation is often implemented by the discrete-event simulation (DES), which can be used to model many real-world systems, such as queues, operations, and networks. Using DES to simulate a system usually involves switching or jumping from one state to another at discrete points in time as events occur. Moreover, the occurrence of events is modeled using probability distributions to model the randomness involved. Several applications of SO have been addressed in the literature, such as the nurse scheduling [103], healthcare [104], breast cancer epidemiology [105], power system [106]. For more details about simulation optimization, see [100, 102]

1.5 Chapter conclusion

In this chapter, we presented a brief overview of the healthcare scheduling problems. The static and dynamic versions of the PAS problem are considered, and the related heuristic and exact algorithms are summarized. We also reviewed the SCS problem, including the proactive and reactive strategies, and the integrated proactive and reactive strategies. Finally, the approaches used in this thesis were also discussed, including constraint aggregation techniques, modeling decision problems under uncertainty, and simulation optimization approaches.

Part II

Contributions

Chapter 2

Solving the patient admission scheduling problem using constraint aggregation

---

In this chapter, we consider the widely studied variant of the PAS problem that has the maximum number of soft constraints, and focus on how to reduce the size of IP formulations of the PAS problem to improve the solving efficiency. We employ a two-stage optimization method where the first stage builds reduced models by constraint aggregation to improve the typical formulation of the PAS problem. Experimental results on the 13 benchmark instances in the literature indicate that our method can obtain new improved solutions (new upper bounds) for 6 instances, including one proven optimal solution. For the 5 other instances whose optimal solutions are known, our approach can reach these known optimal solutions in a shorter computation time compared to the existing methods. In addition, we apply our method to the original PAS problem, which has the maximum number of hard constraints, and perform computational experiments on the same 13 benchmark instances. Our method yields 5 new best solutions and proves optimality for 6 instances. The content of this chapter is based on an article published in *European Journal of Operational Research*.

2.1 Introduction

Various optimization approaches, both heuristic and exact methods, have been proposed to address the PAS problem. The best results on most benchmark instances have been achieved using exact mathematical programming techniques, as reported in [8]. Nevertheless, the MIP model proposed in [8] becomes too large to be solved to optimality in a reasonable time for large instances. The integer programming (IP) model of [27] is more compact, but it is still too large to be solved to optimality on most large benchmark instances. A promising way to improve the efficiency of the solution process is to reduce the size of this model. Constraint aggregation (CA) can help reduce the number of constraints of the optimization model, which is a widely used technique in mathematical programming [76, 80]. Based on this idea, we focus on how better formulations – in this case, through reducing the IP model of [27] by aggregating constraints – can help further improve the efficiency in solving the IP model of the PAS problem. The contribution of this work is as follows:

(1) We propose two aggregated gender policy constraints and one aggregated patient transfer constraint to reduce the size of the IP model of [27], and evaluate the effectiveness of the proposed aggregated constraints through computational experiments.

(2) We apply a two-stage optimization approach using the reduced IP models to obtain optimal solutions. Specifically, for the standard PAS problem, we generate new best solutions for 6 out of the 13 benchmark instances commonly used in the literature, including one with proven optimality. Moreover, we prove the optimality of the solutions for the remaining 5 instances in a short time. Additionally, for the original PAS problem, using the same 13 benchmark instances, we obtain 5 new best solutions, 6 new best lower bounds, and proven optimality of solutions for 6 instances.

Next section presents the definition of the standard PAS problem and the mathematical model. Section 2.3 describes our solution method. Section 2.4 presents computational results of our IP formulations and comparisons with state-of-the-art results. In addition, section 2.4 also describes our solution method for the original PAS problem and reports the computational results. Section 2.5 draws conclusions.

2.2 Problem description and mathematical model

In this section, we provide a detailed description of the standard static PAS problem and its mathematical model.

2.2.1 Problem description

The PAS problem [25] aims to assign patients to a set of beds during patients' hospitalizations within a given planning horizon, where the preference and the requirement of each patient are assumed to be known in advance. Specifically, each patient p has an admission date AD_p when a room is assigned to this patient and a discharge date DD_p when this patient is released from the medical treatment. The *Length of Stay* (LOS) of each patient is the duration between the admission and the discharge dates, which is expressed in nights. Patients who stay at least one night, termed *elective patient*, are eligible to be scheduled. Patients pursue medical treatments during their hospitalization, termed *specialisms*. Most of the patients need one single specialism for their entire treatment. Only a small number of patients need more than one specialism, termed *multi-spec*. Each patient is assigned to a *bed* and each *bed* belongs to a *room*. One of the most important features of the room is the *gender policy*. Rooms that require patients to be of the same gender enforce policy M (only Male) or policy F (only Female). In contrast, rooms where both genders are allowed enforce policy N (mixed gender) or policy D (on any given night, only patients of the same gender are allowed, and the gender is defined by the first patient to be scheduled in that room). There are three types of room capacity: single (one bed), twin (two beds) and ward (four beds), and each patient has a preference for a certain type of room, termed *room preference*. Each room has a different available equipment, such as oxygen and telemetry, termed *room properties*. Patients may require or prefer to be allocated to a room with the specific equipment depending on their treatment. Each room belongs to a *department* and each department is correlated with the specialisms they offer. Moreover, each department and room has its own priority degree for those specialisms. Patients must be treated at the departments where the specialism they need is offered. Each department has an *age policy* which imposes a maximum or minimum age limit for acceptance. Patients can change rooms during their hospitalization, termed *transfers*.

Based on the above problem definition, a solution is feasible if all patients are assigned to a bed such that no hard constraint of types HC1 - HC4, given below, is violated.

HC1: During the planning period, each patient must be assigned to a room.

HC2: Admission and discharge days for each patient can not be adjusted.

HC3: Patient LOS is continuous, and a patient is scheduled until his/her discharge date.

HC4: Beds allocated to patients should not overlap on any given night.

The quality of a feasible solution depends on the satisfaction of 9 types of soft constraints. If a soft constraint is violated by a solution, a penalty (a positive integer) is induced. These soft constraints SC1 - SC9 are defined as follows.

SC1: Gender policy is satisfied for each room.

SC2: Age policy is satisfied for each room.

SC3: A patient is assigned to a room with the required room properties for his/her treatment.

SC4: Some patients is allocated to a single room due to clinical reasons.

SC5: The room type preference for each patient is met.

SC6: A patient is allocated to a department that attends to his/her specialism.

SC7: A patient is allocated to a room that attends to his/her specialism in the first degree of priority.

SC8: A patient is assigned to a room with his/her preferred room properties.

SC9: Transfers should not be allowed.

The optimization objective of the PAS problem is to find a feasible assignment satisfying constraints HC1 - HC4 while minimizing a weighted sum of all the penalties of the unsatisfied soft constraints SC1 - SC9 (see Table 3.3). Formally, let Ω be the set of all feasible solutions (patient-to-bed assignments). For each $\boldsymbol{x} \in \Omega$, its cost is defined by:

$$Z(\boldsymbol{x}) = \sum_{i=1}^{9} W_i \cdot V_i(\boldsymbol{x}) \tag{2.1}$$

where $V_i(\boldsymbol{x})$ represents the number of times the i-th soft constraint is violated in solution $\boldsymbol{x}$, and W_i is the penalty weight corresponding to that soft constraint. The value of W_i are given in Table 2.1. Thus, the goal of the PAS problem is to find a feasible solution $\boldsymbol{x}^*$ such that for all $\boldsymbol{x} \in \Omega$, $Z(\boldsymbol{x}^*) \leq Z(\boldsymbol{x})$.

Table 2.1 – Weights of the constraints.

Constraint	SC1	SC2	SC3	SC4	SC5	SC6	SC7	SC8	SC9
Weight	5	10	5	10	0.8	1	1	2	11

2.2.2 Mathematical model

In this section, we first compare the differences of optimization models among those proposed in the literature for the static PAS problem. Taking into account the fact that beds in the same room have indistinguishable features and constraints, researchers generally formulate the PAS problem as a patient-room assignment (PRA) problem, which involves assigning each patient to a specific room. The main differences of the proposed MIP/IP models in the literature are the hard and soft constraints in the models with respect to the original problem statement. As mentioned in Section 1.2, Demeester et al. [25] first proposed a IP model considering the soft constraint SC1, SC2, SC3, SC4 as hard constraints and not allowing their violations. In contrast, Guido et al. [9] proposed two IP models where HM_{PBA} does not allow the soft constraint SC1, SC2, SC3, SC4, SC9 to be violated, and SM_{PBA} relaxes the restrictions of SC1 (gender policy N,D), SC2, SC3, SC9 (patient transfer are restricted to at most one for those who have two stays). Both Ceschia & Schaerf [27] and Bastos et al. [8] proposed IP/MIP models which allowed all soft constraints SC1 - SC9 to be violated. However, the former is more simplified than the latter, as it merges penalties associated with the patient-room assignment, including SC1 (gender policy M, F, N), SC2, SC3, SC4, SC5, SC6, SC7, and SC8, into a single penalty C_{pr} and avoids the generation of too many constraints and variables. Moreover, it is evident that the optimal solutions, in some instances, obtained by solving the IP/MIP models of [27] and [8] outperform those of [25] and [9], since the latter two models only allowed a subset of soft constraints. Based on the comparative analysis of the existing literature, it can be concluded that the IP model proposed by [27] is more flexible compared to the other models since it allows all soft constraints to be violated.

Since our work is based on the IP model proposed by [27], we summarize their reformulation below while the used notation is shown in Table 2.2. The objective function, denoted by (2.2), aims to minimize the total penalties associated with assigning patients to rooms for the duration of their hospitalization period. The first part of the objective function corresponds to the cost of assigning patients to rooms, which is determined by the combined penalty of soft constraints SC1 (gender policy M, F, N), SC2, SC3, SC4,

SC5, SC6, SC7, and SC8. The second part of the objective function incorporates the cost of violating the room gender policy, while the last part of the function captures the cost associated with patient transfer.

Table 2.2 – Notation used for the PRA model.

Symbol	Description
Sets	
$\mathcal{P}$	Set of patients ($p = 1, \cdots, \lvert\mathcal{P}\rvert$)
$\mathcal{D}$	Set of days ($d = 1, \cdots, \lvert\mathcal{D}\rvert$)
$\mathcal{R}$	Set of rooms ($r = 1, \cdots, \lvert\mathcal{R}\rvert$)
$\mathcal{D}_p \subset \mathcal{D}$	Set of days of patient p stay in hospital ($\mathcal{D}_p = \{AD_p, \cdots, DD_p - 1\}$)
$\mathcal{P}_M \subset \mathcal{P}$	Set of male patients
$\mathcal{P}_F \subset \mathcal{P}$	Set of female patients
$\mathcal{R}_D \subset \mathcal{R}$	Set of dependent rooms
Parameters	
Q_r	Number of beds in room r
LOS_p	Length of stay of patient p ($LOS_p = DD_p - AD_p$)
C_{pr}	the penalty of assigning patient p to room r. All the room penalties are incorporated into the value except SC1 (gender policy D) and SC9
W_{RG}	Weight of gender policy constraint
W_{Tr}	Weight of transfers constraint
Variables	
x_{prd}	1 if patient p is assigned to room r in day d, 0 otherwise
f_{rd}	1 if there is at least one female patient in room r in day d, 0 otherwise
m_{rd}	1 if there is at least one male patient in room r in day d, 0 otherwise
b_{rd}	1 if there are both male and female patients in room r in day d, 0 otherwise
t_{prd}	1 if patient p is transferred from room r in day d, 0 otherwise

$$\textbf{\textit{PRA:}} \quad \text{Min} \sum_{p\in\mathcal{P},r\in\mathcal{R},d\in\mathcal{D}_p} C_{pr}\cdot x_{prd} + \sum_{r\in\mathcal{R}_D,d\in\mathcal{D}} W_{RG}\cdot b_{rd} + \sum_{p\in\mathcal{P},r\in\mathcal{R},d\in\mathcal{D}} W_{Tr}\cdot t_{prd} \quad (2.2)$$

$$\text{s.t.} \quad \sum_{r\in\mathcal{R}} x_{prd} = 1, \quad \forall p\in\mathcal{P}, d\in\mathcal{D}_p \quad (2.3)$$

$$\sum_{p\in\mathcal{P}|d\in\mathcal{D}_p} x_{prd} \le Q_r, \quad \forall d\in\mathcal{D}, r\in\mathcal{R} \quad (2.4)$$

$$f_{rd} \ge x_{prd}, \quad \forall p\in\mathcal{P}_F, d\in\mathcal{D}, r\in R \quad (2.5)$$

$$m_{rd} \ge x_{prd}, \quad \forall p\in\mathcal{P}_M, d\in\mathcal{D}, r\in R \quad (2.6)$$

$$b_{rd} \geq m_{rd} + f_{rd} - 1, \quad \forall d \in \mathcal{D}, r \in R \tag{2.7}$$

$$t_{prd} \geq x_{prd} - x_{pr,d+1}, \quad \forall p \in \mathcal{P}, d \in \mathcal{D}, r \in \mathcal{R} \tag{2.8}$$

$$x_{prd} \in \{0,1\} \quad \forall p \in \mathcal{P}, d \in \mathcal{D}_p, r \in \mathcal{R} \tag{2.9}$$

$$b_{rd} \in \{0,1\} \quad \forall d \in \mathcal{D}, r \in \mathcal{R} \tag{2.10}$$

$$f_{rd} \in \{0,1\} \quad \forall d \in \mathcal{D}, r \in \mathcal{R} \tag{2.11}$$

$$m_{rd} \in \{0,1\} \quad \forall d \in \mathcal{D}, r \in \mathcal{R} \tag{2.12}$$

$$t_{prd} \in \{0,1\} \quad \forall p \in \mathcal{P}, d \in \mathcal{D}_p, r \in \mathcal{R} \tag{2.13}$$

Constraint (2.3) refers to complete assignment constraint which enforces every patient to be assigned to a room between admission and discharge dates. Constraint (2.4) refers to capacity constraint which ensures number of patients assigned to a room for a specific day cannot exceed the capacity of the room. Constraints (2.5)-(2.7) refer to gender policy constraints (GC_0). Variable b_{rd}, m_{rd}, f_{rd} and t_{prd} are all dependent on the different circumstances of the x_{prd} variables, which define the actual search space. If there is a female in a room, Constraint (2.5) forces the auxiliary variable f_{rd} to be equal to 1 to reflect the female existence in that room. A similar approach is taken for constraint (2.6) to reflect that there is a male in a room. If both genders exist in a room, Constraint (2.7) ensures that b_{rd} becomes 1 and gender penalty in the objective value is reflected accordingly. Finally, constraint (2.8) refers to patient transfer constraint (TC), which ensures the auxiliary variable t_{prd} becomes 1 if a patient changes room on two consecutive days. Constraints (2.9) - (2.13) define the domain of the variables.

2.3 Solution approach

To solve the PAS problem, we employ a two-stage optimization approach, which decomposes the given problem into two separate subproblems: a patient-room assignment (PRA) subproblem and a patient-bed assignment (PBA) subproblem. Fig. 2.1 illustrates the general framework of our proposed solution approach. As demonstrated by [27], the optimal solutions derived from these two subproblems can be integrated to achieve the optimal solution for the original problem. Our approach first generates a partial solution by solving an advanced PRA (APRA) model, which is based on the IP model of [27]. However, it is challenging to solve the APRA model directly because of the huge search space resulting from patient-room-day assignment variables. Thus, we employ a

warm start approach in which we solve the APRA model without transfers constraint (APRA^{WT}) to generate a high-quality feasible solution and then use the obtained solution as an initial solution to the APRA model. It is worth noting that using the above warm start approach can yield better results for the tested benchmark instances than directly solving the APRA model, as demonstrated in [8]. Secondly, our approach solves the PBA subproblem to allocate patients to beds of specific rooms according to the PRA solution, which is validated by an application made available online [1] by [25].

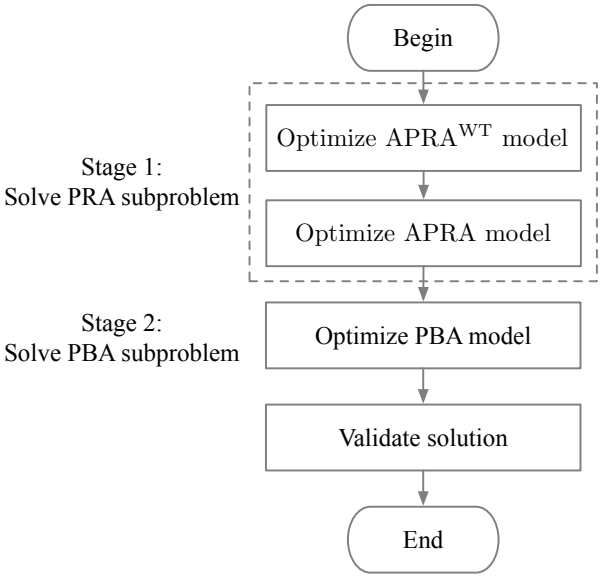


Figure 2.1 – Framework of two-stage optimization approach for the PAS problem.

2.3.1 Advanced patient-room assignment model

Large IP models are incapable of finding a high-quality solution within an acceptable time due to their large sizes. Let the parameter G_r represent the gender policy of room r, with the values 0, 1, 2, and 3 corresponding to the policies D, M, F, and N, respectively. The variables and constraints of the IP model of [27] can be decreased by considering the following rules:

1. Variables x_{prd} can be omitted from the model when $LOS_p = 0$.
2. Variables f_{rd}, m_{rd} and b_{rd} can be omitted from the model when $Q_r = 1$ or $G_r = 1, 2, 3$.

1. https://people.cs.kuleuven.be/~wim.vancroonenburg/pas/

3. Variables t_{prd} can be omitted from the model when $LOS_p < 2$ and $d = DD_p - 1$.
4. Constraints (2.3) and (2.9) can be omitted from the model when $LOS_p = 0$.
5. Constraints (2.5), (2.6) and (2.7) can be omitted from the model when $LOS_p = 0$, or when either $Q_r = 1$ or $G_r = 1, 2, 3$.
6. Constraint (2.8) can be omitted from the model when $Q_r = 1$ or $G_r = 1, 2, 3$.

In order to better apply the above rules, we introduce some notations presented in Table 2.3. It should be noted that a patient may have multiple specialisms in some instances, which means that during a patient's stay, the patient is assigned in the first part of his/her stay to a specialism, and the second part of his/her stay to another specialism. Also, the use of the parameter C_{pr} may lead to incorrect results when patients have multiple specialisms during their hospital stay. We therefore introduce a new parameter C_{prd} which is defined as the penalty of assigning patient p to room r on day d.

Table 2.3 – Notation used for the APRA model.

Symbol	Description
Sets	
$\mathcal{P}^E \subset \mathcal{P}$	Set of elective patients with $LOS_p \geq 1$
$\mathcal{M} \subset \mathcal{P}^E$	Set of male elective patients
$\mathcal{F} \subset \mathcal{P}^E$	Set of female elective patients
$\mathcal{R}_D^M \subset \mathcal{R}$	Set of dependent rooms with more than one bed
Parameter	
C_{prd}	Penalty of assigning patient p to room r on day d. All the room penalties are incorporated into the parameter except SC1 (gender policy D) and SC9

To avoid confusion, we refer to the APRA model under the gender policy constraint GC_0 and the transfer constraint TC as $APRA_{GC_0\&TC}$, which can be formulated as follows:

$$\boldsymbol{APRA}_{\boldsymbol{GC_0\&TC}}: \quad \text{Min} \quad S = \sum_{p\in\mathcal{P}^E, r\in\mathcal{R}, d\in\mathcal{D}_p} C_{prd}\cdot x_{prd} \sum_{r\in\mathcal{R}_D^M}, d\in\mathcal{D}W_{RG}\cdot b_{rd} + \sum_{p\in\mathcal{P}^E|LOS_p\geq 2, r\in\mathcal{R}, d\in\mathcal{D}_p\setminus\{DD_p-1\}} W_{Tr}\cdot t_{prd} \tag{2.14}$$

$$\text{s.t.} \quad \sum_{r\in\mathcal{R}} x_{prd} = 1, \quad \forall p\in\mathcal{P}^E, d\in\mathcal{D}_p \tag{2.15}$$

$$\sum_{p \in \mathcal{P}^E | d \in \mathcal{D}_p} x_{prd} \leq Q_r, \quad \forall d \in \mathcal{D}, r \in \mathcal{R} \tag{2.16}$$

$$f_{rd} \geq x_{prd}, \quad \forall p \in \mathcal{F}, d \in \mathcal{D}_p, r \in \mathcal{R}_D^M \tag{2.17}$$

$$m_{rd} \geq x_{prd}, \quad \forall p \in \mathcal{M}, d \in \mathcal{D}_p, r \in \mathcal{R}_D^M \tag{2.18}$$

$$b_{rd} \geq f_{rd} + m_{rd} - 1, \quad \forall d \in \mathcal{D}, r \in \mathcal{R}_D^M \tag{2.19}$$

$$t_{prd} \geq x_{prd} - x_{pr,d+1}, \quad \forall p \in \mathcal{P}^E | LOS_p \geq 2, d \in \mathcal{D}_p \setminus \{DD_p - 1\}, r \in \mathcal{R} \tag{2.20}$$

$$x_{prd} \in \{0,1\} \quad \forall p \in \mathcal{P}^E, d \in \mathcal{D}_p, r \in \mathcal{R} \tag{2.21}$$

$$b_{rd} \in \{0,1\} \quad \forall d \in \mathcal{D}, r \in \mathcal{R}_D^M \tag{2.22}$$

$$f_{rd} \in \{0,1\} \quad \forall d \in \mathcal{D}, r \in \mathcal{R}_D^M \tag{2.23}$$

$$m_{rd} \in \{0,1\} \quad \forall d \in \mathcal{D}, r \in \mathcal{R}_D^M \tag{2.24}$$

$$t_{prd} \in \{0,1\} \quad \forall p \in \mathcal{P}^E | LOS_p \geq 2, d \in \mathcal{D}_p \setminus \{DD_p - 1\}, r \in \mathcal{R} \tag{2.25}$$

Regarding gender policy constraints, we can modify the formulas presented in [8] to obtain alternative formulations (GC_1). We define a new binary variable u_{rd}, which has the value 1 if room r is reserved for females on day d, and 0 otherwise. The constraints GC_1 can be written as follows:

(GC_1)

$$x_{prd} \leq u_{rd} + b_{rd} \quad \forall p \in \mathcal{F}, d \in \mathcal{D}_p, r \in \mathcal{R}_D^M \tag{2.26}$$

$$x_{prd} \leq (1 - u_{rd}) + b_{rd} \quad \forall p \in \mathcal{M}, d \in \mathcal{D}_p, r \in \mathcal{R}_D^M \tag{2.27}$$

$$u_{rd} \in \{0,1\} \quad \forall d \in \mathcal{D}, r \in \mathcal{R}_D^M \tag{2.28}$$

Constraint (2.26) enforces female patient restrictions and (2.27) enforces male patient restrictions. Both constraints seek to avoid the assignment of two distinct genders to the same room, penalizing allocations in which different genders share a room. Thus, we refer the APRA model under constraints GC_1 and TC as $APRA_{GC_1\&TC}$.

2.3.2 Advanced patient-room assignment model without transfer constraints

It is quite challenging to solve the above two APRA models directly due to the large search space defined by the patient-room-day assignment variables. By prohibiting patient transfer during their stay, we limit the scope of the search space defined by patient-room

assignment variables, resulting in a special case of the APRA model, as used in [8, 9, 27]. In our APRA$^{\mathrm{WT}}$ model, transfers are not allowed so that a patient must stay in the same room during his/her entire length of stay. The solution of APRA$^{\mathrm{WT}}$ model will always be feasible to the APRA model since the solution space of the former is contained in that of the latter. Moreover, since transfers are associated with the highest penalty weight, it is reasonable to expect that the solution of APRA$^{\mathrm{WT}}$ models would be close to the optimal solution of APRA models. Hence, we first solve an APRA$^{\mathrm{WT}}$ model to obtain a feasible solution, which is used as the initial solution of the APRA model for further improvement.

Our APRA$^{\mathrm{WT}}$ models are inherited from the APRA models by removing variable t_{prd} and replacing variable x_{prd} by x_{pr}, a binary variable taking the value of 1 if patient p is allocated to room r, and 0 otherwise. Moreover, parameter C_{prd} is replaced by parameter $C'_{pr} = \sum_{d\in\mathcal{D}_p} C_{prd}$. Thus, the model $APRA^{WT}_{GC_0}$ is formulated as follows:

$$\boldsymbol{APRA}^{\boldsymbol{WT}}_{\boldsymbol{GC_0}}: \quad \text{Min} \quad S = \sum_{p\in\mathcal{P}^E}\sum_{r\in\mathcal{R}} C'_{pr}x_{pr} + \sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}^M_D} W_{RG}b_{rd} \tag{2.29}$$

s.t. constraints (2.19), (2.22), (2.23)

$$\sum_{r\in\mathcal{R}} x_{pr} = 1 \quad \forall p \in \mathcal{P}^E \tag{2.30}$$

$$\sum_{p\in\mathcal{P}^E|d\in\mathcal{D}_p} x_{pr} \le Q_r \quad \forall d \in \mathcal{D}, r \in \mathcal{R} \tag{2.31}$$

$$f_{rd} \ge x_{pr} \quad \forall p \in \mathcal{F}, d \in \mathcal{D}_p, r \in \mathcal{R}^M_D \tag{2.32}$$

$$m_{rd} \ge x_{pr} \quad \forall p \in \mathcal{M}, d \in \mathcal{D}_p, r \in \mathcal{R}^M_D \tag{2.33}$$

$$x_{pr} \in \{0,1\} \quad \forall p \in \mathcal{P}^E, r \in \mathcal{R} \tag{2.34}$$

Like the APRA model, constraint (2.30) refers to complete assignment constraint, constraint (2.31) refers to capacity constraint and constraints (2.19),(2.32)-(2.33) refer to gender policy constraints (GC_0). Furthermore, model $APRA^{WT}_{GC_1}$ can be obtained by replacing GC_0 with GC_1 (2.28), (2.35)-(2.36) in model $APRA^{WT}_{GC_0}$.

(GC_1) Constraint (2.28)

$$x_{pr} \le u_{rd} + b_{rd} \quad \forall p \in \mathcal{F}, d \in \mathcal{D}_p, r \in \mathcal{R}^M_D \tag{2.35}$$

$$x_{pr} \le (1 - u_{rd}) + b_{rd} \quad \forall p \in \mathcal{M}, d \in \mathcal{D}_p, r \in \mathcal{R}^M_D \tag{2.36}$$

2.3.3 Constraint aggregation

Despite using the rules we proposed in Section 2.3.1 to reduce the model size, the APRA and APRA$^{\mathrm{WT}}$ models are still large and hard to solve. To further accelerate the solution process, we propose constraint aggregation to reduce the number of gender policy constraints GC_0, GC_1 and patient transfer constraint TC considering that these constraints account for more than 95% of the total number of constraints in the APRA and APRA$^{\mathrm{WT}}$ models (see Section 2.4.2). Since the APRA$^{\mathrm{WT}}$ models are inherited from the APRA models, we take the APRA models as example to illustrate our aggregation method.

Aggregated gender policy constraint

For gender policy constraints GC_1, we propose aggregated gender policy constraints AGC_1 (2.28), (2.37)-(2.38) by aggregating the constraints (2.26)-(2.27) of different patients with the same gender for the same day and room.

(AGC_1) Constraint (2.28)

$$\sum_{p\in\mathcal{F}|d\in\mathcal{D}_p} x_{prd} \leq \lambda^F_{rd}(u_{rd}+b_{rd}) \quad \forall r\in\mathcal{R}^M_D, d\in\mathcal{D} \tag{2.37}$$

$$\sum_{p\in\mathcal{M}|d\in\mathcal{D}_p} x_{prd} \leq \lambda^M_{rd}(1-u_{rd}+b_{rd}) \quad \forall r\in\mathcal{R}^M_D, d\in\mathcal{D} \tag{2.38}$$

where $\lambda^F_{rd} = \min\{Q_r, |\mathcal{F}_d|\}$ and $\lambda^M_{rd} = \min\{Q_r, |\mathcal{M}_d|\}$ are coefficients, $|\mathcal{F}_d|$ and $|\mathcal{M}_d|$ are the number of female/male elective patients in day d. Thus, the aggregated model $APRA_{AGC_1\&TC}$ can be obtained by replacing GC_1 with AGC_1 in model $APRA_{GC_1\&TC}$.

The $APRA_{GC_1\&TC}$ is equivalent to the $APRA_{AGC_1\&TC}$ (see the proof A.1 of Theorem 1) under the following two conditions: (i) if ($\boldsymbol{x}$, $\boldsymbol{u}$, $\boldsymbol{b}$, $\boldsymbol{t}$) is a feasible solution to the $APRA_{GC_1\&TC}$, then it must be feasible to the $APRA_{AGC_1\&TC}$; (ii) if ($\boldsymbol{x}$, $\boldsymbol{u}$, $\boldsymbol{b}$, $\boldsymbol{t}$) is a feasible solution to the $APRA_{AGC_1\&TC}$, then it must be feasible to the $APRA_{GC_1\&TC}$.

Theorem 1 *The aggregated model $APRA_{AGC_1\&TC}$ is equivalent to the original model $APRA_{GC_1\&TC}$.*

The gender policy constraints GC_0 (2.17)-(2.19) can be reformulated by AGC_0 (2.19), (2.39)-(2.40) as follows:

(AGC_0) Constraint (2.19)

$$\lambda^F_{rd} f_{rd} \geq \sum_{p \in \mathcal{F} | d \in \mathcal{D}_p} x_{prd} \quad \forall d \in \mathcal{D}, r \in \mathcal{R}^M_D \tag{2.39}$$

$$\lambda^M_{rd} m_{rd} \geq \sum_{p \in \mathcal{M} | d \in \mathcal{D}_p} x_{prd} \quad \forall d \in \mathcal{D}, r \in \mathcal{R}^M_D \tag{2.40}$$

Thus, the aggregated model $APRA_{AGC_0\&TC}$ can be obtained by replacing GC_0 with AGC_0 in model $APRA_{GC_0\&TC}$. With reference to the proof of Theorem 1, it is easy to see that the aggregated model $APRA_{AGC_0\&TC}$ is equivalent to the original model $APRA_{GC_0\&TC}$.

To obtain the aggregated gender policy constraints used in the APRA$^{\text{WT}}$ models, we can replace the variable $x_{prd}, \forall p \in \mathcal{P}^E, d \in \mathcal{D}_p, r \in \mathcal{R}$ with $x_{pr}, \forall p \in \mathcal{P}^E, r \in \mathcal{R}$. This allows us to create two APRA$^{\text{WT}}$ models, denoted as $APRA^{WT}_{AGC_0}$ and $APRA^{WT}_{AGC_1}$. Additionally, the aggregated models $APRA^{WT}_{AGC_0}$ and $APRA^{WT}_{AGC_1}$ are equivalent to the original models $APRA^{WT}_{GC_0}$ and $APRA^{WT}_{GC_1}$, respectively.

Aggregated patient transfer constraint

To aggregate patient transfer constraint, variable $t_{prd}, \forall p \in \mathcal{P}^E, d \in \mathcal{D}_p, r \in \mathcal{R}$ is replaced by aggregated variable $z_{pd}, \forall p \in \mathcal{P}^E, d \in \mathcal{D}_p$, a binary variable taking the value of 1 if patient p is transferred to a new room in day d, and 0 otherwise. With this new aggregated variable, the objective function (2.3.1) need to be modified by (2.41).

$$\begin{aligned} \text{Min} \quad S = & \sum_{p \in \mathcal{P}^E, r \in \mathcal{R}, d \in \mathcal{D}_p} C_{prd} \cdot x_{prd} + \sum_{r \in \mathcal{R}^M_D} , d \in \mathcal{D} W_{RG} \cdot b_{rd} + \\ & \sum_{p \in \mathcal{P}^E | LOS_p \geq 2, d \in \mathcal{D}_p \setminus \{DD_p - 1\}} W_{Tr} \cdot z_{pd} \end{aligned} \tag{2.41}$$

Aggregate patient transfer constraint can be achieved by comparing the room number RN_r of two consecutive days to determine whether the patient is transferred. Therefore, aggregated patient transfer constraint ATC (2.42) - (2.44) can be reformulated as follows:

(ATC)

$$|\mathcal{R}| z_{pd} \geq \sum_{r \in \mathcal{R}} RN_r x_{prd} - \sum_{r \in \mathcal{R}} RN_r x_{pr,d+1} \quad \forall p \in \mathcal{P}^E | LOS_p \geq 2, d \in \mathcal{D}_p \setminus \{DD_p - 1\} \tag{2.42}$$

$$|\mathcal{R}|z_{pd} \geq \sum_{r\in\mathcal{R}} RN_r x_{pr,d+1} - \sum_{r\in\mathcal{R}} RN_r x_{prd}$$
$$\forall p \in \mathcal{P}^E | LOS_p \geq 2, d \in \mathcal{D}_p \setminus \{DD_p - 1\} \tag{2.43}$$
$$z_{pd} \in \{0,1\} \quad \forall p \in \mathcal{P}^E | LOS_p \geq 2, d \in \mathcal{D}_p \setminus \{DD_p - 1\} \tag{2.44}$$

Four APRA models can be generated by combining the objective function (2.41), the complete assignment constraint, the capacity constraint, along with different formulations of the (aggregated) gender policy constraint and the aggregated patient transfer constraint, namely $APRA_{GC_0\&ATC}$, $APRA_{GC_1\&ATC}$, $APRA_{AGC_0\&ATC}$, $APRA_{AGC_1\&ATC}$. Furthermore, each of these aggregated APRA models is equivalent to its corresponding APRA model under the constraint TC. For instance, $APRA_{GC_0\&ATC}$ is equivalent to $APRA_{GC_0\&TC}$, $APRA_{GC_1\&ATC}$ is equivalent to $APRA_{GC_1\&TC}$, $APRA_{AGC_0\&ATC}$ is equivalent to $APRA_{AGC_0\&TC}$, $APRA_{AGC_1\&ATC}$ is equivalent to $APRA_{AGC_1\&TC}$. See the proof A.2 of Theorem 2 for details.

Theorem 2 *The APRA model under the aggregated constraint ATC is equivalent to its corresponding APRA model under the constraint TC.*

An illustrative example

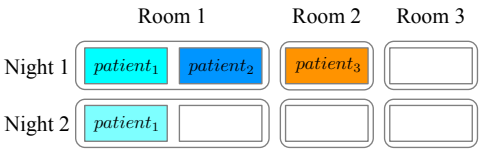
To illustrate our proposed aggregation method, consider an illustrative example of the PRA subproblem with 3 elective patients, 3 rooms, and 2 nights, as shown in Figure 2.2. Room 1, with 2 beds, follows policy D. Rooms 2 and 3 are both 1-bed rooms and follow policy M and F, respectively. The table on the top left lists the input data related to the patients, where the meaning of the symbols corresponds to the definition in Tables 2.2 and 2.3. The lower part of the table lists the constraints related to gender policy and patient transfer. The gender policy constraint GC_0 consists of 6 inequalities, whereas its aggregated counterpart AGC_0 contains 5 inequalities. Similarly, GC_1 involves 4 inequalities, while AGC_1 contains 3. For patient transfer, TC has 3 inequalities, while ATC has 2 inequalities. All 8 APRA models can be obtained by combining different formulations of the (aggregated) gender policy constraint and the (aggregated)patient transfer constraint, as well as the objective function, the complete assignment constraint, and the capacity constraint (here, the objective function, the complete assignment constraint, and the capacity constraint are omitted for brevity). All these APRA models have the same optimal solution, which is shown in the figure on the top right. The optimal objective function value is $S = 13.6$.

Instance

The sets and parameters related to patients

Patient	Gender	D_p	LOS_p	C_{prd}		
				$r=1$	$r=2$	$r=3$
1	F	{1,2}	2	{5, 5}	{10, 10}	{8, 8}
2	F	{1}	1	{0.8}	{5}	{2}
3	M	{1}	1	{3}	{2.8}	{5.8}

Optimal Solution: $S = 13.6$



(Aggregated) gender policy constraints and (aggregated) patient transfer constraints

Original

GC_0

$f_{1,1} \geq x_{1,1,1}$
$f_{1,2} \geq x_{1,1,2}$
$f_{1,1} \geq x_{2,1,1}$
$m_{1,1} \geq x_{3,1,1}$
$b_{1,1} \geq f_{1,1} + m_{1,1} - 1$
$b_{1,2} \geq f_{1,2} + m_{1,2} - 1$

GC_1

$x_{1,1,1} \leq u_{1,1} + b_{1,1}$
$x_{2,1,1} \leq u_{1,1} + b_{1,1}$
$x_{1,1,2} \leq u_{1,2} + b_{1,2}$
$x_{3,1,1} \leq 1 - u_{1,1} + b_{1,1}$

TC

$t_{1,1,1} \geq x_{1,1,1} - x_{1,1,2}$
$t_{1,2,1} \geq x_{1,2,1} - x_{1,2,2}$
$t_{1,3,1} \geq x_{1,3,1} - x_{1,3,2}$

Aggregation

AGC_0

$2f_{1,1} \geq x_{1,1,1} + x_{2,1,1}$
$f_{1,2} \geq x_{1,1,2}$
$m_{1,1} \geq x_{3,1,1}$
$b_{1,1} \geq f_{1,1} + m_{1,1} - 1$
$b_{1,2} \geq f_{1,2} + m_{1,2} - 1$

AGC_1

$x_{1,1,1} + x_{2,1,1} \leq 2(u_{1,1} + b_{1,1})$
$x_{1,1,2} \leq u_{1,2} + b_{1,2}$
$x_{3,1,1} \leq 1 - u_{1,1} + b_{1,1}$

ATC

$3z_{1,1} \geq x_{1,1,1} + 2x_{1,2,1} + 3x_{1,3,1} - x_{1,1,2} - 2x_{1,2,2} - 3x_{1,3,2}$
$3z_{1,1} \geq x_{1,1,2} + 2x_{1,2,2} + 3x_{1,3,2} - x_{1,1,1} - 2x_{1,2,1} - 3x_{1,3,1}$

Figure 2.2 – An illustrative example of PRA subproblem

It is worth noting that using the above aggregated constraint AGC_0, AGC_1 and ATC can reduce the number of constraints of APRA models, but it may also bring some disadvantages when solving the aggregated models using the Branch-and-Bound (B&B) approach. The B&B approach uses a lower bounding strategy based on linear program (LP) relaxation. However, aggregation enlarges the set of feasible solutions of the LP relaxation, thereby leading to weaker lower bounds. The weakened lower bounds may reduce the effectiveness of the B&B approach. The related detailed discussion can be found in [74], where the authors studied the effects of aggregation on the computational ease of the model. Therefore, to verify the effectiveness of the proposed aggregated constraints, we will compare the computational results of the models with and without aggregated constraints in Section 2.4.2.

2.3.4 Patient-bed assignment model

The PBA subproblem is created based on the outputs of the PRA subproblem to generate the patient-bed assignments. To build the PBA model, the hospital stay segment is introduced to indicate the patient transfer. If the room assigned to patient p in day d is the same as the room in consecutive hospitalization days, these days belong to the

same hospital stay segment s. The sets of hospital stay segments for all patients can be easily calculated according to the results of the PRA subproblem. The PBA is a feasibility problem, which is solved with a constant objective function set to zero, as shown in (2.45). The notation for the PBA is provided in Table 2.4, and its formulation is as follows:

Table 2.4 – PBA Notation

Symbol	Description
Sets	
$\mathcal{B}$	Set of beds ($b = 1, \cdots, \lvert\mathcal{B}\rvert$)
$\mathcal{S}_p$	Set of hospital stay segements of patient p ($s = 1, \cdots, \lvert\mathcal{S}_p\rvert$)
$\mathcal{B}_r \subset \mathcal{B}$	Set of beds in room r
$\mathcal{P}_{rd} \subset \mathcal{P}$	Set of patients assigned to room r in day d
$\mathcal{D}_{ps} \subset \mathcal{D}$	Set of days in hospital stay segement s of patient p
Parameters	
G_{ps}	Room assigned to patient p in his/her hospital stay segment s
Variables	
y_{pbs}	1 if patient p is assigned to bed b in hospital stay segment s, 0 otherwise

$$\text{Min } 0 \tag{2.45}$$

$$\text{s.t.} \quad \sum_{b \in \mathcal{B}_{G_{ps}}} y_{pbs} = 1 \quad \forall p \in \mathcal{P}, s \in \mathcal{S}_p \tag{2.46}$$

$$\sum_{p \in \mathcal{P}_{rd}} \sum_{s \in \mathcal{S}_p | d \in \mathcal{D}_{ps}, r = G_{ps}} y_{pbs} \leq 1 \quad \forall r \in \mathcal{R}, b \in \mathcal{B}_r, d \in \mathcal{D} \tag{2.47}$$

$$y_{pbs} \in \{0, 1\} \quad p \in \mathcal{P}, s \in \mathcal{S}_p, b \in \mathcal{B}_{G_{ps}} \tag{2.48}$$

Constraint (2.46) ensures every patient to be assigned to a bed for each segment. Constraint (2.47) limits assignments to the capacity of each bed for each night. Constraint (2.48) define the domain of the decision variable.

2.4 Experimental results and comparisons

In this section, we present computational results of our proposed solution method on the 13 instances provided by [6], and comparisons with existing state-of-the-art methods for PAS problem.

2.4.1 Experimental setting

Table 2.5 shows the details of these 13 instances in terms of the number of rooms ($|\mathcal{R}|$), dependent rooms with more than one bed ($|\mathcal{R}_D^M|$), total patient ($|\mathcal{P}|$), elective patient ($|\mathcal{P}^E|$), room properties ($Prop.$), beds ($|\mathcal{B}|$), specialisms (S), the length of the planning horizon ($|\mathcal{D}|$), and departments (K). In addition, only three room sizes are considered in this benchmark, i.e. 1, 2 and 4 beds.

The first six instances benefit from better patient-room compatibility compared with instances 7-13. In addition, the planning horizon is 14 days for instances 1-7, which is smaller than instances 8-13, where the planning horizon is between 21 days to 91 days. Therefore, instances 8-13 are more complex. It is worth mentioning that the total number of patients includes elective patients as well as patients whose LOS is zero, and patients whose discharge date lies beyond the planning horizon are scheduled until the last planning day. Moreover, only instance 13 has multi-spec patients. Specifically, all the 202 multi-spec patients require two specialisms, and no patient requires more than two specialisms in this benchmark dataset.

Table 2.5 – Characteristics of the problem instances.

Instance	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{R}_D^M\rvert$	$\lvert\mathcal{P}\rvert$	$\lvert\mathcal{P}^E\rvert$	$Prop.$	$\lvert\mathcal{B}\rvert$	S	$\lvert D\rvert$	K
1	98	82	693	652	2	286	4	14	4
2	151	132	778	755	2	465	6	14	6
3	131	114	757	708	2	395	5	14	5
4	155	136	782	746	2	471	6	14	6
5	102	93	631	587	2	325	4	14	4
6	104	93	726	685	2	313	4	14	4
7	162	32	770	519	4	472	6	14	6
8	148	34	895	895	4	441	6	21	6
9	105	18	1400	1400	4	310	4	28	4
10	104	20	1575	1575	4	308	4	56	4
11	107	21	2514	2514	4	318	4	91	4
12	105	28	2750	2750	4	310	4	84	4
13	125	30	907	907	4	368	5	28	5

Our model was implemented and solved using Gurobi Optimizer 9.0.3 with its default parameter settings. Branch-and-cut (B&C) is the default algorithm of Gurobi to solve the MIP models. Experiments are run on a cluster with each node running Linux with Inter(R) Xeon(R) Gold 6226R 2.90GHz CPU and 256Gb RAM. The number of CPU cores used was set to be 10. Experiments revealed that the average time to generate patient-room penalty matrix takes no more than 10 seconds, and solving the PBA model takes no more than 1 second. Thus, given a total time limit, we set 50 % of the run time to solve the

APRA^{WT} model and set the remaining time to solve the PRA model, which is the same as [8].

2.4.2 Evaluating the performance of different models for PRA subproblem

We generate 8 APRA models as well as 4 APRA^{WT} models, aiming to answer two critical questions: (i) do different models perform differently? (ii) if yes, which model performs the best for solving the PRA subproblem and why? In the following, we assess which model best suits the PRA subproblem using the benchmark sets. As described in Section 2.3, we use a warm start approach to solve the PRA subproblem, in which an APRA^{WT} model is solved in the initial step and then an APRA model is solved in the subsequent step. Specifically, the type of formula used for the gender policy constraint remains consistent between the APRA^{WT} model and the APRA model. As a result, this yields 8 different warm start procedures.

Model size can be used to roughly infer the performance of the solution. In general, a smaller model (with less constraints and variables) would be easier to handle. Thus, we first compare the average number of constraints and variables of all APRA^{WT}/APRA models we proposed as well as MIP models of [8], which generated most of the best known solutions and lower bounds, for 13 benchmark instances, as shown in Table 2.6. Notice that [8] also used the warm start approach to solve the PRA subproblem, and refereed to the model used in the initial step as the *simplified model (SM)* and the model used in the subsequent step as *complete model (CM)*. Additionally, the *SM* is a special case of the *CM* forbidding patient transfer.

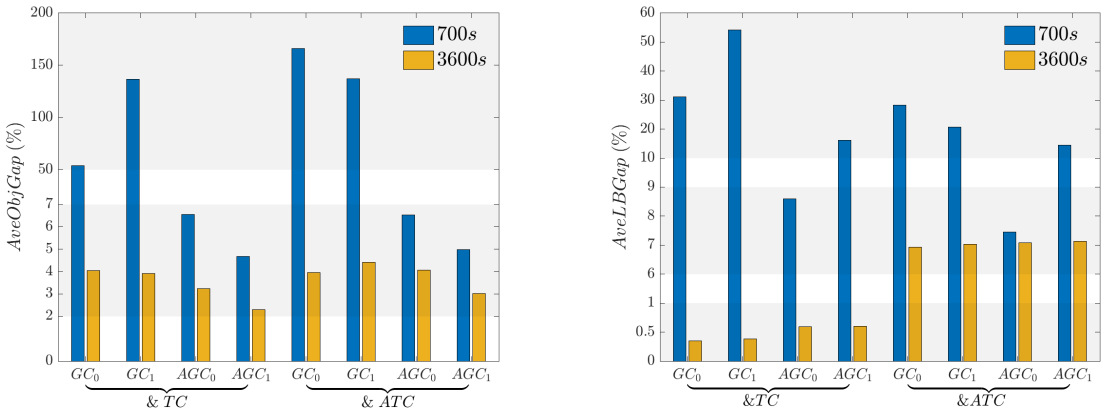
From Table 2.6, firstly, we can observe that our proposed models are significantly smaller than the MIP models of [8] in both initial and subsequent steps. Secondly, the model sizes of our proposed models are also significantly different. Specifically, in the initial step, the average number of variables and constraints of $APRA^{WT}_{GC_1}$/ $APRA^{WT}_{AGC_1}$ decreases by 10^3 compared to $APRA^{WT}_{GC_0}$/ $APRA^{WT}_{AGC_0}$. It is also clear that using the aggregated gender policy constraints can significantly decrease the model size. Compared to $APRA^{WT}_{GC_0}$/ $APRA^{WT}_{GC_1}$, the average number of constraints of $APRA^{WT}_{AGC_0}$/ $APRA^{WT}_{AGC_1}$ decreases by 97% after aggregating the gender policy constraints. Additionally, in the subsequent step, we can observe that the average number of constraints decreases by 33% after using $AGC_0(AGC_1)$ and the average number of constraints decreases by 65% after

using ATC in APRA. If both aggregated constraints $AGC_0(AGC_1)$ and ATC are used, the average number of constraints decreases by 97%. Moreover, the average number of variables decreases 44% after using ATC in APRA.

Table 2.6 – Comparison of different models used in warm start procedures over all benchmark instances

Warm start	Initial step			Subsequent step		
	Model	Var.	Con.	Model	Var.	Con.
Bastos et al. [8]						
WS_0	SM	5.20×10^6	6.60×10^6	CM	5.70×10^6	7.00×10^6
This work						
WS_1	$APRA^{WT}_{GC_0}$	1.41×10^5	2.65×10^5	$APRA_{GC_0\&TC}$	1.27×10^6	7.87×10^5
WS_2				$APRA_{GC_0\&ATC}$	6.58×10^5	2.78×10^5
WS_3	$APRA^{WT}_{AGC_0}$	1.41×10^5	8.65×10^3	$APRA_{AGC_0\&TC}$	1.27×10^6	5.30×10^5
WS_4				$APRA_{AGC_0\&ATC}$	6.58×10^5	2.19×10^4
WS_5	$APRA^{WT}_{GC_1}$	1.40×10^5	2.64×10^5	$APRA_{GC_1\&TC}$	1.17×10^6	7.85×10^5
WS_6				$APRA_{GC_1\&ATC}$	6.57×10^5	2.77×10^5
WS_7	$APRA^{WT}_{AGC_1}$	1.40×10^5	7.35×10^3	$APRA_{AGC_1\&TC}$	1.17×10^6	5.29×10^5
WS_8				$APRA_{AGC_1\&ATC}$	6.57×10^5	2.06×10^4

Next, we present a comparative results by using the above warm start procedures to solve the PRA subproblem. Due to the challenge of the PAS problem, we are more concerned with the *solution-quality* than *proven-optimality*. In order to assess the performance of the above warm start procedures under different solution time limits, two sets of results for each constraint combination were generated. The first was obtained with a short run time limit of 700 seconds. The second was obtained with a long run time limit of 3600 seconds (or until an optimal solution is found). For each warm start procedure, we summarize the average percentage gap $AveObjGap(\%) = \sum_{i\in N} \frac{Obj_i - BKS_i}{BKS_i * N} * 100$ of the best objective values Obj obtained by our approach from the best known objective values BKS reported in [8, 9] over the N benchmark instances ($N = 13$ in our case) and illustrate the results in Figure 2.3(a). Similarly, the average percentage gap $AveLBGap(\%) = \sum_{i\in N} \frac{BLB_i - LB_i}{BLB_i * N} * 100$ of the best lower bounds LB from the best known lower bounds BLB [8] are summarized in Figure 2.3(b). To make the results more readable, we properly scaled the vertical axis of the figures. From Figure 2.3(a), we observe that under the short and long run times, the model under the aggregated gender policy constraint AGC_0 and AGC_1 (WS_3, WS_4, WS_7, WS_8) can significantly improve the so-



(a) Solution quality **(b)** Lower bound (LB) quality

Figure 2.3 – Performance of different models in solving the PRA subproblem

lution quality, while using the aggregated patient transfer constraint ATC (WS_2, WS_6) fails to improve the solution quality. Furthermore, the models under constraint GC_0 (WS_1, WS_2) perform similarly to the models under constraint GC_1 (WS_5, WS_6), whereas the models under constraints AGC_1 always perform better than the models under AGC_0.

From Fig. 2.3(b), we observe that the more aggregation constraints are used, the better LBs are generated in a short run time. On the contrary, the more aggregation constraints are used, the worse LBs are generated in a long run time. Specifically, the LB quality generated by warm start procedures in a long run time are as follows: WS_1 (0.35%) < WS_5 (0.38%) < WS_3 (0.60%) = WS_7 (0.60%) < WS_2 (6.93%) < WS_6 (7.03%) < WS_4 (7.09%) < WS_8 (7.14%). Moreover, the LBs generated by solving the models under the constraint ATC (WS_2, WS_4, WS_6, WS_8) are difficult to improve for a longer run time, while LBs generated by solving the models under the constraint TC (WS_1, WS_3, WS_5, WS_7) are easy to improve in comparison.

These results indicate that the impact of using the aggregated constraints is two-fold. First, using constraint aggregation can significantly reduce the model size and using appropriate aggregated method can make the model easier to solve. Second, solving the model under aggregated constraints may quickly generate a lower bound for the minimization problem, but the quality of the lower bound is difficult to improve in a long run time. The reason for this phenomenon can be explained as Section 2.3.3. In summary, this experiment demonstrates that 1) appropriate formulas of gender policy constraints are essential for solving the APRA$^{\text{WT}}$ model, and 2) the APRA model as the core component

of the first stage of our two-stage optimization method and $APRA_{AGC_1\&TC}$ (WS_7) using constraints AGC_1 and TC perform the best among all the models we examined.

2.4.3 Comparison with state-of-the-art results

In order to compare our best results with those obtained in previous works, we perform additional experiments by running warm start procedure WS_7 for a time limit of 24 hours as the previous works [27, 8]. Due to the differences in computers and Gurobi versions between our work and previous studies, we implement the MIP models of [8] with warm start WS_0 to solve the PRA subproblem and use Gurobi Optimizer 9.0.3 with its default settings to solve the model.

To compare our results with previous studies, we adjusted the reported run times to account for CPU performance. We followed the approach used in [8], which is based on the approach proposed in [107]. The performance ratings of different CPUs have been obtained online [1] and are shown in the table 2.7. In our study, we used 10 of 16 available cores (20 of 32 threads) on the Intel Xeon Gold 6226R 2.90 gigahertz processor. To the best of our knowledge, no public performance data exists for this specific configuration. Given that CPU performance does not scale linearly with the number of cores, we first calculated the ratio of the performance degradation as follows: $\frac{\text{Average CPU mark}}{\text{Total threads}\times\text{Single thread rating}} = \frac{26240}{2294\times32} \approx 0.357$. Then, we calculate the estimated CPU mark for 10 cores by 20 threads $\times$ single thread rating $\times$ ratio of the performance degradation, i.e., $20 \times 2294 \times 0.357 \approx 16379$. The above approach was used to adjust the run times reported in [8] and [9].

Table 2.8 contrasts the best known results in the literature with our best results. Under the header "Literature Results", we present the best known objective values BKS and best known lower bounds BLB for each of the 13 benchmark instances. Moreover, the reference papers and computational times (adjusted following the procedure detailed previously) associated with these values have been reported. We show the results generated by the two-stage approach with the literature's MIP model under the header "Warm start WS_0 (MIP models of [8])", and report our results generated by the two-stage approach using the best model we proposed under the header "Warm start WS_7". For each approach, we record the best objective (Obj), the total computation time to find the best solution, the total computation time when Gurobi either proves the optimality or reaches the time limit (24 hours), the best lower bound (LB) and the number of branch-and-bound nodes

1. https://www.passmark.com/

visited after the root node in *SM*, *CM*, APRA$^{\text{WT}}$ and APRA models. We compute the percentage gaps $GAP(\%) = \frac{Obj-LB^*}{LB^*} \times 100$ of the best objective value found by each approach from the best lower bound LB^*, which is the maximum value among the lower bounds reported in the literature as well as those obtained in our study. Furthermore, we present the objective values, total computation time and the lower bounds as reported by [8], which were obtained using the warm start approach.

Table 2.7 – Optimization solvers and performance evaluation of CPU

Reference	Solver	Processor	Single thread rating	Average CPU mark	Used cores/ Total cores	Used threads/ Total threads
[9]	Cplex 15.5.1	Intel Xeon E5-1620 3.60 gigahertz 32 gigabytes RAM	1774	5863	4*/4	8*/8
[8]	Gurobi 7.5	Intel i7-3960 × 3.3 gigahertz 64 gigabytes RAM	1793	8390	6*/6	12*/12
This study	Gurobi 9.0.3	Intel Xeon Gold 6226R 2.90 gigahertz 256 gigabytes RAM	2294	26240 (16379**)	10/16	20/32

* The authors did not set the specific number of cores. By default, Gurobi and Cplex generally use all of the cores and threads of the machine.
** Estimated CPU mark for 10 cores.

We first compare the solution generated by our reproduced MIP models with the solution presented in [8]. From the perspective of the solution quality (best objective value), the results of 5 instances (9,10,11,12,13) are worse than those in the literature, 6 instances (1,3,4,5,6,7) are the same as in the literature and 2 instances (2,8) are better than those in the literature. From the perspective of the quality of the lower bounds, the LB of instance 4 are worse than it in the literature, 4 instances (1,3,5,6) are same to literature and 8 instances (2,7,8,9,10,11,12,13) are better than those in the literature. The reasons of above results are due to the used Gurobi version and the performance difference of the computing machines.

Second, we note that our approach generated new best solutions for 6 out of the 13 tested benchmark instances (2,4,8,9,10,13, note that solutions obtained for instances 1,3,5,6 and 7 are the same as the best known solutions reported in the literature; nevertheless, they were proven to be optimal by our $APRA_{AGC_1\&TC}$ model within an hour). Furthermore, our approach improved the best lower bound for 6 out of the 13 instances (2,3,4,7,8,13). It is worth noting that the optimality of the solution was also proven for instance 2. Although we have not proven the optimality of instances 4 and 8, the gaps are very low ($< 1\%$). In particular, although our approach fails to improve the best known solutions for instances 11 and 12 within a running time limit of 24 hours, it outperformed the method proposed in [8]. The failure of our approach to improve upon the best known

Table 2.8 Comparison between best known solutions and IP results (new best solutions and new best lower bounds in **bold**, proven optimal solutions in star *).

Instance	Literature results		Warm start WS_0 (MIP models of [8])										Warm start WS_7						
			Reported			Our reproduced													
	BKS (Time to end**)	BLB (Time to end**)	Obj	LB	Time to end**	Obj	Time to best	Time to end	LB	GAP (%)	Node of SM	Node of CM	Obj	Time to best	Time to end	LB	GAP (%)	Node of APRA^{WT}	Node of APRA
1	651.20 (21,226[8])	651.20 (21,226[8])	651.20	651.20	21,226	651.20*	4,229	19,670	651.20	0.00	91,557	90,609	651.20*	303	1,983	651.20	0.00	1,146	10,811
2	1,128.00 (44,258[8], 2,577[9])	1,115.80 (44,258[8])	1,128.00	1,111.60	44,258	**1,125.60**	30,082	86,400	1,116.20	0.00	9,705	1	**1,125.60***	2,947	25,358	**1,125.60**	0.00	627	22,879
3	761.60 (44,258[8], 2,577[9])	758.60 (44,258[8])	761.60	758.60	44,258	761.60	10,584	86,400	758.60	0.00	644,322	88,845	761.60*	1,315	13,561	**761.60**	0.00	32,952	50,736
4	1,151.60 (44,258[8])	1,143.20 (44,258[8])	1,151.60	1,143.20	44,258	1,151.60	23,864	86,400	1,142.80	0.14	83,928	1	**1,151.00**	21,138	86,400	**1,150.00**	0.09	744,351	326,359
5	624.00 (4,227[8], 62[9])	624.00 (4,227[8])	624.00	624.00	4,227	624.00*	1,073	5,196	624.00	0.00	12,453	12,327	624.00*	286	521	624.00	0.00	679	2,275
6	792.60 (10,082[8], 2,577[9])	792.60 (10,082[8])	792.60	792.60	10,082	792.60*	6,979	11,111	792.60	0.00	377	2,420	792.60*	1,185	2,130	792.60	0.00	8,468	1
7	1,176.40 (6,209[8], 2,577[9])	1,176.40 (6,209[8])	1,176.40	1,175.20	44,258	1,176.40*	451	3,105	1,176.40	0.00	5,193	23,729	1,176.40*	139	696	1,176.40	0.00	865	1,028
8	4,063.00 (44,258[8])	4,024.41 (44,258[8])	4,063.00	4,023.09	44,258	**4,058.60**	47,334	86,400	4,030.20	0.51	76,826	468	**4,058.60**	3,184	86,400	**4,038.00**	0.51	18,531	1,041
9	20,718.60 (3,866[9])	19,632.00 (44,258[8])	20,904.60	19,621.73	44,258	21,109.40	71,202	86,400	**19,872.80**	6.22	14,949	1	**20,677.40**	85,995	86,400	19,862.40	4.05	36,965	1
10	7,804.60 (2,577[9])	7,687.33 (44,258[8])	7,830.40	7,676.04	44,258	7,882.80	85,023	86,400	**7,696.60**	2.42	1,688	1	**7,799.80**	85,991	86,400	7,680.40	1.34	13,705	1
11	11,491.80 (2,577[9])	10,987.72 (44,258[8])	11,932.00	10,914.57	44,258	12,014.80	48,384	86,400	10,937.60	9.35	1	0	11,630.20	48,000	86,400	10,727.05	5.85	1	1
12	22,707.20 (682[9])	21,886.60 (44,258[8])	24,198.40	21,600.30	44,258	24,776.00	11,068	86,400	21,845.00	13.20	1	1	23,234.20	48,000	86,400	21,686.00	6.16	1	1
13	9,109.80 (345[9])	8,842.80 (44,258[8])	9,114.40	8,842.11	44,258	9,148.80	64,047	86,400	8,863.20	3.15	32,044	2	**9,102.20**	67,091	86,400	**8,869.30**	2.63	5,465	1

** Total computation time reported by the corresponding reference, adjusted following the procedure from [107]

solutions for instances 11 and 12 can be attributed to the significantly larger number of patients and planning periods in these instances. Specifically, these instances have approximately 2-5 times more patients and 2-6 times more planning periods compared to the others. We also note that our approach is outperformed by the method proposed in [8] in terms of the lower bounds for instances 9, 10, 11 and 12. This can be attributed to the fact, as mentiond in Section 4.3.3., that the quality of the linear program (LP) relaxation bound for aggregated model after constraint aggregation is usally poor compared to the LP relaxation of the original model.

Third, we analyze the performance of various models in terms of the number of branch-and-bound nodes visited after the root node. Specifically, we focus on the $APRA^{WT}_{AGC_1}$ and $APRA_{AGC_1\&TC}$ models from our WS_7, and the SM and CM models from WS_0. On the one hand, we can observe that for the instances that are solved to optimality by WS_7, the number of nodes in the B&C procedure for WS_7 is generally less than WS_0. Specifically, for the instances 1,5,6,7 which are also solved to optimality by WS_0, WS_0 generated on average 27395 nodes in SM and 31727 nodes in CM, while WS_7 generated on average 2790 nodes in APRA$^{\text{WT}}$ and 3529 nodes in APRA. For instances 2 and 3 which can not be solved to optimality by the WS_0, WS_0 generated on average 327014 nodes in SM and 44423 nodes in CM, while WS_7 generated on average 16790 nodes in APRA$^{\text{WT}}$ and 36808 nodes in APRA. On the other hand, for the instances 4,8,9,10,11,12,13 which can not be solved to optimality by both WS_0 and WS_7, the number of nodes in the B&C procedure for WS_7 is generally more than WS_0. In particular, WS_0 generated on average 71348 nodes in SM and 68 nodes in CM, while WS_7 generated on average 117003 nodes in APRA$^{\text{WT}}$ and 46772 nodes in APRA. These results are explained by the fact that the B&C procedure for $APRA^{WT}_{AGC_1}$ and $APRA_{AGC_1\&TC}$ leads to considerably smaller branch-and-bound trees.

The above comparisons demonstrate that our proposed two-stage optimization approach, featuring the APRA$^{\text{WT}}$ and APRA models, is able to significantly reduce the computation time compared to the MIP model proposed by [8]. The APRA$^{\text{WT}}$ and APRA models, refined from the IP model proposed by [27] by our proposed 6 rules and constraint aggregation, have fewer variables and constraints than those of [8]. In particular, $APRA^{WT}_{AGC_1}$ and $APRA_{AGC_1\&TC}$, the best version of our proposed models, achieve reductions by 97.29% and 79.47% for variables, and 99.89% and 92.44% for constraints, respectively. This substantial simplification aligns with the general principle that smaller models are typically easier to solve than their larger counterparts. Consequently, the num-

ber of nodes in the B&C procedure is less than that of [8], which leads to a significant reduction in the computation time.

Finally, we present a breakdown of the cost of our best solutions into different objective components in Table 2.9. It reports for each instance, the total penalty (Cost), the penalty associated with gender policy violations (Gen.), the penalty associated with age policy violations (Age), the penalty not attending to the needed treatment properties (Ned. prop.), the penalty related to single policy violations (Sng.), the penalty for failing to assign a patient to a room with his/her preferred capacity (Room pref.), the penalty incurred by not assigning a patient to the appropriate department (Dept.), the penalty incurred by not accounting for the prioritized specialism (Spec.), the penalty not attending to the preferred treatment properties (Pref. prop.), and the penalty related to transfers policy violations (Trs.).

Table 2.9 – Breakdown of the cost components for the best solutions.

Instance	Cost	Gen.	Age	Ned. pref.	Sng.	Room pref.	Dept.	Spec.	Pref. prop.	Trs.
1	651.2	0.0	0.0	0.0	0.0	651.2	0.0	0.0	0.0	0.0
2	1125.6	0.0	0.0	0.0	0.0	1113.6	0.0	12.0	0.0	0.0
3	761.6	0.0	0.0	0.0	0.0	753.6	0.0	8.0	0.0	0.0
4	1151.0	0.0	0.0	0.0	0.0	1040.0	0.0	75.0	36.0	0.0
5	624.0	0.0	0.0	0.0	0.0	624.0	0.0	0.0	0.0	0.0
6	792.6	0.0	0.0	0.0	0.0	789.6	0.0	3.0	0.0	0.0
7	1176.4	0.0	0.0	0.0	0.0	730.4	20.0	158.0	268.0	0.0
8	4058.6	0.0	0.0	0.0	0.0	1433.6	212.0	869.0	1522.0	22.0
9	20677.4	340.0	4500.0	1010.0	0.0	2702.4	282.0	1124.0	10554.0	165.0
10	7799.8	15.0	0.0	0.0	0.0	2964.8	2.0	486.0	4332.0	0.0
11	11630.2	10.0	0.0	5.0	0.0	4327.2	9.0	959.0	6320.0	0.0
12	23234.2	585.0	0.0	195.0	0.0	4823.2	280.0	1751.0	15600.0	0.0
13	9102.2	25.0	30.0	35.0	0.0	2091.2	655.0	1730.0	4470.0	66.0

Most penalties in instances 1-6 are caused by not being able to satisfy room capacity preferences, and specialisms and room properties preferences also contribute in the same cases, as reported by [10, 8]. In addition to the above penalties, department violations appeared for instance 7. For instances 8-13, preferred treatment properties violations account for most of the cost, and department, Specialism, and preferred room capacity violations have been consistently detected. Moreover, age policy violations appeared for instance 9 and 13, and gender violations appear for instances 9, 10, 11, 12 and 13. Finally, we note that the transfer violations were reported in instances 8, 9 and 13.

We also used our two-stage optimization approach to solve the original PAS problem, and the results are given in Appendix C.

2.5 Chapter conclusion

In this chapter, we presented a two-stage exact method for solving the patient admission scheduling (PAS) problem, which decomposes the PAS problem into two separate problems, including the patient-room assignment (PRA) subproblem and the patient-bed assignment (PBA) subproblem. To solve the PRA subproblem, we applied a warm start approach in which we solve the APRA^{WT} model to generate a high-quality feasible solution and then use the obtained solution as a warm start to the APRA model.

Our approach generated new best solutions for 6 out of the 13 benchmark instances from a publicly available repository, and proved the optimality of the solution for one of these 6 instances. Moreover, for 5 other instances, we obtained the known optimal solutions in a short time compared to the methods in the literature. Finally, we also applied our approach to the original PAS problem and performed computational experiments on the same 13 benchmark instances. We obtained 5 new best solutions, 6 new best lower bounds, and proved optimality for 6 instances.

Chapter 3

Stochastic patient admission scheduling with an exponential number of scenarios

In this chapter, we present a study on a stochastic variant of the patient admission scheduling problem, which aims to assign patients to rooms during hospitalization under the consideration of overstay risk. We build two-stage stochastic programming models to formulate the problem, where the first stage assigns patients to rooms on the planned hospitalization days, and the second stage evaluates the expected costs resulting from patient overstay. Compared to the typical scenario-based model, the proposed state-variable model is significantly reduced by having a pseudo-polynomial number of variables and constraints. To solve the state-variable model efficiently, we introduce the sample average approximation (SAA) method as the first attempt to provide a high-quality initial feasible solution for the Gurobi solver. Extensive computational experiments were conducted to evaluate the performance of the proposed models and SAA-SV method. Experimental results show that our SAA-SV method outperforms the methods of directly solving the scenario-based model and the state-variable model in terms of solution quality and computational time. In particular, the SAA-SV method can provide high-quality solutions with an average optimality gap of 1.73% for instance sizes reaching 500 patients and 3.3 $\times$ 10^{150} scenarios within 1 hour. Additional analysis has also been carried out to verify the advantage of our proposed method over the typical deterministic approach and the SAA method. The content of this chapter is based on an article submitted to *INFORMS Journal on Computing*.

3.1 Introduction

Previous studies on the PASU problem assume deterministic overstay lengths of patients, so the deterministic version of the PASU problem is actually tackled. In practice, inpatient LOS is related with infections, delayed treatments, and postoperative complications [108]. It is thus quite important to consider the stochastic nature of the overstay length when making patient admission plan. Otherwise, prolonged stays and cancellations could be encountered during the execution of the admission plan [8, 12, 109]. Hence, we investigate the first offline PASU problem under the uncertain overstay lengths of patients and call it as the stochastic PAS (SPAS) problem. Note that the consideration of the uncertain LOS has been mostly studied in the surgical case scheduling problem [88, 87, 110, 111, 112, 113, 114, 90, 115]. Our research is dedicated to proposing a novel state-variable model and an efficient solution method for the SPAS problem. The main contributions of our work are summarized as follows.

(1) We propose a new stochastic patient admission scheduling (SPAS) problem by considering the uncertainty of the patient overstay. We treat the SPAS problem as a two-stage stochastic programming problem, where the first stage assigns patients to rooms on the deterministic hospitalization days, and the second stage evaluates the expected cost resulting from patient overstay.

(2) We propose a scenario-based model $SPAS_{SB}$, which evaluates the expected cost by enumerating all possible scenarios. To deal with the exponential number of scenarios, we propose its equivalent state-variable model $SPAS_{SV}$ to reformulate the second stage by introducing a set of state variables, state transition constraints and linking constraints. To solve the $SPAS_{SV}$ model efficiently, we propose a new SAA-SV method by additionally using a sample average approximation (SAA) method to generate an initial feasible solution for the Gurobi solver. To the best of our knowledge, this work is the first attempt to hybridize SAA method with state-variable modeling approach for handling stochastic programming problems.

(3) Extensive computational experiments demonstrate that our proposed SAA-SV method significantly outperforms the methods of directly solving the $SPAS_{SB}$ model and $SPAS_{SV}$ model. In particular, the SAA-SV method is capable of finding solutions with an average optimality gap of 1.73% for large instances reaching 500 patients and 3.3×10^{150} scenarios in 1 hour. In addition, we verify the advantage of the SAA-SV method over the typical deterministic approach and the SAA method.

Next section presents the definition of the SPAS problem along with the $SPAS_{SB}$ formulation. Section 3.3 presents the $SPAS_{SV}$ formulation and the solution method. Section 3.4 presents the computational results and experimental analysis of the proposed formulations and solution method. Section 3.5 draws the conclusions.

3.2 Problem description and scenario-based stochastic programming model

In this section, we provide the problem definition and terminology of the SPAS problem and its scenario-based stochastic programming model.

3.2.1 Problem description

The terminology of the SPAS problem is similar to that of the standard PAS problem which is presetned in Chapter 2. The difference is as follows: In the SPAS problem, each patient has a planned admission date, while the actual admission may be delayed, but no more than a given number of days. Moreover, each patient may have a risk of extending his/her LOS, termed *overstay*, which introduces the possibility that the patient may need to spend more nights in the hospital. Note that Ceschia & Schaerf [12] assumes that patients may spend one extra night in the hospital. We consider that patients may spend more than one extra nights in the hospital, following [116], which is more realistic.

In the SPAS problem, a solution is feasible if each patient is assigned with a bed such that no hard constraint of types HC1 - HC7, given below, is violated.

HC1: *Patient admission* - Each patient is admitted during a specified range of days.

HC2: *Consecutive nights* - Patient's LOS is continuous. Each patient must stay in the hospital for the planned LOS.

HC3: *Department specialism* - Patients must be treated at the departments where the specialisms they need are offered.

HC4: *Mandatory equipment* - A patient is assigned to a room with the required room properties for his/her treatment.

HC5: *Age policy* - Age policy must be satisfied for all rooms.

HC6: *Room capacity* - Beds allocated to patients should not overlap on any given night.

HC7: *No transfers* - Patients cannot be transferred during the overstay period.

The quality of a feasible solution depends on the satisfaction of 7 types of soft constraints. If a soft constraint is violated by a solution, a penalty (a positive integer) is induced. These soft constraints SC1 - SC7 are defined as follows.

SC1: *Delayed admission* - Patient's admission can be delayed, but the sooner the patient is admitted, the better.

SC2: *Gender policy* - Gender policy is satisfied for each room.

SC3: *Levels of expertise* - A patient is assigned to a department with the highest priority degree for his/her specialism.

SC4: *Preferred room capacity* - The room type preference for each patient is met.

SC5: *Preferred equipment* - A patient is assigned to a room with his/her preferred room properties.

SC6: *Transfers* - A patient can be transferred from one room to another room over the planned dates of stay. But the fewer transfers, the better.

SC7: *Overcrowd risk* - Overcrowded rooms are allowed only if they are used by overstay patients and the maximum overcapacity cannot exceed a given threshold.

The optimization objective of the SPAS problem is to find a feasible assignment satisfying constraints HC1 - HC7, while minimizing a weighted sum of all the penalties of the unsatisfied soft constraints SC1 - SC7. Formally, let X be the set of all feasible solutions. For each $\boldsymbol{x} \in X$, its cost is defined by:

$$Z(\boldsymbol{x}) = \sum_{i=1}^{6} W_i V_i^{PL}(\boldsymbol{x}) + \mathbb{E}_\omega \left[\sum_{i=2|i\neq 6}^{7} W_i V_{i\omega}^{OP}(\boldsymbol{x}) \right] \tag{3.1}$$

where $V_i^{PL}(\boldsymbol{x})$ represents the number of times the i-th soft constraint is violated in solution $\boldsymbol{x}$ during the planned LOS of each patient. $V_{i\omega}^{OP}(\boldsymbol{x})$ denotes the number of times the i-th soft constraint is violated in solution $\boldsymbol{x}$ for the overstay period under scenario ω, where each scenario represents a possible realization of the overstay length for each patient. $\mathbb{E}_\omega[\cdot]$ means the expected total penalty for solution $\boldsymbol{x}$ during the overstay period, with respect to the probability distribution of the overstay length of each patient. W_i is the penalty weight of the i-th soft constraint. Thus, the objective of the SPAS problem is to find a feasible solution $\boldsymbol{x}^*$ such that for all $\boldsymbol{x} \in X$, $Z(\boldsymbol{x}^*) \leq Z(\boldsymbol{x})$.

3.2.2 Scenario-based stochastic programming model

To formulate the SPAS problem, we consider it as a two-stage stochastic programming problem as follows: the first stage decides the admission date of each patient from the allowing dates and assigns suitable rooms on their planned hospitalization days; the second stage estimates the expected costs, including penalties for assigning patients to rooms within the overstay period, penalties for overcrowding and violation of gender policy in each room on each day. Therefore, we propose a scenario-based stochastic optimization $SPAS_{SB}$ model, where scenarios are used to represent the probability distributions of the overstay length of each patient. The notation used in $SPAS_{SB}$ is shown in Table 3.1, and the model is formulated as follows:

$$\boldsymbol{SPAS_{SB}}: \text{Min} \sum_{p\in\mathcal{P}}\sum_{i\in\mathcal{D}_p^A} W^{De}(i-D_p^{A0})\alpha_{pi} + \sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}_p}\sum_{i\in\mathcal{D}_p^A}\sum_{d\in\mathcal{D}_{pi}^L} C_{pr}x_{prid} \tag{3.2}$$
$$+ \sum_{p\in\mathcal{P}|L_p\geq 2}\sum_{d\in\mathcal{D}_p} W^{Tr}t_{pd} + \sum_{\omega\in\Omega}\Pr(\omega)Q_1(\boldsymbol{x},\omega)$$

s.t.
$$\sum_{i\in\mathcal{D}_p^A}\alpha_{pi} = 1 \quad \forall p\in\mathcal{P} \tag{3.3}$$
$$\sum_{r\in\mathcal{R}_p} x_{prid} = \alpha_{pi} \quad \forall p\in\mathcal{P}, i\in\mathcal{D}_p^A, d\in\mathcal{D}_{pi}^L \tag{3.4}$$
$$\sum_{p\in\mathcal{P}|r\in\mathcal{R}_p}\sum_{i\in\mathcal{D}_p^A|d\in\mathcal{D}_{pi}^L} x_{prid} \leq B_r \quad \forall r\in\mathcal{R}, d\in\mathcal{D} \tag{3.5}$$
$$t_{pd} \geq \sum_{i\in\mathcal{D}_p^A|d\in\mathcal{D}_{pi}^L} x_{prid} - \sum_{i\in\mathcal{D}_p^A|d-1\in\mathcal{D}_{pi}^L} x_{pri,d-1} - \mathbb{1}(d\in\mathcal{D}_p^A)\alpha_{pd} \tag{3.6}$$
$$\forall p\in\mathcal{P}|L_p\geq 2, r\in\mathcal{R}_p, d = D_p^{A0}+1,...,D_p^{A1}+L_p-1$$
$$\alpha_{pi}\in\{0,1\} \quad \forall p\in\mathcal{P}, i\in\mathcal{D}_p^A \tag{3.7}$$
$$x_{prid}\in\{0,1\} \quad \forall p\in\mathcal{P}, r\in\mathcal{R}_p, i\in\mathcal{D}_p^A, d\in\mathcal{D}_{pi}^L \tag{3.8}$$
$$t_{pd}\in\{0,1\} \quad \forall p\in\mathcal{P}|L_p\geq 2, r\in\mathcal{R}_p, d = D_p^{A0}+1,...,D_p^{A1}+L_p-1 \tag{3.9}$$

where

$$Q_1(\boldsymbol{x},\omega) = \text{Min}\sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}_p}\sum_{i\in\mathcal{D}_p^A}\sum_{d\in\widetilde{\mathcal{D}}_{pi\omega}} C_{pr}x_{pri,i+L_p-1} + \sum_{r\in\mathcal{R}^D}\sum_{d\in\mathcal{D}} W^{RG}b_{rd\omega} \tag{3.10}$$
$$+ \sum_{r\in\mathcal{R}}\sum_{d\in\mathcal{D}} W^{OP}z_{rd\omega}$$

Table 3.1 – Notation used for the $SPAS_{\boldsymbol{SB}}$ model.

Symbol	Description
Sets	
$\mathcal{P}$	Set of patients ($p = 1, \cdots, \lvert\mathcal{P}\rvert$)
$\mathcal{D}$	Set of days ($i, d = 1, \cdots, \lvert\mathcal{D}\rvert$)
$\mathcal{R}$	Set of rooms ($r = 1, \cdots, \lvert\mathcal{R}\rvert$)
Ω	Set of scenarios ($\omega = 1, ..., \lvert\Omega\rvert$)
$\mathcal{P}^M \subset \mathcal{P}$	Set of male patients
$\mathcal{P}^F \subset \mathcal{P}$	Set of female patients
$\mathcal{R}^D \subset \mathcal{R}$	Set of dependent rooms
$\mathcal{R}_p \subset \mathcal{R}$	Set of rooms can be assigned to patient p without violating the hard constraints HC3, HC4 and HC5
$\mathcal{D}_p^A \subset \mathcal{D}$	Set of admission dates of patient p ($d = D_p^{A0}, ..., D_p^{A1}$)
$\mathcal{D}_{pi}^L \subset \mathcal{D}$	Set of hospitalization dates of patient p who is admitted on day i ($d = i, ..., i + L_p - 1$)
$\widetilde{\mathcal{D}}_{pi\omega} \subset \mathcal{D}$	Set of overstay dates of patient p who is admitted on day i ($d = i + L_p, ..., i + L_p + \widetilde{L}_{p\omega} - 1$)
Parameters	
G_p	Gender of patient p (M or F)
B_r	Number of beds in room r
L_p	Length of stay of patient p
$\widetilde{L}_{p\omega}$	Length of overstay of patient p under scenario ω
C_{pr}	The penalty of assigning patient p to room r. The room penalties of SC2 (gender policy M, F, N), SC3 - SC5 are incorporated into the value
W^{RG}	Weight of gender policy constraint
W^{Tr}	Weight of transfers constraint
W^{OP}	Weight of overcrowding constraint
$\Pr(\omega)$	Probability of scenario ω
O_r	Maximum allowable number of overstay patients who exceed the capacity of room r
λ_{rd}^F, λ_{rd}^M	Positive numbers
First-stage decision variables	
α_{pi}	1 if patient p is admitted on day i, 0 otherwise
x_{prid}	1 if patient p is admitted on day i and assigned to room r on hospitalization day d, 0 otherwise
t_{pd}	1 if patient p is transferred on day d, 0 otherwise
Second-stage decision variables	
$u_{rd\omega}$	1 if there is at least one female patient in room r on day d under scenario ω, 0 otherwise
$b_{rd\omega}$	1 if there are both male and female patients in room r on day d under scenario ω, 0 otherwise
$z_{rd\omega}$	Number of patients exceeding the capacity of room r on day d under scenario ω

$$\text{s.t.} \quad z_{rd\omega} \geq \sum_{\substack{p\in\mathcal{P}|\\ r\in\mathcal{R}_p}} \sum_{i\in\mathcal{D}_p^A} \left[\mathbb{1}(d \in \mathcal{D}_{pi}^L) x_{prid} + \mathbb{1}(d \in \widetilde{\mathcal{D}}_{pi\omega}) x_{pri,i+L_p-1}\right] - B_r \tag{3.11}$$

$$\forall r \in \mathcal{R}, d \in \mathcal{D}, \omega \in \Omega$$

$$\lambda_{rd}^F(u_{rd\omega} + b_{rd\omega}) \geq \sum_{\substack{p\in\mathcal{P}^F|\\ r\in\mathcal{R}_p}} \sum_{i\in\mathcal{D}_p^A} \left[\mathbb{1}(d \in \mathcal{D}_{pi}^L) x_{prid} + \mathbb{1}(d \in \widetilde{\mathcal{D}}_{pi\omega}) x_{pri,i+L_p-1}\right] \tag{3.12}$$

$$\forall r \in \mathcal{R}^D, d \in \mathcal{D}, \omega \in \Omega$$

$$\lambda_{rd}^M(1 - u_{rd\omega} + b_{rd\omega}) \geq \sum_{\substack{p\in\mathcal{P}^M|\\ r\in\mathcal{R}_p}} \sum_{i\in\mathcal{D}_p^A} \left[\mathbb{1}(d \in \mathcal{D}_{pi}^L) x_{prid} + \mathbb{1}(d \in \widetilde{\mathcal{D}}_{pi\omega}) x_{pri,i+L_p-1}\right] \tag{3.13}$$

$$\forall r \in \mathcal{R}^D, d \in \mathcal{D}, \omega \in \Omega$$

$$b_{rd\omega} \in \{0,1\} \quad \forall d \in \mathcal{D}, r \in \mathcal{R}^D, \omega \in \Omega \tag{3.14}$$

$$u_{rd\omega} \in \{0,1\} \quad \forall d \in \mathcal{D}, r \in \mathcal{R}^D, \omega \in \Omega \tag{3.15}$$

$$z_{rd\omega} \in [0, O_r] \quad \forall d \in \mathcal{D}, r \in \mathcal{R}, \omega \in \Omega \tag{3.16}$$

In the first stage, the objective function (3.2) minimizes violations of the soft constraints. Each soft constraint has an associated penalty cost that is incurred if it is violated. The first part calculates the delayed admissions with respect to the earliest admission date, related to soft constraint SC1. The second part computes the cost of assigning patients to rooms, which is determined by the combined penalty of soft constraints SC2 (gender policy M, F, N), SC3, SC4 and SC5. The third part captures the cost associated with patient transfer related to soft constraint SC6. The last part of the function $Q_1(\boldsymbol{x}, \omega)$ is the second-stage recourse function. Constraint (3.3) ensures that each patient is admitted only once and within the range of admission dates. Constraint (3.4) ensures that each patient is assigned to a room for every day of their hospitalization based on the assigned admission date. Constraint (3.5) ensures that the number of patients assigned to each room does not exceed the room capacity. Constraint (3.6) ensures that the auxiliary variable t_{pd} becomes 1 if a patient changes room on two consecutive days. Constraints (3.7)-(3.9) define the domains of the first-stage variables.

In the second stage, the objective function (3.10) minimizes the expected costs, including penalties for assigning patients to rooms within the overstay period, penalties for overcrowding and for violation of gender policy. Constraint (3.11) refers to overcrowding constraint, which calculates the number of patients exceeding the capacity of each room for every day under each scenario. In this constraint, $\mathbb{1}(\cdot)$ is an indicator function that is

equal to 1 if condition $(\cdot)$ is satisfied, and 0 otherwise. Constraints (3.12) and (3.13) refer to gender policy constraints, where $\lambda^F_{rd} = \min\{O_r + B_r, |\mathcal{P}^F_{rd}|\}$ and $\lambda^M_{rd} = \min\{O_r + B_r, |\mathcal{P}^M_{rd}|\}$, $|\mathcal{P}^F_{rd}|$ and $|\mathcal{P}^M_{rd}|$ are the number of female/male patients who can be assigned to room r on day d without violating hard constraints HC3, HC4 and HC5. These constraints are formulated based on the work of [7]. Specifically, constraint (3.12) enforces female patient restrictions, and constraint (3.13) enforces male patient restrictions. Both constraints seek to avoid the assignment of two distinct genders to the same room, penalizing allocations where different genders share a room. Constraints (3.14)-(3.16) define the domains of the second-stage variables.

Suppose that there are N patients who need to be assigned in the given planning horizon, and each patient may need to spend a maximum number of M extra nights in the hospital. Then, the number of scenarios is $(M+1)^N$, where 1 represents that patients leave the hospital on the day of their planned discharge. If the number of patients is large, the number of scenarios will increase exponentially. Therefore, the $SPAS_{SB}$ model will become difficult to solve as it contains a large number of second-stage constraints and variables for these scenarios.

3.3 State-variable modeling and solution method

In this section, we propose a state-variable model $SPAS_{SV}$ to reformulate the second-stage SPAS problem by introducing a set of state variables, state transition constraints and linking constraints. To better introduce the $SPAS_{SV}$ model, we first reconsider the second-stage SPAS problem from the perspective of Markov Decision Process (MDP). Then, we present the $SPAS_{SV}$ model based on the definition of the given MDP. Finally, we propose a solution method to solve the $SPAS_{SV}$ model efficiently.

3.3.1 MDP perspective on the second-stage SPAS problem

As mentioned in Section 3.2.2, the second-stage SPAS problem is to accurately estimate the expected costs when the patient-room-admissionDay-hospitalizationDay assignment is given. The penalties for assigning patients to rooms within the overstay period can be observed individually for each patient. The penalties for overcrowding and violation of gender policy can be observed collectively for each room and each day. Since the overstay length of each patient is uncertain, it is difficult to directly observe the above

expected cost. To address this issue, we consider the second-stage SPAS problem from the viewpoint of MDP by defining the state and action of a specific (d, r) pair as follows.

Given a set of patients who may stay in room r on day d, we need to sequentially decide whether to reserve a bed for each patient. Note that all decisions that have been made at an earlier time are irrevocable, we can only observe the result at the end of the horizon. By doing so, we can frame the second-stage SPAS problem as a finite-horizon discrete-time MDP. In the following, we provide the basic elements of the MDP for the second-stage SPAS problem for room r on day d, including decision epochs, states, actions, transition, and costs.

Decision epochs

Let $\hat{\mathcal{P}}_{dr} = \{p | r \in R_p, i \in \mathcal{D}_p^A, d \in \mathcal{D}_{pi}^L \cup \hat{\mathcal{D}}_{pi}\}$ be the set of patients who may stay in room r on day d, where $\hat{\mathcal{D}}_{pi} = \{d | d = i + L_p, ..., i + L_p + \max_{\omega \in \Omega} \widetilde{L}_{p\omega} - 1\}$ is the set of days when patient p may overstay after he/she is admitted on day i. In the MDP, we make the decisions sequentially for patient $p \in \hat{\mathcal{P}}_{dr}$, following the ascending order of their indices. Let $\mathcal{T}_{dr} = \{0, 1, ..., |\hat{\mathcal{P}}_{dr}|\}$ be a set of discrete time steps and at each time step $\tau \in \mathcal{T}_{dr} \setminus \{|\hat{\mathcal{P}}_{dr}|\}$ we need to make a decision. Note that the final time step $\tau = |\hat{\mathcal{P}}_{dr}|$ does not require any decision but signifies the system's transition into the terminal state.

States

We define the state at time step τ, denoted as

$$s_\tau = \begin{cases} (p_\tau, n_\tau, g_\tau), & r \in \mathcal{R}^D \\ (p_\tau, n_\tau), & r \notin \mathcal{R}^D \end{cases} \tag{3.17}$$

where p_τ is the patient whose decision has been made in the last time step, n_τ is the number of occupied beds, and g_τ is the class of mixed-gender occupancy. The state space, $\mathcal{S}$, is therefore

$$\mathcal{S} = \begin{cases} \{(p_\tau, n_\tau, g_\tau) | \tau \in \mathcal{T}, p_\tau \in \hat{\mathcal{P}}_{dr}^0, n_\tau \in \mathcal{N}_{\tau dr}, g_\tau \in \mathcal{G}_{\tau dr}\}, & r \in \mathcal{R}^D \\ \{(p_\tau, n_\tau) | \tau \in \mathcal{T}, p_\tau \in \hat{\mathcal{P}}_{dr}^0, n_\tau \in \mathcal{N}_{\tau dr}\}, & r \notin \mathcal{R}^D \end{cases} \tag{3.18}$$

where $\hat{\mathcal{P}}_{dr}^0 = \hat{\mathcal{P}}_{dr} \cup \{0\}$, the index 0 introduces a dummy patient, meaning an initial state with no patients assigned to room r on day d. For time step $\tau = 0$, the dummy patient 0

is systematically assigned to each room on each day. $\mathcal{N}_{\tau dr} = \{0, 1, ..., \min(\tau, B_r + O_r)\}$ is the set of all possible values for the number of occupied beds used by any patient 0 to p_τ who has a bed reserved in room r on day d. $\mathcal{G}_{\tau dr} = \{None, \bigcup_{j=1}^{\tau} G_{p_j}\}$ is set of all possible room gender state for a dependent room r after assigning any subset of patients 0 to p_τ to that room on day d. The expression $\bigcup_{j=1}^{\tau} G_{p_j}$ can result in states including $OnlyM$, $OnlyF$, $Mixed$, corresponding to rooms occupied by male patients only, female patients only, or by both genders, respectively.

Actions

We use a_τ to define action to make the state transition process clear. At each time step $\tau \in \mathcal{T}_{dr} \setminus \{|\hat{\mathcal{P}}_{dr}|\}$, the decision requires to determine whether to reserve a bed for patient $p_{\tau+1}$ in room r on day d. We consider that the first-stage variables $x_{p_{\tau+1}rid}, \forall i \in \mathcal{D}^A_{p_{\tau+1}}$ represent that action, i.e., $a_\tau = \{x_{p_{\tau+1}rid} | i \in \mathcal{D}^A_{p_{\tau+1}}\}$. Since constraints (3.3) and (3.4) hold, we have $\sum_{i \in \mathcal{D}^A_{p_{\tau+1}}} x_{p_{\tau+1}rid} = 0$ or 1. Specifically, $\sum_{i \in \mathcal{D}^A_{p_{\tau+1}}} x_{p_{\tau+1}rid} = 0$ means that no bed is reserved for patient $p_{\tau+1}$ in room r on day d, while $\sum_{i \in \mathcal{D}^A_{p_{\tau+1}}} x_{p_{\tau+1}rid} = 1$ means to reserve a bed for patient $p_{\tau+1}$ in room r on day d.

Transition

Once the action a_τ and state s_τ are given, the only stochastic element in the transition to the next state $s_{\tau+1}$ is that whether patient $p_{\tau+1}$ stays in the hospital or not. We assume that the overstay length of each patient is stochastic with a known probability distribution. Let $\Pr(p, l)$ be the probability that patient p needs to stay at the hospital on l-th day after admission. Due to the state $s_\tau = (p_\tau, n_\tau)$ is a special case of the state $s_\tau = (p_\tau, n_\tau, g_\tau)$, we will present the state transition process for the latter. Consider that the system is currently in state $s_\tau = (p_\tau, n_\tau, g_\tau)$. Due to the fact that n_τ and g_τ are independent of each other, we can consider them separately. The state transition process for $n_\tau \to n_{\tau+1}$ follows Eq. (3.19). We can see that there are three possible transitions for $n_\tau \to n_{\tau+1}$: 1) not reserving a bed for patient $p_{\tau+1}$ in room r on the day d, then $n_{\tau+1} = n_\tau$; 2) reserving a bed for patient $p_{\tau+1}$ in room r on day d, and he/she still stays in the hospital, then $n_{\tau+1} = n_\tau + 1$; 3) reserving a bed for patient $p_{\tau+1}$ in room r on day d, but he/she has been discharged, then $n_{\tau+1} = n_\tau$.

$$n_{\tau+1} = \begin{cases} n_\tau, & \sum_{i \in \mathcal{D}^A_{p_{\tau+1}}} x_{p_{\tau+1}rid} = 0 \\ n_\tau + 1 \text{ with probability } \Pr(p_{\tau+1}, d-i+1), & x_{p_{\tau+1}rid} = 1 \\ n_\tau \text{ with probability } 1 - \Pr(p_{\tau+1}, d-i+1), & \end{cases} \tag{3.19}$$

The state transition process for $g_\tau \to g_{\tau+1}$ follows Eq. (3.20), which is similar to the state transition process for $n_\tau \to n_{\tau+1}$.

$$g_{\tau+1} = \begin{cases} g_\tau, & \sum_{i \in \mathcal{D}^A_{p_{\tau+1}}} x_{p_{\tau+1}rid} = 0 \\ G_{p_{\tau+1}} \cup g_\tau \text{ with probability } \Pr(p_{\tau+1}, d-i+1), & x_{p_{\tau+1}rid} = 1 \\ g_\tau \text{ with probability } 1 - \Pr(p_{\tau+1}, d-i+1), & \end{cases} \tag{3.20}$$

Cost

The cost associated with a given state-action pair $(s_\tau,\ a_\tau)$ includes the penalties for overcrowding and violating gender policy when reserving a bed for patient $p_{\tau+1}$ in room r on day d. It is easy to know that these penalties are zero if $\sum_{i \in \mathcal{D}^A_{p_{\tau+1}}} x_{p_{\tau+1}rid} = 0$. Thus, we only need to consider $x_{p_{t+1}rid} = 1$. Let $W^{EOP}_{dr\tau ni}$ be the expected overcrowding penalty for reserving a bed for patient $p_{\tau+1}$ admitted on day i in room r on day d with n occupied beds. Its value can be computed as Eq. (3.21).

$$W^{EOP}_{dr\tau ni} = \begin{cases} W^{OP} \Pr(p_{\tau+1}, d-i+1), & n_\tau \geq B_r \\ 0, & n_\tau \leq B_r - 1 \end{cases} \tag{3.21}$$

Let $W^{ERG}_{dr\tau gi}$ be the expected cost of violating the gender policy for reserving a bed for patient $p_{\tau+1}$ admitted on day i in dependent room r on day d if mixed-gender occupancy is g. Its value can be computed as Eq. (3.22).

$$W^{ERG}_{dr\tau gi} = \begin{cases} W^{RG} \Pr(p_{\tau+1}, d-i+1), & (g_\tau = OnlyM \wedge G_{p_{\tau+1}} = F) \\ & \vee\ (g_\tau = OnlyF \wedge G_{p_{\tau+1}} = M) \\ 0, & \text{otherwise} \end{cases} \tag{3.22}$$

3.3.2 State-variable model

Based on the above definition, our state-variable model $SPAS_{SV}$ computes the second-stage objective by observing the state following the first-stage decisions over a number of time steps. In addition to the notations mentioned above, the following two sets are also used.

$\mathcal{G}_g^A$: set of all possible predecessor states immediately prior to state g in the state transition process, i.e., $\mathcal{G}_{None}^A = \{None\}, \mathcal{G}_{OnlyM}^A = \{None, OnlyM\}, \mathcal{G}_{OnlyF}^A = \{None, OnlyF\}$, $\mathcal{G}_{Mixed}^A = \{OnlyF, OnlyM, Mixed\}$.

$\mathcal{N}'_{\tau dr}$: set of all possible values for the number of occupied beds by any patients 0 to p_τ who has a bed reserved in room r, assuming that there is a bed available for patient $p_{\tau+1}$ on day d ($n = 0, 1, ..., \min(\tau, B_r + O_r - 1)$). Note that the difference between $\mathcal{N}_{\tau dr}$ and $\mathcal{N}'_{\tau dr}$ lies that we ignore adding the future patient $p_{\tau+1}$ in $\mathcal{N}_{\tau dr}$. On the contrary, we consider adding patient $p_{\tau+1}$ in $\mathcal{N}'_{\tau dr}$ to make sure that there will be a bed available for patient $p_{\tau+1}$ in room r on day d.

We introduce the following variables to denote the probability distribution of each state-action pair. We refer to these variables as state variables and index each variable by the corresponding state and action. Due to the fact that the state elements n_τ and g_τ are independent of each other, we consider them separately. According to Section 3.3.1, the state transition depends on the admission date of the next patient when reserving a bed for that patient. Thus, the state variables are accordingly indexed by the admission date.

$y^1_{dr\tau ni}$: The probability of the number of occupied beds being equal to n on day d in room r at time step τ, and we also reserve a bed in the same room for patient $p_{\tau+1}$ who is admitted on day i (i.e., $x_{p_{\tau+1},rid} = 1$).

$y^0_{dr\tau n}$: The probability of the number of occupied beds being equal to n on day d in room r at time step τ, and we do not reserve a bed in the same room for patient $p_{\tau+1}$ (i.e., $\sum_{i \in \mathcal{D}^A_{p_{\tau+1}}} x_{p_{\tau+1},rid} = 0$).

$q^1_{dr\tau gi}$: The probability of mixed-gender occupancy being g on day d in room r at time step τ, and we also reserve a bed in the same room for patient $p_{\tau+1}$ who is admitted on day i (i.e., $x_{p_{\tau+1},rid} = 1$).

$q^0_{dr\tau g}$: The probability of mixed-gender occupancy being g on day d in room r at time step τ, and we do not reserve a bed in the same room for patient $p_{\tau+1}$ (i.e., $\sum_{i \in \mathcal{D}^A_{p_{\tau+1}}} x_{p_{\tau+1},rid} = 0$).

To better understand the state variables $y^1_{dr\tau ni}$ and $y^0_{dr\tau n}$, Figure 3.1 provides an illustration of state transition (3.19) using these state variables for room r on day d. Specifically, each node in the figure represents a state, and each arrow line represents a state variable. The initial state is (0-0), and the final state is (*End*). Multiple arrow lines connect node (p_τ-n_τ) to node ($p_{\tau+1}$-$n_\tau+1$), where each arrow line represents a state variable $y^1_{dr\tau ni}$. The total number of arrow lines between the nodes (p_τ-n_τ) and ($p_{\tau+1}$-$n_\tau+1$) is equal to the number of days in the set $\mathcal{D}^A_{p_{\tau+1}}$. In addition, there are two types of arrow lines connecting the node (p_τ-n_τ) to the node ($p_{\tau+1}$-n_τ) including the arrow lines representing the state variable $y^1_{dr\tau ni}$ and the arrow lines representing the state variable $y^0_{dr\tau n}$. Thus, the total number of arrow lines between the nodes (p_τ-n_τ) and ($p_{\tau+1}$-n_τ) is equal to $|\mathcal{D}^A_{p_{\tau+1}}|+1$. Due to that no patient needs to be decided at time step $\tau = |\hat{\mathcal{P}}_{dr}|$, only one arrow line connects the node ($p_{|\hat{\mathcal{P}}_{dr}|}$-$n$) and the node (*End*) representing the state variable $y^0_{dr|\hat{\mathcal{P}}_{dr}|n}$.

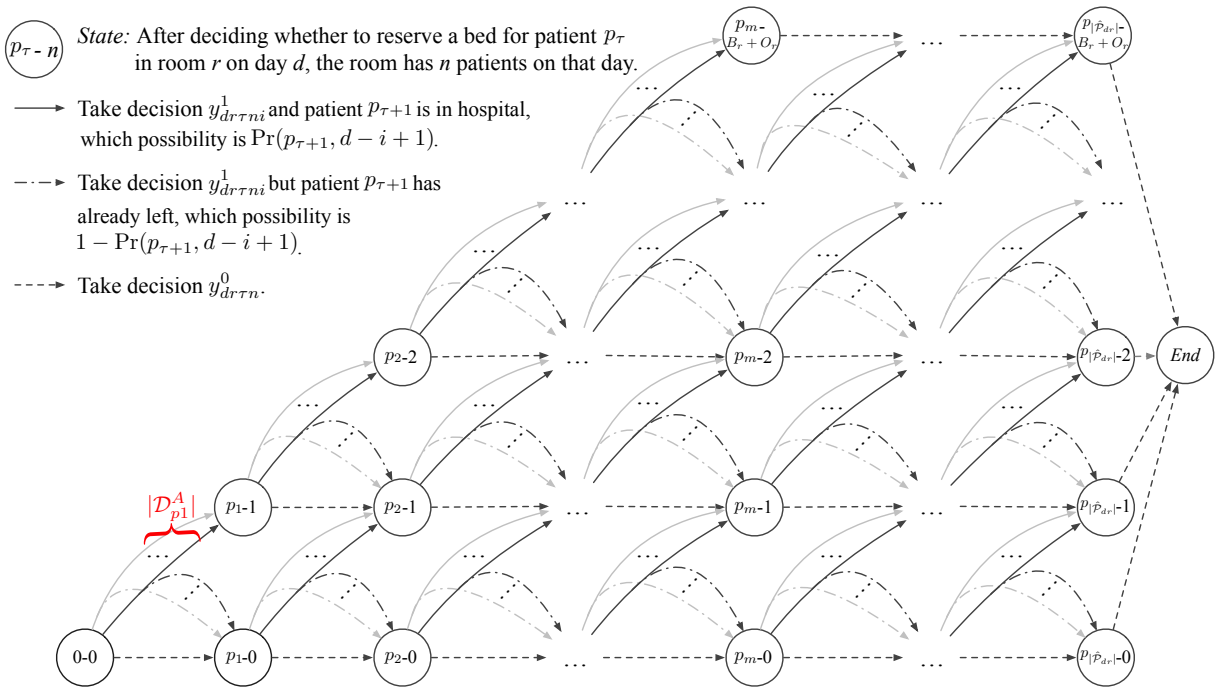


Figure 3.1 – An illustration of state transition (3.19) using state variables $y^1_{dr\tau ni}$ and $y^0_{dr\tau n}$.

It is clear that, for any state of the figure except the states (0-0) and (*End*), the sum of probabilities of all outgoing arrow lines is equal to the sum of probabilities of all incoming arrow lines. Therefore, we can write a state transition constraint that links the state variables of state (p_τ, n_τ) to the state variables of previous states $(p_{\tau-1}, n_{\tau-1})$ for room r on day d, as constraint (3.23). Both sides of the constraint (3.23) independently represent

the probability that the number of occupied beds at time step τ is equal to n in room r on day d. The left-hand side properly calculates this probability value by condition probability relations based on the probabilities $y^1_{dr,\tau-1,ni}$, $y^1_{dr,\tau-1,n-1,i}$ and $y^0_{dr,\tau-1,n}$. Moreover, the right-hand side calculates this probability value by using the probabilities $y^1_{dr\tau ni}$ and $y^0_{dr\tau n}$. According to the transition process (3.19), it is clear that the left-hand side of constraint (3.23) computes this probability correctly. Considering the actions that can be taken in the state (p_τ, n_τ), it is clear that the right-hand side of it also computes this probability correctly.

$$\begin{aligned}
&\sum_{i\in\mathcal{D}^A_{p_\tau}} \Big\{\mathbb{1}\left(n-1\in\mathcal{N}'_{\tau-1,dr}\right) y^1_{dr,\tau-1,n-1,i}\Pr(p_\tau, d-i+1)\\
&\qquad +\mathbb{1}\left(n\in\mathcal{N}'_{\tau-1,dr}\right) y^1_{dr,\tau-1,ni}\left[1-\Pr(p_\tau, d-i+1)\right]\Big\}\\
&+\mathbb{1}\left(n\in\mathcal{N}_{\tau-1,dr}\right) y^0_{dr,\tau-1,n} = \mathbb{1}\left(\tau\neq|\hat{\mathcal{P}}_{dr}|\wedge n\in\mathcal{N}'_{\tau dr}\right)\sum_{i\in\mathcal{D}^A_{p_{\tau+1}}} y^1_{dr\tau ni}+y^0_{dr\tau n}\\
&\qquad \forall d\in\mathcal{D}, r\in\mathcal{R}, \tau\in\mathcal{T}_{dr}\setminus\{0\}, n\in\mathcal{N}_{\tau dr} \qquad (3.23)
\end{aligned}$$

Figure 3.2 provides an example to illustrate the state transition (3.20) using the state variables $q^1_{dr\tau gi}$ and $q^0_{dr\tau g}$ for room r on day d. Note that this state transition diagram depends on the gender of each patient. For simplicity, we give an example where only patient p_1 is male ($G_{p_1} = M$), and all others are female ($G_{p_\tau} = F$, $\tau = 2, ..., |\hat{\mathcal{P}}_{dr}|$). Like Figure 3.1, each node in this figure represents a state, and each arrow line represents a state variable. The initial state is (0-*None*) and the final state is (*End*). The total number of arrow lines between the nodes (p_τ-g_τ) and ($p_{\tau+1}$-$G_{p_{\tau+1}}\cup g_\tau$) depends on the state $G_{p_{\tau+1}}\cup g_\tau$. For example, the number is equal to $2|\mathcal{D}^A_{p_{\tau+1}}|+1$ if $G_{p_{\tau+1}}\cup g_\tau = g_\tau$. The number of arrow lines connecting the node ($p_{|\hat{\mathcal{P}}_{dr}|}$-$g_{|\hat{\mathcal{P}}_{dr}|}$) and the node (*End*) is 1, which represents the state variable $q^0_{dr|\hat{\mathcal{P}}_{dr}|g}$.

Constraint (3.24) is the state transition constraint that links the state variables of state (p_τ, g_τ) to the state variables of previous states $(p_{\tau-1}, g_{\tau-1})$ for room r on day d. Both sides of constraint (3.24) independently represent the probability that mixed-gender occupancy at time step τ is g in room r on day d. The left-hand side properly calculates this probability value by condition probability relations based on the probabilities $q^1_{dr,\tau-1,gi}$, $q^1_{dr,\tau-1,g',i}$ and $q^0_{dr,\tau-1,g}$. Moreover, the right-hand side calculates this probability value by using the probabilities $q^0_{dr\tau g}$ and $q^1_{dr\tau gi}$. Considering the transition process (3.20) for state

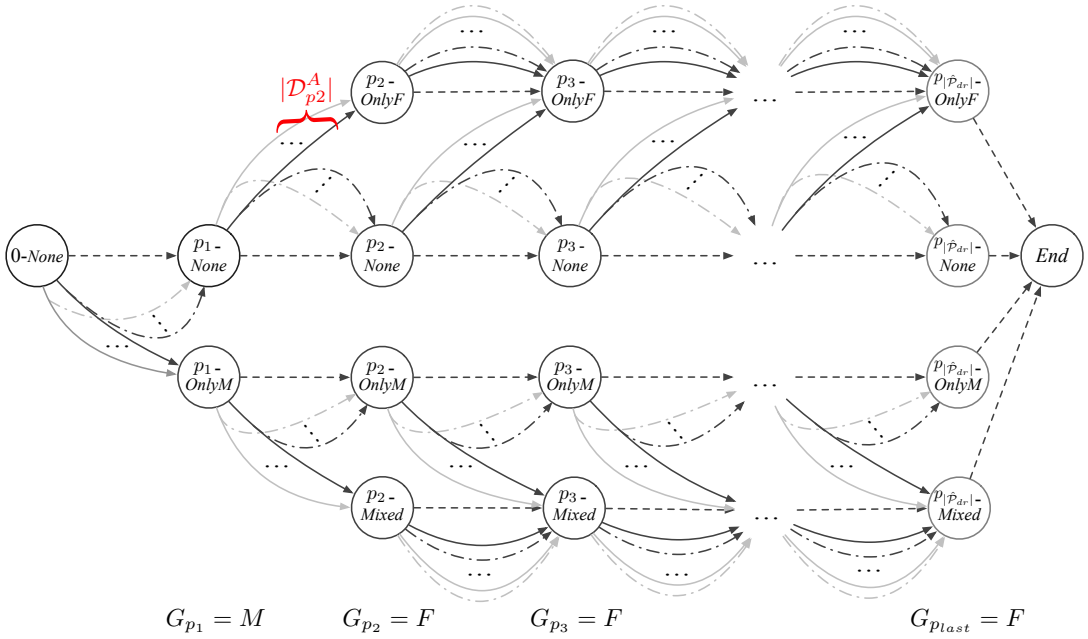


Figure 3.2 – An illustrative example of state transition constraint (3.20) using state variables $q^1_{dr\tau gi}$ and $q^0_{dr\tau g}$.

element g, it is clear that the left-hand side of constraint (3.24) computes this probability correctly. Considering those actions that can be taken in the state (p_τ, n_τ), it is clear that the right-hand side of it also computes this probability correctly.

$$\begin{aligned}
&\sum_{i\in\mathcal{D}^A_{p_\tau}} \Bigg\{ \sum_{g'\in\mathcal{G}^A_g | g'\in\mathcal{G}_{\tau-1,dr}} q^1_{dr,\tau-1,g'i} \Pr(p_\tau, d-i+1) \\
&\quad +\mathbb{1}\left(g\in\mathcal{G}_{\tau-1,dr}\right) q^1_{dr,\tau-1,gi} \left[1-\Pr(p_\tau, d-i+1)\right]\Bigg\} \\
&+\mathbb{1}\left(g\in\mathcal{G}_{\tau-1,dr}\right) q^0_{dr,\tau-1,s} = \mathbb{1}\left(\tau \neq |\hat{\mathcal{P}}_{dr}|\right) \sum_{i\in\mathcal{D}^A_{p_{\tau+1}}} q^1_{dr\tau gi} + q^0_{dr\tau g} \\
&\forall d\in\mathcal{D}, r\in\mathcal{R}^D, \tau\in\mathcal{T}_{dr}\setminus\{0\}, g\in\mathcal{G}_{\tau dr} \qquad (3.24)
\end{aligned}$$

Based on the above definition of the state variables and the state transition constraints,

we formulate the $SPAS_{SV}$ model as follows:

$$\textbf{\textit{SPAS}}_{\textbf{\textit{SV}}}\text{: Min } \sum_{p\in\mathcal{P}}\sum_{i\in\mathcal{D}_p^A} W^{De}(i-D_p^{A0})\alpha_{pi} + \sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}_p}\sum_{i\in\mathcal{D}_p^A}\sum_{d\in\mathcal{D}_{pi}^L} C_{pr}x_{prid} + \sum_{p\in\mathcal{P}|L_p\geq 2}\sum_{d\in\mathcal{D}_p} W^{Tr}t_{pd} + Q_2(\boldsymbol{x}) \tag{3.25}$$

s.t. Constraints (3.3) – (3.9)

where

$$Q_2(\boldsymbol{x}) = \text{Min } \sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}_p}\sum_{i\in\mathcal{D}_p^A}\sum_{d\in\hat{\mathcal{D}}_{pi}} \Pr(p,d-i+1)C_{pr}x_{pri,i+L_p-1} + \sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}}\sum_{\tau\in\mathcal{T}_{dr}\setminus\{|\hat{\mathcal{P}}_{dr}|\}}\sum_{n\in\mathcal{N}'_{\tau dr}}\sum_{i\in\mathcal{D}^A_{p_{\tau+1}}} W^{EOP}_{dr\tau ni}y^1_{dr\tau ni} + \sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}^D}\sum_{\tau\in\mathcal{T}_{dr}\setminus\{|\hat{\mathcal{P}}_{dr}|\}}\sum_{g\in\mathcal{G}_{\tau dr}}\sum_{i\in\mathcal{D}^A_{p_{\tau+1}}} W^{ERG}_{dr\tau gi}q^1_{dr\tau gi} \tag{3.26}$$

s.t. Constraints (3.23) and (3.24)

$$\sum_{n\in\mathcal{N}'_{\tau dr}} y^1_{dr\tau ni} = \mathbb{1}\left(d\in\mathcal{D}^L_{p_{\tau+1}i}\right)x_{p_{\tau+1}rid} + \mathbb{1}\left(d\in\hat{\mathcal{D}}_{p_{\tau+1}i}\right)x_{p_{\tau+1}ri,i+L_p-1} \tag{3.27}$$

$$\forall d\in\mathcal{D}, r\in\mathcal{R}, \tau\in\mathcal{T}_{dr}\setminus\{|\hat{\mathcal{P}}_{dr}|\}, i\in\mathcal{D}^A_{p_{\tau+1}}$$

$$\sum_{n\in\mathcal{N}_{\tau dr}} y^0_{dr\tau n} = 1 - \sum_{i\in\mathcal{D}^A_{p_{\tau+1}}}\left[\mathbb{1}\left(d\in\mathcal{D}^L_{p_{\tau+1}i}\right)x_{p_{\tau+1}rid} + \mathbb{1}\left(d\in\hat{\mathcal{D}}_{p_{\tau+1}i}\right)x_{p_{\tau+1}ri,i+L_p-1}\right] \tag{3.28}$$

$$\forall d\in\mathcal{D}, r\in\mathcal{R}, \tau\in\mathcal{T}_{dr}\setminus\{|\hat{\mathcal{P}}_{dr}|\}$$

$$\sum_{g\in\mathcal{G}_{\tau dr}} q^1_{dr\tau gi} = \mathbb{1}\left(d\in\mathcal{D}^L_{p_{\tau+1}i}\right)x_{p_{\tau+1}rid} + \mathbb{1}\left(d\in\hat{\mathcal{D}}_{p_{\tau+1}i}\right)x_{p_{\tau+1}ri,i+L_p-1} \tag{3.29}$$

$$\forall d\in\mathcal{D}, r\in\mathcal{R}^D, \tau\in\mathcal{T}_{dr}\setminus\{|\hat{\mathcal{P}}_{dr}|\}, i\in\mathcal{D}^A_{p_{\tau+1}}$$

$$\sum_{g\in\mathcal{G}_{\tau dr}} q^0_{dr\tau g} = 1 - \sum_{i\in\mathcal{D}^A_{p_{\tau+1}}}\left[\mathbb{1}\left(d\in\mathcal{D}^L_{p_{\tau+1}i}\right)x_{p_{\tau+1}rid} + \mathbb{1}\left(d\in\hat{\mathcal{D}}_{p_{\tau+1}i}\right)x_{p_{\tau+1}ri,i+L_p-1}\right] \tag{3.30}$$

$$\forall d\in\mathcal{D}, r\in\mathcal{R}^D, \tau\in\mathcal{T}_{dr}\setminus\{|\hat{\mathcal{P}}_{dr}|\}$$

$$y^0_{dr\tau n}\in[0,1] \quad \forall d\in\mathcal{D}, r\in\mathcal{R}, \tau\in\mathcal{T}_{dr}, n\in\mathcal{N}_{\tau dr} \tag{3.31}$$

$$y^1_{dr\tau ni}\in[0,1] \quad \forall d\in\mathcal{D}, r\in\mathcal{R}, \tau\in T_{dr}\setminus\{|\hat{\mathcal{P}}^0_{dr}|\}, i\in\mathcal{D}^A_{p_{\tau+1}}, n\in\mathcal{N}'_{\tau dr} \tag{3.32}$$

$$q^0_{dr\tau g}\in[0,1] \quad \forall d\in\mathcal{D}, r\in\mathcal{R}^D, \tau\in\mathcal{T}_{dr}, g\in\mathcal{G}_{\tau dr} \tag{3.33}$$

$$q^1_{dr\tau gi}\in[0,1] \quad \forall d\in\mathcal{D}, r\in\mathcal{R}^D, \tau\in\mathcal{T}_{dr}\setminus\{|\hat{\mathcal{P}}^0_{dr}|\}, i\in\mathcal{D}^A_{p_{\tau+1}}, g\in\mathcal{G}_{\tau dr} \tag{3.34}$$

The first stage of the $SPAS_{SV}$ model and the $SPAS_{SB}$ model is the same. In the second stage, the objective function (3.26) minimizes the expected cost of reserving beds for patients within the overstay period, as well as the penalties for overcrowding and for violation of gender policy. Constraints (3.27) and (3.28) are the linking constraints between first-stage variables $x_{\tau+1,rid}$ and the state-variables $y^1_{dr\tau ni}$ and $y^0_{dr\tau n}$. Constraint (3.27) implies that if $x_{\tau+1,rid}$ is equal to 1 (or 0), then the sum of left-hand side state variables $y^1_{dr\tau ni}$ must be equal to 1 (or 0). Constraint (3.28) enforces the sum of left-hand side state variables $y^0_{dr\tau n}$ must be equal to 1 (or 0) if the sum of $x_{\tau+1,rid}$ is equal to 0 (or 1). Similarly, constraints (3.29) and (3.30) are the linking constraints between first-stage variables $x_{\tau+1,rid}$ and the state-variables $q^1_{dr\tau gi}$ and $q^0_{dr\tau g}$. Constraint (3.29) implies that if $x_{\tau+1,rid}$ is equal to 1 (or 0), then the sum of left-hand side state variables $q^1_{dr\tau g}$ must be equal to 1 (or 0). Constraint (3.30) enforces the sum of left-hand side state variables $q^0_{dr\tau g}$ must be equal to 1 (or 0) if the sum of $x_{\tau+1,rid}$ is equal to 0 (or 1). Constraints (3.31) to (3.34) define the domains of the state variables.

3.3.3 Solution method for state-variable model

Although the $SPAS_{SV}$ model doesn't need to create variables for all possible scenarios, it is still difficult to solve the model directly due to a large number of variables and constraints. Considering that the SAA method can obtain a good approximate solution with relatively small sample scenarios [90], it is natural to apply the warm-starting strategy [117] which uses SAA to obtain an approximate solution as the initial solution of the $SPAS_{SV}$ model. Therefore, our solution method *SAA-SV* first solves the $SPAS_{SAA}$ model to obtain an approximate solution and then solves the $SPAS_{SV}$ model using the obtained solution as the initial solution.

General scheme

Algorithm 1 presents the general scheme of our proposed *SAA-SV* method. After initializing the necessary parameters, it generates a set of scenarios Ω^R. Then, it solves the $SPAS_{SAA}$ model within a time limit TL_{SAA} to obtain an approximate solution ($\boldsymbol{x}$, $\boldsymbol{t}$). To evaluate the objective value Obj of the solution ($\boldsymbol{x}$, $\boldsymbol{t}$) under the exponential number of scenarios, it fixes the values of the first-stage variables based on the solution ($\boldsymbol{x}$, $\boldsymbol{t}$) in the $SPAS_{SV}$ model and solves it. Once a better solution is found, the best solution ($\boldsymbol{x_{SAA}}, \boldsymbol{t_{SAA}}$) of the SAA procedure along with its objective value Obj_{SAA} are

updated. Since the optimal solution of the $SPAS_{SAA}$ model could be obtained before the time limit TL_{SAA} is reached, we utilize the remaining time TL_{SAA} - *elapsed_time* to search for new solutions using a different set of scenarios Ω^R and launch a new round SAA procedure. Given that it is hard to improve the solution quality significantly in the subsequent iterations, we set a maximum number of iterations $MaxIter$ for the SAA procedure. The above SAA iterative procedure is repeated until either the time limit TL_{SAA} or the maximum number of iterations $MaxIter$ is reached. Note that for $Iter > 0$, we use the best solution $(\boldsymbol{x_{SAA}}, \boldsymbol{t_{SAA}})$ as the initial solution of the new $SPAS_{SAA}$ model to accelerate the solution procedure. This is inspired from the iterative local search [118, 119, 120, 121]. Therefore, our proposed SAA process may differ from the traditional SAA procedure, referred to [122, 123, 124, 90]. After the SAA iterative procedure is completed, the $SPAS_{SV}$ model is solved using $(\boldsymbol{x_{SAA}}, \boldsymbol{t_{SAA}})$ as the initial solution within the remaining time. The above-mentioned *SAA-SV* algorithm can be considered as a new solution framework for solving two-stage stochastic programming problems, featured by first producing a high-quality initial solution in a relatively short time and using it as a good lower bound (i.e. maximum problem) for pruning the search tree [125].

Algorithm 1: Outline of the *SAA-SV* method for the SPAS problem

Input : A given problem instance NP, the time limit TL_{SAA} and TTL, the maximum number $MaxIter$, the number of scenarios $|\Omega^R|$.

Output : The best solution $(\boldsymbol{\alpha}^*, \boldsymbol{x}^*, \boldsymbol{t}^*)$ with objective value Obj^* found so far.

Initialize the best objective value in SAA process $Obj_{SAA} \leftarrow +\infty$, set the iteration counter $iter \leftarrow 0$, and the time limit for the current iteration $TL^{(iter)} \leftarrow TL_{SAA}$;
while *iter* $< MaxIter$ *and the elapsed time does not reach* TL_{SAA} **do**
  $\Omega^R \leftarrow$ ScenarioGeneration$(NP, |\Omega^R|)$;
  if $iter = 0$ **then**
    $(\boldsymbol{x}, \boldsymbol{t}) \leftarrow$ SolveModel($SPAS_{SAA}$, NP, Ω^R, $TL^{(iter)}$);
  else
    $(\boldsymbol{x}, \boldsymbol{t}) \leftarrow$ SolveModel($SPAS_{SAA}$, NP, Ω^R, $\boldsymbol{x_{SAA}}, \boldsymbol{t_{SAA}}$, $TL^{(iter)}$);
  $Obj \leftarrow$ Fix-and-Solve($SPAS_{SV}$, NP, $\boldsymbol{x}, \boldsymbol{t}$);
  if $Obj < Obj_{SAA}$ **then**
    $(\boldsymbol{x_{SAA}}, \boldsymbol{t_{SAA}}) \leftarrow (\boldsymbol{x}, \boldsymbol{t})$, $Obj_{SAA} \leftarrow Obj$;
  $iter \leftarrow iter + 1$, $TL^{(iter)} \leftarrow TL_{SAA} -$ elapsed_time;
$(\boldsymbol{\alpha}^*, \boldsymbol{x}^*, \boldsymbol{t}^*, Obj^*) \leftarrow$ SolveModel($SPAS_{SV}$, NP, $\boldsymbol{x_{SAA}}, \boldsymbol{t_{SAA}}$, $TTL -$ elapsed_time);
return $(\boldsymbol{\alpha}^*, \boldsymbol{x}^*, \boldsymbol{t}^*, Obj^*)$;

Scenario generation

The performance of the SAA procedure is dependent on the number of samples available [94]. Generally, more samples lead to a better approximation of the objective function, but also increase the computational efforts. Thus, scenario selection is crucial for obtaining a high-quality solution of the $SPAS_{SAA}$ model. We are inspired from the approach of [71] to use a relatively small number of scenarios to provide information regarding the overstay distributions. In detail, we generate a certain number of scenarios $|\Omega^R|$, where the first scenario uses the expected overstay for each patient. Let $\Pr(p,l)$ be the probability of patient p needing to stay at the hospital on l-th day after admission. Then, the expected overstay days of patient p is $\widetilde{L}_p = \arg\min\{l - L_p - 1 : \Pr(p,l) < 0.5\}$. The remaining scenarios are constructed by random sampling to capture the overstay distribution.

$SPAS_{SAA}$ model

The $SPAS_{SAA}$ model can be derived by replacing all scenarios with sampled scenarios Ω^R and substituting $\Pr(\omega)$ in the objective function (3.2) with the sampling probability. However, the search space of the $SPAS_{SAA}$ model is large due to the patient-room-admissionDay-hospitalizationDay assignment variables. By prohibiting patient transfer during their stay, we limit the search space by considering the patient-room-admissionDay assignment variables, resulting in a special case of the $SPAS_{SAA}$ model. Thus, to accelerate the solution procedure, we solve the $SPAS_{SAA}$ model without transfers constraint, which is similar to the work of [7, 8, 9].

Our $SPAS_{SAA}$ model is inherited from the $SPAS_{SB}$ model by removing the variables α_{pi} and t_{pd} and replacing x_{prid} by x_{pri}, a binary variable taking the value of 1 if patient p is admitted on day i and allocated to room r, and 0 otherwise. The corresponding objective function and constraints are also adjusted accordingly. Thus, the $SPAS_{SAA}$ model is formulated as follows:

$$\boldsymbol{SPAS}_{SAA}: \text{Min} \sum_{p\in\mathcal{P}} \sum_{r\in\mathcal{R}_p} \sum_{i\in\mathcal{D}_p^A} \left(W^{De}(i - D_p^{A0}) + L_p C_{pr}\right) x_{pri} + \frac{1}{|\Omega^R|} \sum_{\omega\in\Omega^R} Q_1'(\boldsymbol{x},\omega) \quad (3.35)$$

$$\text{s.t.} \quad \sum_{r\in\mathcal{R}_p} \sum_{i\in\mathcal{D}_p^A} x_{pri} = 1 \quad \forall p \in \mathcal{P} \quad (3.36)$$

$$\sum_{p\in\mathcal{P}|r\in\mathcal{R}_p} \sum_{i\in\mathcal{D}_p^A|d\in\mathcal{D}_{pi}^L} x_{pri} \le B_r \quad \forall r \in \mathcal{R}, d \in \mathcal{D} \quad (3.37)$$

$$x_{pri} \in \{0,1\} \quad \forall p \in \mathcal{P}, r \in \mathcal{R}_p, i \in \mathcal{D}_p^A \quad (3.38)$$

where

$$Q'_1(\boldsymbol{x},\omega) = \sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}_p}\sum_{i\in\mathcal{D}_p^A}\widetilde{L}_{p\omega}C_{pr}x_{pri} + \sum_{r\in\mathcal{R}^D}\sum_{d\in\mathcal{D}}W^{RG}b_{rd\omega} + \sum_{r\in\mathcal{R}}\sum_{d\in\mathcal{D}}W^{OP}z_{rd\omega} \tag{3.39}$$

s.t. Constraints (3.14) – (3.16), where Ω^R is used instead of Ω

$$z_{rd\omega} \geq \sum_{p\in\mathcal{P}|r\in\mathcal{R}_p}\sum_{i\in\mathcal{D}_p^A|d\in\mathcal{D}_{pi}^L\cup\widetilde{\mathcal{D}}_{pi\omega}} x_{pri} - B_r \quad \forall r\in\mathcal{R}, d\in\mathcal{D}, \omega\in\Omega^R \tag{3.40}$$

$$\lambda^F_{rd}(u_{rd\omega}+b_{rd\omega}) \geq \sum_{p\in\mathcal{P}^F|r\in\mathcal{R}_p}\sum_{i\in\mathcal{D}_p^A|d\in\mathcal{D}_{pi}^L\cup\widetilde{\mathcal{D}}_{pi\omega}} x_{pri} \quad \forall r\in\mathcal{R}^D, d\in\mathcal{D}, \omega\in\Omega^R \tag{3.41}$$

$$\lambda^M_{rd}(1-u_{rd\omega}+b_{rd\omega}) \geq \sum_{p\in\mathcal{P}^M|r\in\mathcal{R}_p}\sum_{i\in\mathcal{D}_p^A|d\in\mathcal{D}_{pi}^L\cup\widetilde{\mathcal{D}}_{pi\omega}} x_{pri} \quad \forall r\in\mathcal{R}^D, d\in\mathcal{D}, \omega\in\Omega^R \tag{3.42}$$

3.4 Computational experiments

In this section, we present computational results to demonstrate the effectiveness of the proposed $SPAS_{SB}$ and $SPAS_{SV}$ models as well as the SAA-SV method.

3.4.1 Instances design and experimental protocol

We carried out computational experiments on a total of 40 benchmark instances using the instance generator of [12]. The generator receives the number of departments, rooms, features, patients, and days as input parameters. It creates a random instance based on their predefined distributions concerning various features such as the length of stay, the room capacity, the number of specialisms, etc. Note that Ceschia & Schaerf [12] consider that the patients might need to spend one extra night in the hospital, which is a special case of our study. Thus, we moderately adjusted their code (`https://bitbucket.org/satt/PASU/`) to create instances that allow patients to overstay for more than one nights.

Table 3.2 shows the main characteristics of the benchmark instances. The instances are divided into 5 families, each of which is characterized by a different number of departments, rooms, equipment, patients, specialities and days. For each family, we generate 8 instances with different Demand-to-Supply Ratios (DSR) and different overstay lengths. The DSR is defined as the ratio of the total bed demand from all patients, excluding overstays, to the total number of beds available for all hospitalization days, which we set to be 40%, 50%, 60% and 70%. The overstay length is defined as the maximum number of days that a patient may overstay. We set the overstay length to be 1 and 2. Thus, we

have 40 instances in total. All the instances and the code are publicly available (https://github.com/NWPU-ORMS/SPAS).

Table 3.2 – Main characteristics of the benchmark instances.

Family of instances	Departments	Rooms	Equipment	Patients	Specialites	Days
Small short (S-S)	4	8	4	50	3	14
Small mid (S-M)	4	8	4	100	3	28
Small long (S-L)	4	8	4	200	3	56
Med short (M-S)	6	40	5	250	10	14
Med mid (M-M)	6	40	5	500	10	28

Table 3.3 lists the cost setting used to penalize soft constraint violations, which is the same as [12, 41, 108]. Moreover, the maximum allowable number of overstay patients who exceed the capacity of room r is set to be B_r. The value of B_r is provided by the instance generator, with values being 1, 2, 4, or 6 for each room.

Table 3.3 – Weights of the soft constraints.

Constraint	SC1	SC2	SC3	SC4	SC5	SC6	SC7
Weight	2	50	20	10	20	100	1

Our models were implemented and solved using Gurobi Optimizer 11.0.0 with its default parameter settings. Branch-and-cut (B&C) is the default algorithm of Gurobi to solve the MIP models. Experiments were run on a cluster with each node running Linux with Inter(R) Xeon(R) Gold 6226R 2.90GHz CPU and 256Gb RAM. The number of CPU cores used was set to be 1. All experiments were performed for a time limit of 3600 seconds. For SAA-SV method, we set the number of scenarios $|\Omega^R|$ to 10 by referring to [71]. The maximum number of iterations $MaxIter$ is set to 3 and the maximum running time to solve the $SPAS_{SAA}$ model is set to half of the total time limit.

3.4.2 Computational Results

As mentioned in Section 3.2.2, the number of scenarios increases exponentially with the number of patients, making the $SPAS_{SB}$ model difficult to solve. In our experiments, we found that all instances are intractable for the $SPAS_{SB}$ model due to insufficient memory errors. Thus, we do not report the computational results of the $SPAS_{SB}$ model. To simplify the expression, we refer to solving the $SPAS_{SV}$ model directly as the SV

method in the following. Moreover, we directly solved a relaxed $SPAS_{SV}$ model within a maximum computation time of 24 hours to obtain lower bounds, which we refer to as the RSV method.

Tables 3.4 and 3.5 report computational results obtained by RSV, SV and SAA-SV methods on instances with 1 day and 2 days overstay, respectively. Under the header "Ω", we give the approximate number of scenarios for each instance. Under the header "$Var.$" and "$Con.$", we present the number of variables and constraints in the $SPAS_{SV}$ model. For the RSV method, we record the optimal objective value which also serves as the best lower bound. For the SV and SAA-SV methods, we record the best objective value (Obj), the total computation time to find the best solution, the total computation time when Gurobi either proves the optimality or reaches the time limit, the best lower bound (LB). We compute the percentage gaps $Gap(\%) = (Obj - LB^*)/LB^* \times 100$ of the best objective value found by each method from the best lower bound LB^* among these three methods. The symbol "-" indicates that no feasible solution or no lower bound is found within the time limit.

Table 3.4 – Computational comparisons among RSV, SV and SAA-SV for instances with 1 day overstay (best solutions and best lower bounds in **bold**, proven optimal solutions in star*).

Instance	Ω	*Var.*	*Con.*	*RSV*	*SV*					*SAA-SV*				
					Obj	Time to Best	Time to End	*LB*	*Gap* (%)	*Obj*	Time to Best	Time to End	*LB*	*Gap* (%)
S-S-40		32,984	21,389	6,078.5	**6,180.3***	335.3	335.7	**6,180.3**	0.00	**6,180.3***	0.2	166.8	**6,180.3**	0.00
S-S-50		49,848	30,288	6,365.4	**6,596.0***	619.5	664.6	**6,596.0**	0.00	**6,596.0***	429.4	575.1	**6,596.0**	0.00
S-S-60	1.1×10^{15}	34,552	22,885	6,607.3	**6,616.7***	73.7	74.0	**6,616.7**	0.00	**6,616.7***	48.2	48.8	**6,616.7**	0.00
S-S-70		35,022	24,057	10,665.3	**10,828.5***	636.9	642.3	**10,828.5**	0.00	**10,828.5***	4.4	233.5	**10,828.5**	0.00
S-M-40		108,329	56,211	11,108.2	**11,125.1***	369.7	371.1	**11,125.1**	0.00	**11,125.1***	0.6	156.3	**11,125.1**	0.00
S-M-50		118,719	66,127	17,299.4	**17,376.9***	1,150.7	1,152.6	**17,376.9**	0.00	**17,376.9***	10.3	269.6	**17,376.9**	0.00
S-M-60	1.3×10^{30}	96,803	55,695	16,249.6	**16,455.6**	1,839.4	3,600.0	**16,382.8**	0.44	**16,455.6**	1,332.7	3,600.0	16,359.1	0.44
S-M-70		108,678	65,247	22,515.5	23,193.4	699.2	3,600.0	**22,700.9**	2.17	**22,984.3**	617.9	3,600.0	22,698.1	1.25
S-L-40		635,815	251,268	43,046.9	43,116.6	3,515.5	3,600.0	43,083.4	0.05	**43,095.5***	1,337.5	1,338.6	**43,095.5**	0.00
S-L-50		623,616	276,242	44,575.8	52,802.6	3,070.4	3,600.0	44,726.9	18.03	**45,093.8**	468.4	3,600.0	**44,737.4**	0.80
S-L-60	1.6×10^{60}	406,517	194,530	64,509.9	68,485.0	3,599.5	3,600.0	**64,879.7**	5.56	**66,330.0**	1,800.4	3,600.0	64,816.9	2.24
S-L-70		570,769	256,261	39,686.5	43,137.5	2,937.7	3,600.0	**39,871.2**	8.19	**40,350.4**	1,272.9	3,600.0	1,070.3	1.20
M-S-40		620,306	386,973	28,687.3	-	-	3,600.0	-	-	**29,050.1**	185.2	3,600.0	**28,792.7**	0.89
M-S-50		620,306	386,973	**43,411.5**	107,637.7	26.2	3,600.0	-	147.95	**44,029.0**	1,800.7	3,600.0	-	1.42
M-S-60	1.8×10^{75}	577,077	356,233	**52,500.7**	-	-	3,600.0	-	-	**53,131.0**	595.3	3,600.0	-	1.20
M-S-70		755,658	462,863	**44,714.4**	-	-	3,600.0	-	-	**45,899.4**	1,760.6	3,600.0	-	2.65
M-M-40		3,144,281	1,486,832	**82,670.9**	218,393.8	151.6	3,600.0	-	164.17	**83,351.7**	1,820.4	3,600.0	-	0.82
M-M-50		2,474,668	1,260,412	**105,120.4**	-	-	3,600.0	-	-	**106,411.2**	1,816.1	3,600.0	-	1.23
M-M-60	3.3×10^{150}	2,919,879	1,441,813	**99,437.1**	-	-	3,600.0	-	-	**102,299.9**	1,200.3	3,600.0	-	2.88
M-M-70		3,229,928	1,656,556	**88,747.3**	-	-	3,600.0	-	-	**90,514.3**	1,477.3	3,600.0	-	1.99

In terms of lower bounds, we find that the SAA-SV method and the SV method find the same best lower bounds for 12 instances. For other instances, the SAA-SV, SV and RSV methods finds best lower bounds for 5 instances, 10 instances, and 13 instances, respectively. In addition, the SV and SAA-SV methods fail to find lower bounds for 14

Table 3.5 – Computational comparisons among *RSV*, *SV* and *SAA-SV* for instances with 2 days overstay (best solutions and best lower bounds in **bold**, proven optimal solutions in star*).

Instance	Ω	*Var.*	*Con.*	*RSV*	*SV*					*SAA-SV*				
					Obj	Time to Best	Time to End	*LB*	*Gap* (%)	*Obj*	Time to Best	Time to End	*LB*	*Gap* (%)
S-S-40		38,265	23,770	6,260.1	**6,385.9***	16.9	20.2	**6,385.9**	0.00	**6,385.9***	203.9	204.5	**6,385.5**	0.00
S-S-50		56,765	33,612	6,503.4	**6,770.9***	1,599.8	1,600.1	**6,770.9**	0.00	**6,770.9***	1,561.1	1,561.4	**6,770.9**	0.00
S-S-60	7.2×10^{23}	40,640	25,866	6,894.5	**6,920.6***	98.5	98.9	**6,920.6**	0.00	**6,920.6***	4.6	83.2	**6,920.6**	0.00
S-S-70		40,367	27,083	10,990.4	**11,155.4***	329.1	399.0	**11,155.4**	0.00	**11,155.4***	345.7	370.1	**11,155.4**	0.00
S-M-40		127,104	63,189	11,521.4	**11,609.6***	414.0	547.1	**11,609.6**	0.00	**11,609.6***	371.2	372.3	**11,609.6**	0.00
S-M-50		137,506	74,289	18,109.1	**18,220.3***	497.0	498.2	**18,220.3**	0.00	**18,220.3***	397.3	398.0	**18,220.3**	0.00
S-M-60	5.2×10^{47}	112,564	62,504	17,095.5	17,327.6	3,503.4	3,600.0	**17,232.1**	0.55	**17,313.9**	611.2	3,600.0	17,226.9	0.47
S-M-70		126,577	73,235	23,588.0	24,816.0	3,601.0	3,600.0	23,784.6	4.25	**24,165.0**	2,063.6	3,600.0	**23,804.6**	1.51
S-L-40		739,778	275,602	45,174.7	45,250.6	3,138.9	3,600.0	45,218.0	0.04	**45,233.2***	2,225.4	2,297.1	**45,233.2**	0.00
S-L-50		722,985	306,415	46,326.7	55,144.8	2,886.0	3,600.0	**46,529.9**	18.51	**47,046.1**	1,797.1	3,600.0	-	1.11
S-L-60	2.7×10^{95}	467,628	215,592	67,428.9	73,062.5	3,231.3	3,600.0	**67,863.0**	7.66	**69,601.9**	3,287.9	3,600.0	67,699.5	2.56
S-L-70		654,485	282,130	**41,165.2**	-	-	3,600.0	-	-	**41,911.0**	767.0	3,600.0	1,081.2	1.81
M-S-40		834,050	459,281	29,554.0	37,777.2	3,464.5	3,600.0	**29,663.5**	27.35	**30,064.2**	309.2	3,600.0	-	1.35
M-S-50		703,440	427,317	**44,430.3**	109,754.9	30.5	3,600.0	-	147.03	**45,130.7**	1,757.0	3,600.0	-	1.58
M-S-60	1.9×10^{119}	659,295	397,177	54,156.4	60,482.6	3,088.2	3,600.0	**54,407.1**	11.17	**55,081.4**	1,117.3	3,600.0	54,311.2	1.24
M-S-70		849,857	508,277	46,105.2	56,722.3	2,734.8	3,600.0	**46,229.7**	22.70	**47,509.3**	1,565.0	3,600.0	-	2.77
M-M-40		3,631,575	1,657,379	**85,576.5**	229,843.6	243.7	3,600.0	-	168.58	**86,698.4**	1,220.5	3,600.0	-	1.31
M-M-50		2,839,908	1,403,393	**108,832.2**	-	-	3,600.0	-	-	**110,687.5**	863.2	3,600.0	-	1.70
M-M-60	3.6×10^{238}	3,347,442	1,603,485	**99,516.8**	-	-	3,600.0	-	-	**108,930.2**	1,826.0	3,600.0	-	9.46
M-M-70		3,698,436	1,848,135	**91,763.7**	-	-	3,600.0	-	-	**95,413.3**	1,828.3	3,600.0	-	3.98

and 15 instances, respectively. Hence, it is useful to solve the relaxed $SPAS_{SV}$ model for discovering better lower bounds.

In terms of best objective values, the *SAA-SV* method dominates the *SV* method. Specifically, *SAA-SV* succeeds in finding feasible solutions for all 40 instances, while the *SV* method fails for 10 instances. These two methods find the same best solutions for 13 instances, of which 12 optimal solutions are attained. For the remaining 27 instances, *SAA-SV* finds better solutions with 2 proven optimal solutions. Moreover, as the size of the instances increases, the obtained gaps range from 0.00% to 168.58% for *SV* and from 0.00% to 9.46% for *SAA-SV*. It is worth noting that in Table 3.4, *SAA-SV* finds high-quality feasible solutions with an average optimality gap of 1.73% for instances with 500 patients and 3.3 $\times10^{150}$ scenarios.

To conclude, these results demonstrate that our proposed *SAA-SV* method is quite effective in terms of both solution quality and computational time.

3.4.3 Effect of stochasticity

We use the Value of Stochastic Solution (VSS) to measure the value gained by considering uncertain information when solving the problem with known distributions of random parameters. To compute the VSS, we first need to define the Expected Value Problem (EVP). Specifically, we replace the random overstay lengths with the expected

values to create the scenario for the EVP. Moreover, as we mentioned in Section 3.2.1, the SPAS problem aims to minimize a weighted sum of all the penalties of the unsatisfied soft constraints SC1-SC7. However, the weight W^{OP} of the overcrowding constraint (SC7) is only set to 1, which makes the overcrowding constraint less important than the other soft constraints. Thus, we also consider the weight of the W^{OP} to be 1, 10, 100 and 1000 to measure the effect of stochasticity.

Tables 3.6 and 3.7 present the objective values of EVP and $SAA\text{-}SV$. To compute Obj_{EVP}, we solve the EVP and use the obtained solution ($\boldsymbol{x}$, $\boldsymbol{t}$) in the $SPAS_{SV}$ model to obtain the objective value under the exponential number of scenarios. Moreover, the $Obj_{SAA\text{-}SV}$ is the best objective value of the $SAA\text{-}SV$ method. Moreover, the VSS value is computed as $VSS = (Obj_{EVP} - Obj_{SAA\text{-}SV})/Obj_{EVP} \times 100$. The symbol "-" indicates that the solution ($\boldsymbol{x}$, $\boldsymbol{t}$) obtained by solving the EVP model is infeasible in the $SPAS_{SV}$ model.

Table 3.6 – Computational results of EVP and $SAA\text{-}SV$ on instances with 1 day overstay with different W^{OP}.

Instance	$W^{OP}=1$			$W^{OP}=10$			$W^{OP}=100$			$W^{OP}=1000$		
	EVP	$SAA\text{-}SV$	VSS (%)	EVP	$SAA\text{-}SV$	VSS (%)	EVP	$SAA\text{-}SV$	VSS (%)	EVP	$SAA\text{-}SV$	VSS (%)
S-S-40	6,180.7	6,180.3	0.01	6,281.4	6,278.3	0.05	6,878.8	6,861.8	0.25	8,683.6	7,117.5	22.00
S-S-50	6,596.0	6,596.0	0.00	6,683.0	6,683.0	0.00	7,006.9	6,945.8	0.88	7,922.0	7,101.7	11.55
S-S-60	6,619.9	6,616.7	0.05	6,667.6	6,658.0	0.14	6,911.6	6,775.5	2.01	8,423.4	6,870.6	22.60
S-S-70	10,836.9	10,828.5	0.08	10,924.2	10,914.6	0.09	11,606.5	11,506.0	0.87	12,693.7	11,907.0	6.61
S-M-40	11,125.1	11,125.1	0.00	11,329.8	11,305.7	0.21	12,220.0	12,015.3	1.70	16,954.0	12,538.0	35.22
S-M-50	17,439.4	17,376.9	0.36	17,577.0	17,488.7	0.50	18,283.1	18,031.7	1.39	22,973.3	18,231.8	26.01
S-M-60	16,461.4	16,455.6	0.03	16,615.1	16,597.8	0.10	17,368.5	17,148.8	1.28	21,823.4	17,459.6	24.99
S-M-70	23,076.1	22,984.3	0.40	23,248.0	23,213.2	0.15	24,508.3	24,283.5	0.93	30,069.7	26,343.9	14.14
S-L-40	43,095.9	43,095.5	0.00	43,282.8	43,258.9	0.06	43,917.2	43,575.6	0.78	48,795.0	43,629.6	11.84
S-L-50	45,683.8	45,093.8	1.31	45,840.8	45,399.8	0.97	47,627.3	46,900.8	1.55	55,335.6	47,491.1	16.52
S-L-60	66,444.1	66,330.0	0.17	67,083.6	66,358.6	1.09	69,559.4	68,231.2	1.95	79,019.0	69,081.1	14.39
S-L-70	40,943.0	40,350.4	1.47	41,269.9	40,794.4	1.17	44,247.0	43,789.0	1.05	56,930.2	46,951.8	21.25
M-S-40	29,106.1	29,050.1	0.19	29,645.5	29,622.4	0.08	32,208.0	31,796.7	1.29	43,437.3	33,451.6	29.85
M-S-50	44,424.5	44,029.0	0.90	44,836.6	44,451.9	0.87	48,064.6	47,169.2	1.90	60,791.4	48,411.2	25.57
M-S-60	53,580.7	53,131.0	0.85	53,982.9	53,594.8	0.72	56,807.4	56,114.0	1.24	69,263.8	59,132.4	17.13
M-S-70	46,377.4	45,899.4	1.04	46,795.8	46,251.6	1.18	50,597.3	48,764.9	3.76	61,260.7	55,459.0	10.46
M-M-40	83,834.4	83,351.7	0.58	84,778.1	84,539.9	0.28	91,710.0	88,770.5	3.31	119,850.1	100,320.9	19.47
M-M-50	107,441.9	106,411.2	0.97	108,391.9	107,758.8	0.59	116,431.9	113,936.1	2.19	145,139.8	117,336.9	23.69
M-M-60	104,817.1	102,299.9	2.46	106,928.5	103,379.6	3.43	115,609.8	109,727.8	5.36	145,133.0	114,370.2	26.90
M-M-70	92,426.5	90,514.3	2.11	93,449.2	92,166.0	1.39	101,864.7	100,043.8	1.82	130,695.8	105,326.3	24.09
Average			0.65			0.65			1.78			20.21

From Table 3.6, we observe that as the weight W^{OP} increases, the VSS values also increase. Specifically, the average VSS value is 0.65%, 0.65%, 1.78% and 20.21% when the weight of the W^{OP} is 1, 10, 100 and 1000, respectively. From Table 3.7, we observe that the EVP leads to feasible solutions for 39 instances, while the $SAA\text{-}SV$ method can find feasible solutions for all instances. The reason for the solutions obtained by the

Table 3.7 – Computational results of EVP and $SAA\text{-}SV$ on instances with 2 days overstay with different W^{OP}.

Instance	$W^{OP}=1$			$W^{OP}=10$			$W^{OP}=100$			$W^{OP}=1000$		
	EVP	$SAA\text{-}SV$	VSS (%)	EVP	$SAA\text{-}SV$	VSS (%)	EVP	$SAA\text{-}SV$	VSS (%)	EVP	$SAA\text{-}SV$	VSS (%)
S-S-40	-	6,385.9	-	6,546.9	6,523.3	0.36	-	7,487.2	-	11,482.2	9,681.7	18.60
S-S-50	-	6,770.9	-	6,939.6	6,891.5	0.70	7,682.5	7,510.0	2.30	12,085.2	8,325.0	45.17
S-S-60	6,948.4	6,920.6	0.40	7,026.1	6,998.7	0.39	7,588.7	7,225.7	5.02	10,874.8	7,486.6	45.26
S-S-70	11,155.4	11,155.4	0.00	11,283.2	11,283.2	0.00	12,334.9	12,202.8	1.08	15,404.6	14,324.5	7.54
S-M-40	-	11,609.6	-	-	11,875.5	-	13,510.6	13,155.6	2.70	24,836.2	14,380.5	72.71
S-M-50	18,272.5	18,220.3	0.29	18,475.9	18,395.6	0.44	19,799.8	19,294.4	2.62	28,826.2	20,126.2	43.23
S-M-60	17,371.7	17,313.9	0.33	17,644.9	17,577.8	0.38	19,195.2	18,790.4	2.15	28,783.2	20,969.3	37.26
S-M-70	-	24,165.0	-	-	24,484.0	-	26,960.9	26,439.8	1.97	40,599.4	35,067.9	15.77
S-L-40	45,248.0	45,233.2	0.03	45,534.1	45,496.5	0.08	46,870.5	46,062.6	1.75	57,955.8	46,443.0	24.79
S-L-50	47,177.2	47,046.1	0.28	48,230.9	47,657.9	1.20	51,547.0	50,494.7	2.08	71,457.3	53,077.7	34.63
S-L-60	-	69,601.9	-	-	69,955.2	-	-	73,316.3	-	-	76,404.7	-
S-L-70	42,547.8	41,911.0	1.52	-	42,771.1	-	-	47,660.3	-	75,981.1	57,473.6	32.20
M-S-40	-	30,064.2	-	-	30,861.1	-	35,041.9	34,375.4	1.94	59,874.0	36,862.9	62.42
M-S-50	-	45,130.7	-	-	45,873.8	-	51,250.1	50,052.3	2.39	73,422.7	53,089.4	38.30
M-S-60	-	55,081.4	-	-	55,808.8	-	-	60,221.1	-	-	71,426.2	-
M-S-70	-	47,509.3	-	-	48,038.0	-	-	52,368.1	-	-	73,149.7	-
M-M-40	-	86,698.4	-	-	88,188.5	-	-	95,950.3	-	-	103,331.6	-
M-M-50	-	110,687.5	-	-	112,442.2	-	-	124,258.5	-	-	138,442.3	-
M-M-60	-	108,930.2	-	-	110,389.6	-	-	124,397.1	-	-	137,565.7	-
M-M-70	-	95,413.3	-	-	97,662.6	-	116,990.7	110,044.9	6.31	182,760.7	135,298.1	35.08
Average			0.41			0.44			2.69			36.64

EVP being infeasible in $SPAS_{SV}$ can be attributed to the fact that the EVP model only considers one scenario thus fails to provide enough information regarding all possible overstay days for each patient and leads to exceeding the capacity threshold in some scenarios. Moreover, the average VSS value for the instances where feasible solutions can be found with the EVP is 0.41%, 0.44%, 2.69% and 36.64% when the weight W^{OP} is 1, 10, 100 and 1000, respectively. This implies that the $SAA\text{-}SV$ method yields solutions that are significantly better than those obtained by the EVP method, especially when a large penalty is applied to the overcrowding risk.

3.4.4 Contribution of two models in the $SAA\text{-}SV$ method

In order to confirm the contribution of the $SPAS_{SAA}$ and $SPAS_{SV}$ models in the $SAA\text{-}SV$ method, we further identify the objective values and computational times obtained by each model. Tables 3.8 and 3.9 present the best objective Obj and the total computation time $Time$ to find the best solution of each model. Specifically, column SAA gives the results obtained by solving the $SPAS_{SAA}$ model, while column $SAA\text{-}SV$ gives the results obtained by additionally adding the $SPAS_{SV}$ model. For a fair comparison, the Obj reported under column $SPAS_{SAA}$ are the objective values evaluated by the $SPAS_{SV}$

model. The row labeled "#Better" indicates the number of instances for which better solutions are obtained by incorporating the $SPAS_{SAA}$ and $SPAS_{SV}$ models.

Table 3.8 – Contribution of two models in *SAA-SV* on instances with 1 day overstay with different W^{OP}.

Instance	$W^{OP}=1$				$W^{OP}=10$				$W^{OP}=100$				$W^{OP}=1000$			
	SAA		*SAA-SV*		*SAA*		*SAA-SV*		*SAA*		*SAA-SV*		*SAA*		*SAA-SV*	
	Obj	*Time*	*Obj*	*Time*	*Obj*	*Time*	*Obj*	*Time*	*Obj*	*Time*	*Obj*	*Time*	*Obj*	*Time*	*Obj*	*Time*
S-S-40	6,180.3	0.2	6,180.3	0.2	6,278.3	0.2	6,278.3	0.2	6,861.8	1.7	6,861.8	1.7	7,382.8	0.2	**7,117.5**	95.0
S-S-50	6,644.7	14.3	**6,596.0**	429.4	6,712.8	5.8	**6,683.0**	774.0	6,955.3	1.4	**6,945.8**	83.8	7,101.7	8.8	7,101.7	8.8
S-S-60	6,617.6	0.3	**6,616.7**	48.2	6,658.0	0.2	6,658.0	0.2	6,779.0	3.2	**6,775.5**	292.9	7,335.6	0.3	**6,870.6**	1,655.5
S-S-70	10,828.5	4.4	10,828.5	4.4	10,914.6	16.9	10,914.6	16.9	11,539.1	15.4	**11,506.0**	488.6	11,975.0	22.9	**11,907.0**	155.4
S-M-40	11,125.1	0.6	11,125.1	0.6	11,306.6	0.4	**11,305.7**	139.7	12,015.3	5.4	12,015.3	5.4	13,199.8	6.6	**12,538.0**	3,130.2
S-M-50	17,376.9	10.3	17,376.9	10.3	17,491.1	0.7	**17,488.7**	279.7	18,046.5	0.8	**18,031.7**	2,319.5	18,314.4	5.0	**18,231.8**	3,600.6
S-M-60	16,461.8	24.3	**16,455.6**	1,332.7	16,605.7	17.6	**16,597.8**	1,742.5	17,153.4	24.7	**17,148.8**	3,366.5	17,576.1	25.3	**17,459.6**	3,006.4
S-M-70	22,984.3	617.9	22,984.3	617.9	23,213.2	874.8	23,213.2	874.8	24,283.5	745.2	24,283.5	745.2	28,180.6	982.6	**26,343.9**	2,338.0
S-L-40	43,095.6	3.5	**43,095.5**	1,337.5	43,262.8	3.1	**43,258.9**	1,876.9	43,600.6	18.7	**43,575.6**	1,276.9	43,629.6	10.7	43,629.6	10.7
S-L-50	45,093.8	468.4	45,093.8	468.4	45,399.8	1,224.1	45,399.8	1,224.1	46,900.8	1,089.3	46,900.8	1,089.3	47,527.6	976.5	**47,491.1**	3,536.7
S-L-60	66,330.0	1,800.4	66,330.0	1,800.4	66,362.2	1,473.3	**66,358.6**	1,804.0	68,231.2	1,576.4	68,231.2	1,576.4	69,096.1	740.1	**69,081.1**	3,078.1
S-L-70	40,350.4	1,272.9	40,350.4	1,272.9	40,794.4	518.2	40,794.4	518.2	43,789.0	1,792.0	43,789.0	1,792.0	46,999.4	1,735.1	**46,951.8**	3,419.0
M-S-40	29,050.1	185.2	29,050.1	185.2	29,622.4	180.2	29,622.4	180.2	31,796.7	279.6	31,796.7	279.6	33,451.8	246.0	**33,451.6**	273.2
M-S-50	44,029.0	1,800.7	44,029.0	1,800.7	44,454.8	1,104.2	**44,451.9**	1,133.3	47,169.2	778.8	47,169.2	778.8	48,411.2	1,466.2	48,411.2	1,466.2
M-S-60	53,131.0	595.3	53,131.0	595.3	53,594.8	1,743.1	53,594.8	1,743.1	56,114.0	1,792.2	56,114.0	1,792.2	59,132.4	1,621.4	59,132.4	1,621.4
M-S-70	45,899.4	1,760.6	45,899.4	1,760.6	46,251.6	1,144.9	46,251.6	1,144.9	48,764.9	1,450.6	48,764.9	1,450.6	55,459.0	1,117.1	55,459.0	1,117.1
M-M-40	83,352.0	1,800.2	**83,351.7**	1,820.4	84,539.9	958.8	84,539.9	958.8	88,770.5	1,142.7	88,770.5	1,142.7	100,320.9	1,531.6	100,320.9	1,531.6
M-M-50	106,411.6	1,723.2	**106,411.2**	1,816.1	107,758.8	1,117.5	107,758.8	1,117.5	113,936.1	1,267.5	113,936.1	1,267.5	117,336.9	1,750.8	117,336.9	1,750.8
M-M-60	102,299.9	1,200.3	102,299.9	1,200.3	103,380.3	1,800.6	**103,379.6**	1,831.4	109,727.8	1,574.5	109,727.8	1,574.5	114,370.2	1,092.5	114,370.2	1,092.5
M-M-70	90,514.3	1,477.3	90,514.3	1,477.3	92,166.0	1,808.4	92,166.0	1,808.4	100,043.8	1,808.3	100,043.8	1,808.3	105,326.3	1,804.4	105,326.3	1,804.4
#Better	6				8				6				11			

Table 3.9 – Contribution of two models in *SAA-SV* on instances with 2 days overstay with different W^{OP}.

Instance	$W^{OP}=1$				$W^{OP}=10$				$W^{OP}=100$				$W^{OP}=1000$			
	SAA		*SAA-SV*		*SAA*		*SAA-SV*		*SAA*		*SAA-SV*		*SAA*		*SAA-SV*	
	Obj	*Time*	*Obj*	*Time*	*Obj*	*Time*	*Obj*	*Time*	*Obj*	*Time*	*Obj*	*Time*	*Obj*	*Time*	*Obj*	*Time*
S-S-40	6,403.5	3.2	**6,385.9**	203.9	6,523.9	0.2	**6,523.3**	68.7	7,531.9	0.2	**7,487.2**	704.1	11,514.0	0.2	**9,681.7**	3,493.9
S-S-50	6,798.3	17.1	**6,770.9**	1,561.1	6,926.0	8.6	**6,891.5**	811.9	7,510.0	7.3	7,510.0	7.3	8,736.6	19.1	**8,325.0**	3,349.7
S-S-60	6,920.6	4.6	6,920.6	4.6	7,006.0	3.8	**6,998.7**	141.8	7,247.7	0.3	**7,225.7**	1,998.4	7,753.7	0.3	**7,486.6**	2,944.4
S-S-70	11,183.5	8.6	**11,155.4**	345.7	11,309.0	9.3	**11,283.2**	333.3	12,230.9	21.3	**12,202.8**	2,136.6	14,414.5	12.7	**14,324.5**	1,835.0
S-M-40	11,611.0	0.9	**11,609.6**	371.2	11,875.5	1.0	11,875.5	1.0	13,223.0	1.1	**13,155.6**	3,600.5	15,262.9	20.4	**14,380.5**	3,605.8
S-M-50	18,230.1	5.7	**18,220.3**	397.3	18,447.2	0.4	**18,395.6**	400.5	19,325.3	7.6	**19,294.4**	2,484.2	20,683.5	0.7	**20,126.2**	3,604.8
S-M-60	17,316.4	38.7	**17,313.9**	611.2	17,585.3	22.3	**17,577.8**	3,377.9	18,943.5	46.6	**18,790.4**	1,994.5	22,127.3	12.5	**20,969.3**	3,601.8
S-M-70	24,165.4	1,801.1	**24,165.0**	2,063.6	24,489.6	1,474.2	**24,484.0**	2,267.4	26,464.3	1,299.8	**26,439.8**	2,744.5	35,526.9	276.0	**35,067.9**	2,786.8
S-L-40	45,233.4	20.6	**45,233.2**	2,225.4	45,501.0	4.7	**45,496.5**	3,582.9	46,108.8	21.8	**46,062.6**	2,536.7	46,449.0	41.1	**46,443.0**	3,093.1
S-L-50	47,046.1	1,797.1	47,046.1	1,797.1	47,657.9	1,375.1	47,657.9	1,375.1	50,494.7	1,088.8	50,494.7	1,088.8	53,077.7	781.0	53,077.7	781.0
S-L-60	69,620.7	1,592.9	**69,601.9**	3,287.9	69,955.2	1,700.6	69,955.2	1,700.6	73,316.3	1,210.0	73,316.3	1,210.0	76,475.7	1,552.4	**76,404.7**	3,384.2
S-L-70	41,911.0	767.0	41,911.0	767.0	42,771.1	822.6	42,771.1	822.6	47,660.3	960.6	47,660.3	960.6	57,473.6	1,602.6	57,473.6	1,602.6
M-S-40	30,064.2	309.2	30,064.2	309.2	30,861.1	118.9	30,861.1	118.9	34,375.4	373.9	34,375.4	373.9	36,862.9	1,488.9	36,862.9	1,488.9
M-S-50	45,130.4	1,757.0	45,130.7	1,757.0	45,873.8	1,062.6	45,873.8	1,062.6	50,052.3	1,313.8	50,052.3	1,313.8	53,089.4	1,564.6	53,089.4	1,564.6
M-S-60	55,081.4	1,117.3	55,081.4	1,117.3	55,812.7	1,800.2	55,808.8	1,811.2	60,222.4	1,054.2	**60,221.1**	1,825.0	71,426.2	782.8	71,426.2	782.8
M-S-70	47,509.3	1,565.0	47,509.3	1,565.0	48,038.0	1,800.9	48,038.0	1,800.9	52,368.1	1,800.1	52,368.1	1,800.1	73,149.7	1,801.5	73,149.7	1,801.5
M-M-40	86,698.4	1,220.5	86,698.4	1,220.5	88,424.9	882.9	**88,188.5**	1,079.7	95,950.3	1,676.5	95,950.3	1,676.5	103,331.6	1,802.3	103,331.6	1,802.3
M-M-50	110,687.5	863.2	110,687.5	863.2	112,442.2	1,803.9	112,442.2	1,803.9	124,258.5	1,721.9	124,258.5	1,721.9	138,442.3	1,803.3	138,442.3	1,803.3
M-M-60	108,930.6	1,442.0	108,930.2	1,826.0	110,389.6	1,646.2	110,389.6	1,646.2	124,397.1	394.5	124,397.1	394.5	137,565.7	594.8	137,565.7	594.8
M-M-70	95,925.0	1,805.1	**95,413.3**	1,828.3	97,662.6	1,808.0	97,662.6	1,808.0	110,044.9	1,540.7	110,044.9	1,540.7	135,298.1	1,816.4	135,298.1	1,816.4
#Better	10				9				9				10			

From Table 3.8, we observe that as the W^{OP} increases, the number of better solutions found by incorporated $SPAS_{SAA}$ and $SPAS_{SV}$ models also increases. Specifically, if the $SPAS_{SV}$ model is incorporated, it finds better solutions for 6 instances with $W^{OP}=1$, 8 instances with $W^{OP}=10$, 6 instances with $W^{OP}=100$, 11 instances with $W^{OP}=1000$. From Table 3.9, incorporating the $SPAS_{SAA}$ and $SPAS_{SV}$ models can find better solutions for 10 instances with $W^{OP}=1$ or 1000, and 9 instances with $W^{OP}=10$ or 100. It

is worth noting that solving the $SPAS_{SAA}$ model can obtain optimal solutions in some instances, such as S-S-40, S-S-70, S-M-40, S-M-50 in Table 3.8, and S-S-60 in Table 3.9. The reason can be concluded as follows: 1) The impact of the overstay risk is insignificant, which means the penalty for the overstay risk is not high enough to affect the solution quality. 2) The scenarios we sampled, which followed the approach of [71], are representative of capturing the uncertainty of patient overstays. Furthermore, with larger numbers of scenarios, the objective value of the SAA model tends to be a more accurate estimate of the true objective value [126]. Thus, although solving the $SPAS_{SAA}$ model can obtain high-quality solutions through sampling a small number of scenarios, the calculation of their objective function is inaccurate. Incorporating the $SPAS_{SV}$ model, which utilizes an exponential number of scenarios, can improve the accuracy of their objective values. In addition, as the size of the instances increases, the number of better solutions found by incorporating $SPAS_{SAA}$ and $SPAS_{SV}$ models decreases. This is because solving the $SPAS_{SV}$ model becomes difficult as the size of the instances increases, which is consistent with the results in Section 3.4.2. For these large-scale instances, the $SPAS_{SAA}$ model can provide a high-quality solution. Thus, it is useful to solve the $SPAS_{SAA}$ model first. In summary, the above results indicate that the proposed SAA-SV method can effectively combine the advantages of the $SPAS_{SAA}$ and $SPAS_{SV}$ models to improve the solution quality and computational efficiency.

3.5 Chapter conclusion

In this chapter, we studied a stochastic variant of the patient admission scheduling problem (SPAS), which aims to assign patients to rooms during their planned hospitalization periods while considering the uncertainty of the overstay days. We considered the SPAS problem to be a two-stage stochastic programming problem where the first stage assigns patients to rooms on their planned hospitalization days, and the second stage evaluates the expected costs resulting from patient overstay. To solve the SPAS problem, we first proposed a scenario-based model $SPAS_{SB}$, which evaluates the expected cost by enumerating all possible scenarios. However, it is difficult to produce a solution as the model size grows exponentially with the number of scenarios. To address this difficulty, we proposed its equivalent state-variable model $SPAS_{SV}$, which is derived by reformulating the second stage of the $SPAS_{SB}$ model by introducing a set of state variables, state transition constraints and linking constraints. To solve the $SPAS_{SV}$ model efficiently, we

elaborated a solution method SAA-SV where we solve the $SPAS_{SAA}$ model to generate a high-quality feasible solution and then use it as an initial solution to solve the $SPAS_{SV}$ model.

We conducted extensive computational experiments to evaluate the performance of the proposed models. First, we compared directly solving the $SPAS_{SB}$ model, the $SPAS_{SV}$ model, and using the SAA-SV method. To ensure the solution quality, we also solved the relaxed $SPAS_{SV}$ model to find better lower bounds. The results show that the SAA-SV method effectively improves the solution quality and computational time. Second, we measured the effect of stochasticity on the SPAS problem by comparing the solutions obtained by solving the EVP and the SAA-SV method. The results demonstrate that, especially when a large penalty is applied to the overcrowding risk, the implementation of the SAA-SV method yields solutions that are significantly better than those obtained by the EVP method. Third, we confirmed the contribution of the $SPAS_{SAA}$ and $SPAS_{SV}$ models in the SAA-SV method.

Chapter 4

Integrated proactive and reactive surgical case scheduling in flexible operating rooms under uncertainty

---

In this chapter, we present a study on the surgical case scheduling problem in flexible operating rooms (ORs) under uncertainty (SSFU), which consists of operational decisions of assignment (assign patients to OR blocks in a given time horizon) and sequencing (determine the start time of the assigned surgeries in each OR block) while considering the uncertainties associated with the durations of elective surgeries, the arrivals of emergency surgeries, and their durations. The challenge of handling the SSFU problem lies in the uncertainty of emergency surgery arrival and the trade-off between elective surgeries and emergency surgeries. To solve this problem, we adopt an integrated proactive and reactive strategy, where a proactive SSFU model is first solved to generate an initial elective surgery plan, and then a reactive SSFU model is solved to dynamically adjust the plan based on actual surgery durations and emergency arrivals. Moreover, we implement three mechanisms — reserving capacity, Break-In-Moment, and buffer — to improve the robustness of the plan. Extensive computational experiments were conducted to evaluate the performance of the proposed models and mechanisms. The content of this chapter is based on an article submitted to *Production and Operations Management*.

4.1 Introduction

To handle the trade-off between elective surgeries and emergency surgeries, three types of OR policy can be adopted: dedicated, flexible, and hybrid [14]. The dedicated policy consists of reserving one or more ORs each day to perform emergency surgeries. Conversely, the flexible policy allows elective and emergency surgeries to be performed in all ORs. Moreover, the hybrid policy is a combination of the dedicated and flexible policies. A few papers [54, 127, 128] provided a (partial) comparison between different policies. Generally speaking, the flexible policy can provide a better trade-off between the performance of elective and emergency surgeries. However, when adopting the flexible policy, inserting emergency surgeries can create disruptions to the elective surgeries, implying longer elective waiting times, OR overtime, cancellations, and rescheduling.

To mitigate the negative impact of inserting emergency surgeries, multiple policies can be considered in both advance scheduling and allocation scheduling. In the former, a possible policy is reserving capacity for emergency surgeries [50]. In the latter, two policies can be considered: distributing the completion times of elective surgeries in a multi-OR setting (referred to as Break-In-Moments, or BIMs) as evenly as possible [58], and leaving open slots or larger intervals between scheduled surgeries (referred as buffers, or breaks) [59]. Moreover, the buffers protect against unforeseen emergency surgeries, but can also protect against duration uncertainty [14]. It is worth noting that the buffers and reserving capacity are distinct policies since the buffers are spread out over the ORs and over time and can be variable in size, whereas the reserved capacity is a continuous number of OR slices. Therefore, reserving capacity and buffers should be considered simultaneously when solving the integration of advance scheduling and allocation scheduling problem. Despite their potential, there is a noticeable research gap, as no study has combined these mechanisms to solve the integration of advance scheduling and allocation scheduling problem under duration uncertainty and arrival uncertainty while adopting the flexible policy. This motivates our work to coordinate the use of these three mechanisms to improve the efficiency of ORs and increase patient satisfaction. Here, we refer to this problem as the surgical case scheduling problem in flexible ORs under uncertainty (SSFU).

The main contributions of our work are summarized as follows:

(1) We study the SSFU problem by adopting the integrated proactive and reactive strategies, where a proactive SSFU problem is first solved to generate an initial elective surgery plan, and then a reactive SSFU problem is solved to dynamically adjust the

plan based on actual surgery durations and emergency arrivals. To deal with the uncertainty of surgery durations and emergency arrivals in the proactive SSFU problem, we implement three policies — reserving capacity, BIM, and buffer — to improve the robustness of the plan.

(2) We formulate the proactive SSFU problem as a two-stage stochastic programming problem, where the first stage assigns elective surgeries to OR blocks and determines the start time of each surgery, and the second stage evaluates the quality of the obtained plan under different durations of elective/emergency surgeries. We propose a scenario-based model $SSFU_{PS}$ with the objective of minimizing the longest possible waiting time of emergency surgeries by inserting buffers, which is different from the traditional surgical case scheduling problem. We also propose a MIP model $SSFU_{RS}$ to formulate the proactive SSFU problem.

(3) We carried out extensive experimental analysis using challenging test instants with different characteristics. Extensive computational experiments demonstrate that our proposed models significantly outperform the traditional deterministic approach under different block layout strategies. Moreover, we discuss the impact of the buffer mechanism on the performance of the surgery plan, and analyze the parameter sensitivity of the buffer mechanism.

The next section describes the definition of the SSFU problem. Section 4.3 describes the solution method in detail. Section 4.4 presents the computational results of our proposed models. Section 4.5 draws the conclusions.

4.2 Problem description

SSFU concerns the OR planning and scheduling over a planning horizon of $\mathcal{D}$ days. Specifically, the OR capacity is divided into surgery blocks $\mathcal{B}$ over the planning horizon, which are assigned in advance. Each surgery block $b \in \mathcal{B}$ is dedicated to only one type of surgical specialty, and multiple blocks may be assigned to the same specialty. Moreover, each surgery block b has a pre-allocated time duration L, and allowed maximum overtime O. Note that for each surgery block, overtime and idle time will be penalized.

The OR capacity is shared between elective surgeries and random emergency surgeries. The waiting list of elective surgeries, denoted as $\mathcal{I}$, is given at the beginning of the planning horizon. Each elective surgery $i \in \mathcal{I}$ has a surgery type and can be assigned to any of the blocks dedicated to the corresponding surgery type during the planning horizon. The

surgery duration D_i of elective surgery i includes not only the surgery time but also set-up time, cleaning, etc., which is random and depends on the surgery type. Associated with each elective surgery, there is a cost for performing and postponing it. For each elective surgery $i \in \mathcal{I}$, we define a set of costs $C_{ib}(b = 1, 2, \ldots, |\mathcal{B}|)$ and C_i^P to represent the cost of performing elective surgery i in surgery block b and postponing it to the next planning horizon, respectively. We assume that $C_i^P > C_{ib}, \forall b \in \mathcal{B}$. At the start of the planning horizon, an initial surgery plan for elective surgeries is established. This plan comprises a set of elective surgeries to be scheduled within each surgery block, along with their corresponding start times.

Emergency surgeries arrive randomly, and their durations are also unknown. In addition, emergency surgeries can be performed in any open operating room. When an emergency surgery arrives, one must immediately reschedule the surgeries that have not yet been performed to accommodate it. Possible decisions include canceling or postponing one or more pre-scheduled elective surgeries. The waiting time of emergency surgeries (the time from the arrival to the start of the surgery), along with the tardiness (the time from the scheduled start time to the actual start time) and cancellation of elective surgeries, will be penalized as they directly impact patient satisfaction and OR efficiency.

The objective of SSFU is to find a plan while maximizing OR efficiency and patient satisfaction. In detail, we define two classes of objectives as a function of costs related to performing or postponing elective surgeries, costs related to OR overtime and total idle time, costs related to tardiness and cancellation of elective surgeries, and costs related to emergency waiting times. Note that the costs related to performing or postponing elective surgeries can be determined once the initial surgery plan is established. The rest of the costs are uncertain and can only be observed after the surgery plan has been executed.

4.3 Solution method

We adopt an integrated proactive and reactive strategy to solve the SSFU problem. Figure 4.1 illustrates the framework of our solution method. Specifically, it starts with a scenario generation procedure to generate the scenarios $\mathcal{S}$. Then, a proactive SSFU problem is solved to generate an initial elective surgery plan. Finally, reactive SSFU problems are solved to dynamically adjust the plan based on actual surgery durations and emergency arrivals. To deal with the uncertainty of surgery durations and emergency arrivals in the proactive SSFU problem, we implement three mechanisms — reserving

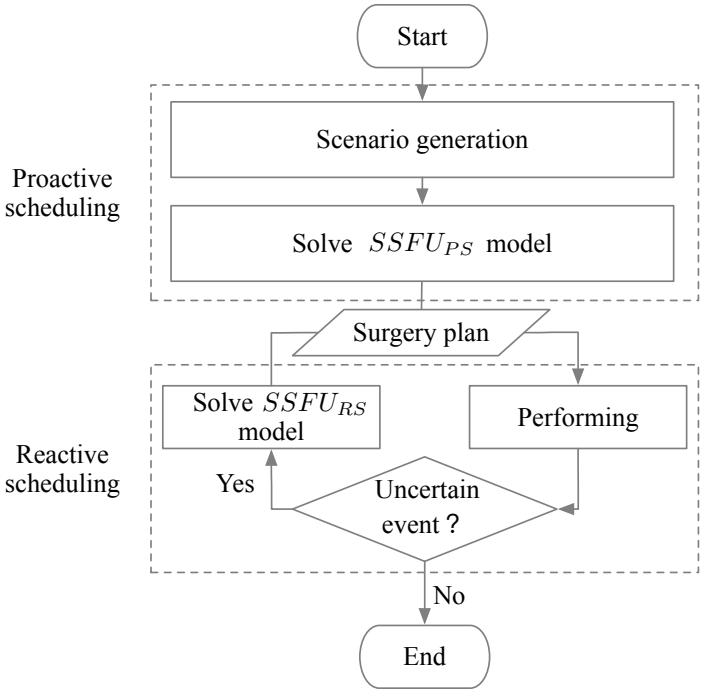


Figure 4.1 – Framework of solution method for the SSFU problem.

capacity, BIM, and buffer — to improve the robustness of the plan.

4.3.1 Proactive SSFU model

Proactive SSFU problem aggregates advance scheduling and allocation scheduling, which generates an elective surgery plan that assigns elective surgeries to OR blocks and determines the start time of each surgery while taking into account the stochastic surgery duration and unpredictable arrivals of emergency surgeries. An elective surgery plan must be determined before the exact number of emergencies and their arrival time are known. Consequently, a two-stage modeling framework is appropriate for the optimization model. Specifically, the first stage decisions are selecting elective surgeries from the waiting list, assigning them to the surgery blocks, and determining the start time of each selected surgery, while the second stage decisions focus on scheduling the emergency surgeries and canceling elective surgeries if necessary. The uncertainties in the optimization model are the capacity demand of emergency surgeries and the durations of elective surgeries. These two uncertainties are represented by scenarios, with each scenario being a complete realization of both aspects.

Considering that the emergency surgeries can arrive at any time during the day, and

must be performed immediately, the elective surgeries should be scheduled in a way that allows for the potential arrival of emergency surgeries. Thus, we propose to minimize the maximum possible waiting time of emergency surgeries, which is the worst-case waiting time of all emergency surgeries. To define the available moments for inserting emergency surgeries, we adopt the concept of break-in-moments (BIMs) [129]. Figure 4.2 illustrates the waiting time of emergency surgeries for a given surgery schedule, which involves seven elective surgeries scheduled in three ORs. Specifically, these moments include the start and end time of the occupied interval, as well as all finish times of surgeries within the occupied interval. The interval between two subsequent BIMs is a break-in-interval (BII). The interval between the end time of one surgery and the start time of the next is a buffer, represented by the shadowed block. The interval between the end time of the last surgery and the end time of the OR's operating time is considered as a slack time, represented by the gray block. Collectively, the BIMs, buffers, and slack constitute potential times when emergency surgeries can break into the surgery schedule. It is evident that the longest possible waiting time of emergency surgeries is the maximum length of BIIs. Therefore, minimizing the maximum length of BIIs is equivalent to minimizing the longest possible waiting time of emergency surgeries.

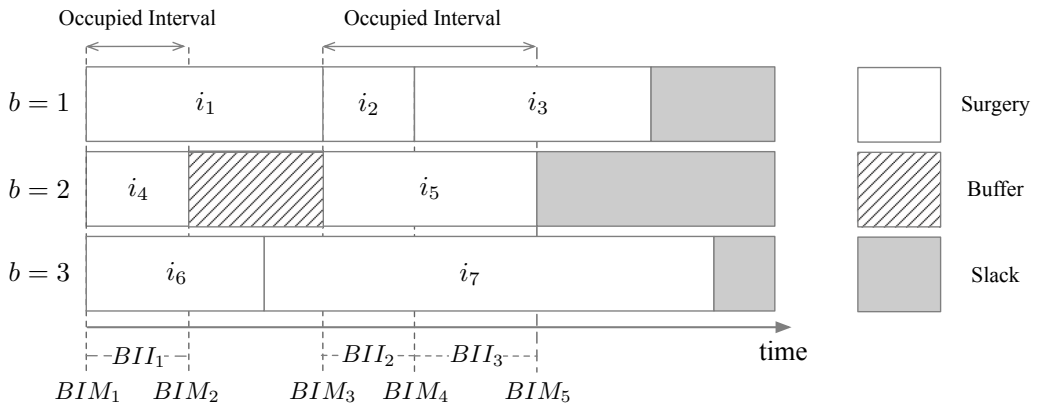


Figure 4.2 – Waiting time diagram for emergency surgeries in a specific surgery schedule.

To reduce the length of BIIs using an optimization model, a fundamental problem is how to calculate the length of each BII for a given surgery schedule. From Figure 4.2, we observe that if all ORs are occupied at the beginning of surgery i, a BII must exist, with its start time equal to the start time of surgery i. All BIIs can be calculated by the following Theorem 3.

Theorem 3 (The BII existence theorem) *Let $f_{i'}$ be the completion time of the surgery i', and z_i be the start time of surgery i. A BII exists whose length is equal to $f_{i'} - z_i$ if the following two conditions are satisfied:*

- *The ORs are fully occupied when surgery i starts, meaning that all ORs are either in use or beginning surgeries at time z_i.*
- *$f_{i'}$ is the closest completion time strictly later than z_i among all surgeries, i.e., $argmin_{i' \in \mathcal{I} | f_{i'} > z_i} f_{i'} - z_i$, including the case where $i' = i$.*

The proof of Theorem 3 is given in the Appendix A.3.

Note that using Theorem 3, we can calculate the length of one BII multiple times, if the start time of other surgeries is equal to the start time of surgery i which satisfies the conditions of Theorem 3. For example, in Figure 4.2, the length of BII_1 can be calculated by $f_4 - z_1$, $f_4 - z_2$, and $f_4 - z_3$. Based on Theorem 3, to identify whether all ORs are occupied at the beginning of elective surgery i, we define two types of binary variables $g_{ii'd}$ and $\phi_{ii'd}$. The first, $g_{ii'd} = 1$ if the start time of surgery i is no less than the start time of surgery i' in day d, and $g_{ii'd} = 0$ otherwise. The second, $\phi_{ii'd} = 1$ if the completion time of surgery i' is strictly greater than the start time of surgery i in day d, and $\phi_{ii'd} = 0$ otherwise. Figure 4.3 shows different possible cases of time overlap between surgeries i and i' and the corresponding values of $g_{ii'd}$ and $\phi_{ii'd}$. We can observe that both $g_{ii'd}$ and $\phi_{ii'd}$ are equal to 1 in cases b), c), and d), where these two ORs are occupied at the beginning of surgery i. In addition, we define variable $w_{ii'd} = 1$ if there exists a BII whose start time is equal to the start time of surgery i and finishes at the completion time of surgery i' on day d, and $w_{ii'd} = 0$ otherwise.

The notation used in $SSFU_{PS}$ is shown in Table 4.1, and the model is formulated as follows:

$$\textbf{SSFU}_{\textbf{PS}}\text{: } \min \sum_{b \in \mathcal{B}} \sum_{i \in \mathcal{I}_b} C_{ib} x_{ib} + \sum_{i \in \mathcal{I}} C_i^P \left(1 - \sum_{b \in \mathcal{B} | i \in \mathcal{I}_b} x_{ib}\right) + \sum_{b \in \mathcal{B}} C^O o_b^I + \sum_{d \in \mathcal{D}} C^I l_d + \sum_{d \in \mathcal{D}} C^W q_d^{MAX} + \sum_{s \in \mathcal{S}} P(s) Q(\boldsymbol{x}, s) \tag{4.1}$$

$$\text{s.t.} \quad \sum_{b \in \mathcal{B} | i \in \mathcal{I}_b} x_{ib} \leq 1 \quad \forall i \in \mathcal{I} \tag{4.2}$$

$$f_i \geq \sum_{b \in \mathcal{B} | i \in \mathcal{I}_b} \bar{D}_i x_{ib} \quad \forall i \in \mathcal{I} \tag{4.3}$$

Table 4.1 – Notation used for the $SSFU_{PS}$ model.

Symbol	Description
Sets	
$\mathcal{I}$	Set of elective surgeries ($i = 1, \cdots, \lvert\mathcal{I}\rvert$)
$\mathcal{B}$	Set of blocks ($b = 1, \cdots, \lvert\mathcal{B}\rvert$)
$\mathcal{D}$	Set of days ($d = 1, \cdots, \lvert\mathcal{D}\rvert$)
$\mathcal{S}$	Set of scenarios ($s = 1, \cdots, \lvert\mathcal{S}\rvert$)
$\mathcal{E}_{ds}$	Set of emergency surgeries arrived in day d under scenario s ($i = 1, \cdots, \lvert\mathcal{E}_{ds}\rvert$)
$\mathcal{I}_b \subseteq \mathcal{I}$	Set of elective surgeries who can be assigned to block b
$\mathcal{I}_d \subseteq \mathcal{I}$	Set of elective surgeries who can be assigned to day d
$\mathcal{B}_d \subseteq \mathcal{B}$	Set of blocks in day d
Parameters	
D_{is}	Surgery duration of elective surgery i under scenario s
$\bar{D}_i$	Average surgery duration of elective surgery i
$\bar{D}_s^E$	Average surgery duration of emergency surgery under scenario s
L	Regular time of each block
O	Maximum allowable amount of overtime
C_{ib}	Cost of performing surgery i in block b
C_i^P	Cost of postponing surgery i
C_i^Q	Cost of cancelling elective surgery i
C^O	Unit overtime cost for each block
C_b^I	Unit idle cost for each block b
C_i^W	Unit cost of waiting time for possible emergency surgery i
ϵ	A very small positive number
$P(s)$	Probability of scenario s
M	A very large positive number
First-stage Variables	
x_{ib}	1 if elective surgery i is assigned to block b, 0 otherwise
$\gamma_{ii'b}$	1 if surgery i and i' are operated in surgery block b while surgery i is operated after surgery i' , 0 otherwise
$g_{ii'd}$	1 if the start time of i is no less than the start time of i' in day d, 0 otherwise
$\phi_{ii'd}$	1 if the finish time of i is no less than the start time of i' in day d, 0 otherwise
$w_{ii'd}$	1 if there exists a BII whose start time is equal to the start time of surgery i and finishes at the completion time of surgery i' on day d, 0 otherwise
z_i	Start time of surgery i
f_i	Finish time of surgery i
o_b^I	The overtime of surgery block b
l_d	Total idle time in day d
v_b	End time of surgery block b
q_d^{MAX}	The maximum length of all BIIs in day d
Scecond-stage Variables	
u_{ibs}	1 if surgery i is cancelled in block b under scenario s, 0 otherwise
y_{bs}	Number of emergency surgeries are assigned to block b under scenario s
o_{bs}^{II}	Continuous variable for additional overtime of block b under scenario s

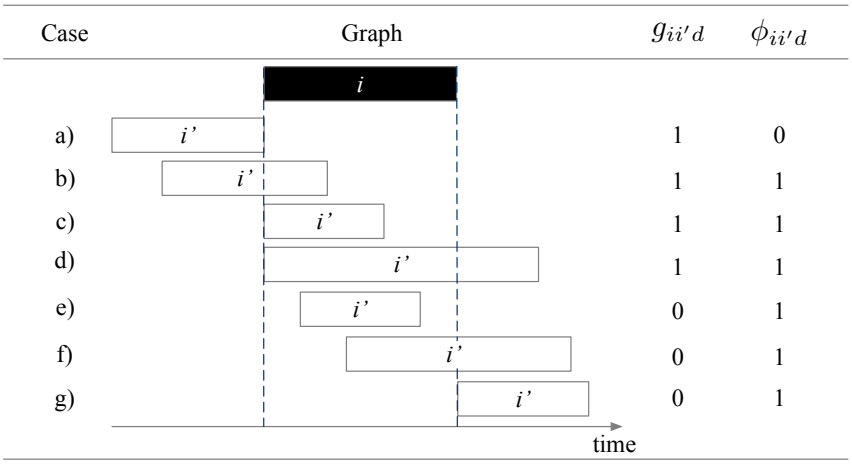


Figure 4.3 – Different possible overlap cases of surgery i with other surgeries i' which assigned to same day d.

$$f_i \leq M \sum_{b \in \mathcal{B} | i \in \mathcal{I}_b} x_{ib} \quad \forall i \in \mathcal{I} \tag{4.4}$$

$$f_i \geq f_{i'} + \bar{D}_i - M(3 - x_{ib} - x_{i'b} - \gamma_{ii'b}) \quad \forall b \in \mathcal{B}; i, i' \in \mathcal{I}_b | i < i' \tag{4.5}$$

$$f_{i'} \geq f_i + \bar{D}_{i'} - M(2 - x_{ib} - x_{i'b} + \gamma_{ii'b}) \quad \forall b \in \mathcal{B}; i, i' \in \mathcal{I}_b | i < i' \tag{4.6}$$

$$z_i = f_i - \sum_{b \in \mathcal{B} | i \in \mathcal{I}_b} \bar{D}_i x_{ib} \quad \forall i \in \mathcal{I} \tag{4.7}$$

$$v_b \geq f_i - M(1 - x_{ib}) \quad \forall d \in \mathcal{D}; b \in \mathcal{B}_d; i \in \mathcal{I}_d \tag{4.8}$$

$$l_d \geq \sum_{b \in \mathcal{B}_d} \left(v_b - \sum_{i \in \mathcal{I}_d} \bar{D}_b x_{ib} \right) \quad \forall d \in \mathcal{D} \tag{4.9}$$

$$o_b^I \geq v_b - L \quad \forall b \in \mathcal{B} \tag{4.10}$$

$$z_i - z_{i'} \leq M \left(2 - \sum_{b \in \mathcal{B}_d | i \in \mathcal{I}_b} x_{ib} - \sum_{b \in \mathcal{B}_d | i' \in \mathcal{I}_b} x_{i'b} + g_{ii'd} \right) \quad \forall d \in \mathcal{D}; i, i' \in \mathcal{I}_d | i \neq i' \tag{4.11}$$

$$z_{i'} - z_i - \epsilon \leq M \left(3 - \sum_{b \in \mathcal{B}_d | i \in \mathcal{I}_b} x_{ib} - \sum_{b \in \mathcal{B}_d | i' \in \mathcal{I}_b} x_{i'b} - g_{ii'd} \right) \quad \forall d \in \mathcal{D}; i, i' \in \mathcal{I}_d | i \neq i' \tag{4.12}$$

$$f_{i'} - z_i \leq M \left(2 - \sum_{b \in \mathcal{B}_d | i \in \mathcal{I}_b} x_{ib} - \sum_{b \in \mathcal{B}_d | i' \in \mathcal{I}_b} x_{i'b} + \phi_{ii'd} \right) \quad \forall d \in \mathcal{D}; i, i' \in \mathcal{I}_d | i \neq i' \tag{4.13}$$

$$z_i - f_{i'} + \epsilon \leq M \left(3 - \sum_{b \in \mathcal{B}_d | i \in \mathcal{I}_b} x_{ib} - \sum_{b \in \mathcal{B}_d | i' \in \mathcal{I}_b} x_{i'b} - \phi_{ii'd} \right) \quad \forall d \in \mathcal{D}; i, i' \in \mathcal{I}_d | i \neq i' \tag{4.14}$$

$$2g_{ii'd} \leq \sum_{b \in \mathcal{B}_d | i \in \mathcal{I}_b} x_{ib} + \sum_{b \in \mathcal{B}_d | i' \in \mathcal{I}_b} x_{i'b} \quad \forall d \in \mathcal{D}; i, i' \in \mathcal{I}_d | i \neq i' \tag{4.15}$$

$$2\phi_{ii'd} \leq \sum_{b\in\mathcal{B}_d|i\in\mathcal{I}_b} x_{ib} + \sum_{b\in\mathcal{B}_d|i'\in\mathcal{I}_b} x_{i'b} \quad \forall d\in\mathcal{D}; i,i'\in\mathcal{I}_d|i\neq i' \tag{4.16}$$

$$\sum_{i'\in\mathcal{I}_d} w_{ii'd} \geq \sum_{i'\in\mathcal{I}_d|i\neq i'} \left(g_{ii'd} + \phi_{ii'd} - \sum_{b\in\mathcal{B}_d|i'\in\mathcal{I}_b} x_{i'b} \right) - |\mathcal{B}_d| + 2 \quad \forall d\in\mathcal{D}; i\in\mathcal{I}_d \tag{4.17}$$

$$\sum_{i'\in\mathcal{I}_d} w_{ii'd} \leq \sum_{b\in\mathcal{B}_d|i\in\mathcal{I}_b} x_{ib} \quad \forall d\in\mathcal{D}; i\in\mathcal{I}_d \tag{4.18}$$

$$2w_{ii'd} \leq g_{ii'd} + \phi_{ii'd} \quad \forall d\in\mathcal{D}; i,i'\in\mathcal{I}_d|i\neq i' \tag{4.19}$$

$$q_d^{MAX} \geq f_{i'} - z_i - M(1-w_{ii'd}) \quad \forall d\in\mathcal{D}; i,i'\in\mathcal{I}_d \tag{4.20}$$

$$x_{ib}\in\{0,1\} \quad \forall b\in\mathcal{B}; i\in\mathcal{I}_b \tag{4.21}$$

$$g_{ii'd}, \phi_{ii'd}\in\{0,1\} \quad \forall d\in\mathcal{D}; i,i'\in\mathcal{I}_d|i\neq i' \tag{4.22}$$

$$w_{ii'd}\in\{0,1\} \quad \forall d\in\mathcal{D}; i,i'\in\mathcal{I}_d \tag{4.23}$$

$$z_i \in \left[0, L+O-\bar{D}_b\right] \quad \forall i\in\mathcal{I} \tag{4.24}$$

$$f_i\in[0,L+O] \quad \forall i\in\mathcal{I} \tag{4.25}$$

$$o_b^I\in[0,O] \quad \forall b\in\mathcal{B} \tag{4.26}$$

$$l_d\in[0,(O+L)|\mathcal{B}_d|] \quad \forall d\in\mathcal{D} \tag{4.27}$$

$$v_b\in[0,L+O] \quad \forall b\in\mathcal{B} \tag{4.28}$$

$$q_d^{MAX}\geq 0 \quad \forall d\in\mathcal{D} \tag{4.29}$$

Second-stage model:

$$Q(\boldsymbol{x},s) = \min \sum_{i\in\mathcal{I}}\sum_{b\in\mathcal{B}} C_i^Q u_{ibs} + \sum_{b\in\mathcal{B}} C^O o_{bs}^{II} \tag{4.30}$$

s.t.

$$\sum_{b\in\mathcal{B}_d} y_{bs} = |\mathcal{E}_{ds}| \quad \forall s\in\mathcal{S}; d\in\mathcal{D} \tag{4.31}$$

$$o_{bs}^{II} - o_b^I \geq \sum_{i\in\mathcal{I}_b} D_{is}(x_{ib}-u_{ibs}) + \bar{D}_s^E y_{bs} - L \quad \forall b\in\mathcal{B}_d; s\in\mathcal{S} \tag{4.32}$$

$$u_{ibs}\leq x_{ib} \quad \forall b\in\mathcal{B}, i\in\mathcal{I}_b, s\in\mathcal{S} \tag{4.33}$$

$$y_{bs}\in[0,|\mathcal{E}_{ds}|] \quad b\in\mathcal{B}_d; s\in\mathcal{S} \tag{4.34}$$

$$u_{ibs}\in\{0,1\} \quad \forall b\in\mathcal{B}; i\in\mathcal{I}_b; s\in\mathcal{S} \tag{4.35}$$

$$o_b^{II}\in[0,O] \quad \forall b\in\mathcal{B} \tag{4.36}$$

The first-stage objective function (4.1) minimizes the cost of performing and postponing surgeries, and the total cost related to OR overtime, idle time, surgeries waiting

times, and cancellation of scheduled surgeries, where $Q(\boldsymbol{x}, s)$ is the second-stage recourse function. Constraint (4.2) states that each elective surgery is performed at most in one block. Constraint (4.3) computes the completion time of each surgery. Constraint (4.4) ensures that a surgery is scheduled if and only if the surgery is assigned. Constraints (4.5) and (4.6) are either-or constraints, which determine the precedence among surgeries in the same block. Constraint (4.7) computes the start time of each surgery. Constraint (4.8) computes the completion time of each block. Constraint (4.9) computes the total idle time of each day. Constraint (4.10) computes the overtime of each block.

Constraints (4.11)-(4.20) are the BIMs constraints, which calculate the length of the BIIs. Constraints (4.11) and (4.12) ensure that if the start time of surgery i is no less than the start time of surgery i' on day d, then the variable $g_{ii'd}$ is set to 1. Constraints (4.13) and (4.14) ensure that if the completion time of surgery i is no less than the start time of surgery i' on day d, then the variable $\phi_{ii'd}$ is set to 1. Constraints (4.15) and (4.16) ensure that the variables $g_{ii'd}$ and $\phi_{ii'd}$ can measure the overlap between two surgeries if and only if these two surgeries are operated. Constraint (4.17) ensures that there must exist a BII whose start time equals the start time of surgery i if all ORs are occupied at the beginning of the surgery i. The proof of the validity of the constraint (4.17) is given in Appendix A.6. Constraint (4.18) ensures that there is at most one BII whose start time equals the start time of each surgery i. Constraint (4.19) states that for surgery i, if surgery i' overlaps with it and satisfies the overlaping cases b), c) and d) in Figure 4.3, then the variable $w_{ii'd}$ can be set to 1. Constraint (4.20) computes the largest length of BII. Note that by using constraints (4.11)-(4.20), we do not ensure that the completion time of surgery i' is the closest one to the start time of surgery i when $w_{ii'd} = 1$ for each feasible solution, but ensure that this objective is satisfied for the optimal solution. This method is used to avoid the numerical issue and accelerate the solving process. Parameter ϵ is used to avoid the numerical issue when the completion time of surgery i is equal to the start time of surgery i'. Finally, constraints (4.21)-(4.29) define the domain of the first-stage variables.

The second-stage objective function (4.30) minimizes the cost of canceling surgeries and OR overtime costs caused by emergency surgeries. Constraints (4.31) ensure that all emergency surgeries each day are assigned to the corresponding blocks. Constraint (4.32) computes the additional overtime of each block caused by emergency surgeries. Constraint (4.33) states that an elective surgery can be canceled if and only if the surgery is assigned. Constraints (4.34)-(4.36) define the domain of the second-stage variables.

Note that the $SSFU_{PS}$ model is developed by considering the trade-off between com-

plexity and accuracy. However, the model has the following limitations: (1) It does not consider the impact of the stochastic duration when determining the start time of surgeries. This omission may result in scheduled surgeries finishing earlier or later than planned; (2) It does not consider the impact of rescheduling surgeries to other surgery blocks when the emergency surgeries arrive. Therefore, the objective value of the $SSFU_{PS}$ model is an approximation of the actual cost, and the solution may be suboptimal. To address these limitations, we propose a simulation-optimization approach in the next section.

4.3.2 Reactive SSFU model

Elective surgeries are performed according to an initial plan on surgery day. Due to the uncertainty of the duration of elective surgeries, the arrival of emergency surgery, and its duration, five types of events can occur during the surgical process:

(1) *Surgery start*: surgery i starts at a certain time $\hat{Z}_i$.

(2) *Finish on time*: surgery i is finished at a certain time $\hat{F}_i$. This event conforms to the condition $0 \leq F_i - \hat{F}_i \leq MAD$, where F_i is the planned finish time and MAD is the maximum allowable deviation.

(3) *Early finish*: surgery i is finished earlier than the planned time F_i, adhering to the condition $F_i - \hat{F}_i > MAD$.

(4) *Late finish*: surgery i is still operating in the planned finish time F_i, adhering to the condition $\hat{F}_i > F_i$.

(5) *Emergency arrive*: an emergency surgery i arrives at a certain time A_i, and need to be schedule immediately.

The former two events are consistent with the surgery plan, while the last three events will deviate from the initial plan. Especially, the given surgery plan will be infeasible if the last two events occur. Thus, it is necessary to reschedule the surgeries that have not yet started when the last two events occur.

Once one of the above two events occurs at a certain time t in surgery day d, the rescheduling procedure is triggered. At time t, each surgery can be in one of three states: finished, ongoing, or not yet started. To reschedule the surgeries that have not yet started, we introduce the reactive SSFU model $SSFU_{RS}$, which can be derived from the $SSFU_{PS}$ model with some modifications made to the objective function and constraints. Compared to the $SSFU_{PS}$ model, we do not use the reserving capacity mechanism but only retain the BIM and buffer mechanisms in the $SSFU_{RS}$ model. Moreover, the $SSFU_{RS}$ model allows

for the rescheduling of elective surgeries to surgery blocks that are exclusively dedicated to the corresponding surgical types.

The $SSFU_{RS}$ model is formulated as a mixed-integer linear programming model, whose notation is given in Table 4.2.

Table 4.2 – Notation used for the $SSFU_{RS}$ model.

Symbol	Description
Sets	
$\mathcal{I}(t)$	Set of surgeries need to be scheduled at time t
$\mathcal{I}^o(t) \subseteq \mathcal{I}(t)$	Set of surgeries that are operating at time t
$\mathcal{I}^n(t) \subseteq \mathcal{I}(t)$	Set of elective surgeries that are not started at time t
$\mathcal{E}(t) \subseteq \mathcal{I}(t)$	Set of emergency surgeries that are waiting to start at time t
$\mathcal{I}_b(t) \subseteq \mathcal{I}(t)$	Set of unstarted surgeries which can be assigned to surgery block b at time t
Parameters	
Z_i	The planning start time of elective surgery i
F_i	The planning finish time of elective surgery i
B_i	The surgery block assigned to elective surgery i
A_i	The arrival time of emergency surgery i
$\widetilde{D}_i$	The possible duration of surgery i (Remaining duration for ongoing surgeries, total duration for not yet started surgeries)
C_i^d	Unit delay cost for surgery i
Decision Variables	
x_{ib}^R	1 if surgery i is assigned to surgery block b, 0 otherwise
z_i^R	The rescheduling start time of surgery i
f_i^R	The rescheduling finish time of surgery i
Auxiliary Variables	
o_b	The overtime of surgery block b
l	Total idle time
v_b	End time of surgery block b
$\pi_{ii'b}$	1 if surgery i and i' are operated in surgery block b while surgery i is operated after surgery i' , 0 otherwise
$g_{ii'}$	1 if the start time of i is no less than the start time of i', 0 otherwise
$\phi_{ii'}$	1 if the finish time of i is no less than the start time of i', 0 otherwise
$w_{ii'}$	1 if the finish time of surgery i' is the closest one and strictly later than the start time of surgery i, 0 otherwise
q^{MAX}	The maximum length of all BIIs

$$\textbf{\textit{SSFU}}_{\textbf{\textit{RS}}}\text{: } \min \sum_{i\in\mathcal{I}(t)} \sum_{b\in\mathcal{B}_d|i\in\mathcal{I}_b(t)} C_{ib}x_{ib} + \sum_{i\in\mathcal{I}(t)} C_i^Q \left(1 - \sum_{b\in\mathcal{B}_d|i\in\mathcal{I}_b(t)} x_{ib}\right) + \sum_{b\in\mathcal{B}_d} C^O o_b$$
$$+C^l l + C^W q^{MAX} + \sum_{i\in\mathcal{I}^n(t)} C_i^d|z_i - Z_i| + \sum_{i\in\mathcal{E}(t)} C^W\left(z_i - A_i\right) \tag{4.37}$$

s.t.
$$\sum_{b\in\mathcal{B}_d} x_{ib} = 1 \quad \forall i \in \mathcal{E}(t) \tag{4.38}$$
$$\sum_{b\in\mathcal{B}_d|i\in\mathcal{I}_b(t)} x_{ib} \leq 1 \quad \forall i \in \mathcal{I}^n(t) \tag{4.39}$$
$$x_{ib} = 1 \quad \forall i \in \mathcal{I}^o(t); b = B_i \tag{4.40}$$
$$\sum_{b\in\mathcal{B}_d|b\neq B_i} x_{ib} = 0 \quad \forall i \in \mathcal{I}^o(t) \tag{4.41}$$
$$f_i \geq \sum_{b\in\mathcal{B}_d|i\in\mathcal{I}_b(t)} \widetilde{D}_i x_{ib} + t \quad \forall i \in \mathcal{I}(t) \tag{4.42}$$
$$f_i \leq \sum_{b\in\mathcal{B}_d|i\in\mathcal{I}_b(t)} M x_{ib} + t \quad \forall i \in \mathcal{I}(t) \tag{4.43}$$
$$f_i \geq f_{i'} + \widetilde{D}_i - M(3 - \sum_{b\in\mathcal{B}_d|i\in\mathcal{I}_b(t)} x_{ib} - \sum_{b\in\mathcal{B}_d|i'\in\mathcal{I}_b(t)} x_{i'b} - \gamma_{ii'b}) \quad \forall b \in \mathcal{B}_d; i, i' \in \mathcal{I}_b(t)|i' < i \tag{4.44}$$
$$f_{i'} \geq f_i + \widetilde{D}_{i'} - M(2 - \sum_{b\in\mathcal{B}_d|i\in\mathcal{I}_b(t)} x_{ib} - \sum_{b\in\mathcal{B}_d|i'\in\mathcal{I}_b(t)} x_{i'b} + \gamma_{ii'b}) \quad \forall b \in \mathcal{B}_d; i, i' \in \mathcal{I}_b(t)|i' < i \tag{4.45}$$
$$z_i = f_i - \sum_{b\in\mathcal{B}_d|i\in\mathcal{I}_b(t)} \widetilde{D}_i x_{ib} \quad \forall i \in \mathcal{I}(t) \tag{4.46}$$
$$z_i = t \quad \forall i \in \mathcal{I}^o(t) \tag{4.47}$$
$$v_b \geq f_i - M\left(1 - x_{ib}\right) \quad \forall b \in \mathcal{B}_d; i \in \mathcal{I}_b(t) \tag{4.48}$$
$$l \geq \sum_{b\in\mathcal{B}_d} \left(v_b - t - \sum_{i\in\mathcal{I}^n(t)\cup\mathcal{E}(t)|b\in\mathcal{B}_d} \widetilde{D}_p x_{ib} - \sum_{i\in\mathcal{I}^o(t)|b=B_i} (f_i - t)\right) \tag{4.49}$$
$$o_b \geq f_i - L - M\left(1 - x_{ib}\right) \quad \forall b \in \mathcal{B}_d; i \in \mathcal{I}_b(t) \tag{4.50}$$
$$z_i^R - z_{i'}^R \leq M\left(2 - \sum_{b,b'\in\mathcal{B}_d|b\neq b'} (x_{ib} + x_{i'b'}) + g_{ii'}\right) \quad \forall i \in \mathcal{I}_b(t), i' \in \mathcal{I}_{b'}(t)|b \neq b' \tag{4.51}$$
$$z_{i'}^R - z_i^R - \epsilon \leq M\left(3 - \sum_{b,b'\in\mathcal{B}_d|b\neq b'} (x_{ib} + x_{i'b'}) - g_{ii'}\right) \quad \forall i \in \mathcal{I}_b(t), i' \in \mathcal{I}_{b'}(t)|b \neq b' \tag{4.52}$$
$$f_{i'}^R - z_i^R \leq M\left(2 - \sum_{b,b'\in\mathcal{B}_d|b\neq b'} (x_{ib} + x_{i'b'}) + \phi_{ii'}\right) \quad \forall i \in \mathcal{I}_b(t), i' \in \mathcal{I}_{b'}(t)|b \neq b' \tag{4.53}$$

$$z_i^R - f_{i'}^R + \epsilon \leq M\left(2 - \sum_{b,b' \in \mathcal{B}_d | b \neq b'} (x_{ib} + x_{i'b'}) + \phi_{ii'}\right) \quad \forall i \in \mathcal{I}_b(t), i' \in \mathcal{I}_{b'}(t) | b \neq b' \tag{4.54}$$

$$2g_{ii'} \leq \sum_{b,b' \in \mathcal{B}_d | b \neq b'} (x_{ib} + x_{i'b'}) \quad \forall i \in \mathcal{I}_b(t), i' \in \mathcal{I}_{b'}(t) | b \neq b' \tag{4.55}$$

$$2\phi_{ii'} \leq \sum_{b,b' \in \mathcal{B}_d | b \neq b'} (x_{ib} + x_{i'b'}) \quad \forall i \in \mathcal{I}_b(t), i' \in \mathcal{I}_{b'}(t) | b \neq b' \tag{4.56}$$

$$\sum_{i' \in \mathcal{I}(t)} w_{ii'} \leq \sum_{b \in \mathcal{B}_d | i \in \mathcal{I}_b(t)} x_{ib}, \quad \forall i \in \mathcal{I}(t) \tag{4.57}$$

$$\sum_{i' \in \mathcal{I}(t)} w_{ii'} \geq \sum_{i' \in \mathcal{I}(t) | i \neq i'} (g_{ii'} + \phi_{ii'} - \sum_{b \in \mathcal{B}_d | i' \in \mathcal{I}_b(t)} x_{i'b}) - |\mathcal{B}_d| + 2, \quad \forall i \in \mathcal{I}(t) \tag{4.58}$$

$$2w_{ii'} \leq g_{ii'} + \phi_{ii'} \quad \forall i, i' \in \mathcal{I}(t) | i \neq i' \tag{4.59}$$

$$q^{MAX} \geq f_{i'}^R - z_i^R - M(1 - w_{ii'}), \quad \forall i, i' \in \mathcal{I}(t) \tag{4.60}$$

$$x_{ib} \in \{0, 1\}, \quad \forall b \in \mathcal{B}_d, i \in \mathcal{I}_b(t) \tag{4.61}$$

$$z_i \in [t, L + O - \widetilde{D}_i], \quad \forall i \in \mathcal{I}(t) \tag{4.62}$$

$$f_i \geq t, \quad \forall i \in \mathcal{I}(t) \tag{4.63}$$

$$v_b \geq 0, \quad \forall b \in \mathcal{B}_d \tag{4.64}$$

$$o_b \in [0, O], \quad \forall b \in \mathcal{B}_d \tag{4.65}$$

$$l \in [0, (O + L)\mathring{\mathrm{u}}|\mathcal{B}_d|] \tag{4.66}$$

$$\pi_{ii'} \in \{0, 1\}, \quad \forall b \in \mathcal{B}_d, i, i' \in \mathcal{I}_b(t) | i < i' \tag{4.67}$$

$$g_{ii'}, \phi_{ii'} \in \{0, 1\}, \quad \forall i, i' \in \mathcal{I}(t) | i \neq i' \tag{4.68}$$

$$w_{ii'} \in \{0, 1\}, \quad \forall i, i' \in \mathcal{I}(t) \tag{4.69}$$

The objective function (4.37) minimizes the cost of performing and cancellation surgeries, and the expected total cost related to OR overtime, idle time, the waiting time for future and arrived emergencies, and tardiness of scheduled surgeries. Constraints (4.38) and (4.39) are the reassignment constraints. Constraint (4.38) ensures that emergency surgeries must be assigned in a surgery block. Constraint (4.39) states that each unstarted elective surgery can only be assigned to one surgery block. Constraints (4.40) and (4.41) ensure that the ongoing surgeries are operated exclusively within their assigned surgical blocks. Constraints (4.42)-(4.46) computes the start and finish time of each surgery, which are similar to constraints (5.7)-(5.10). Constraint (4.47) ensures that the ongoing surgeries are in progress at time t. Constraints (4.48)-(4.50) computes the idle time and overtime, which are similar to constraints (5.11)-(5.13). Constraints (4.51)-(4.60) are the

BIMs constraints. Finally, constraints (4.61)-(4.69) define the domain of the variables.

4.3.3 Discussion on the BIMs for surgical case scheduling

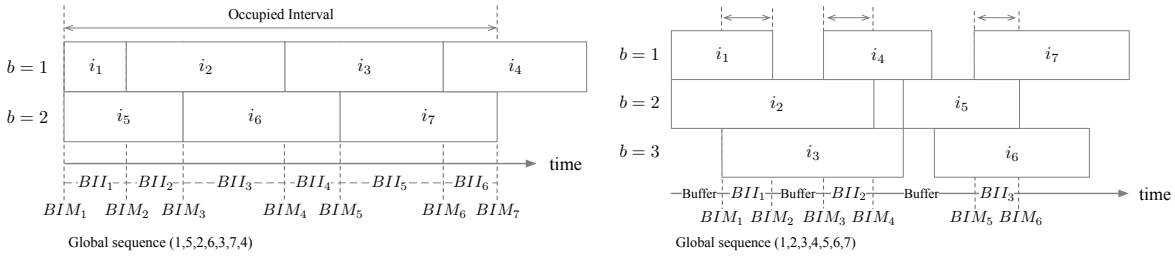
The most important part of the $SSFU_{PS}$ model is the BIMs constraints. As mentioned in Section 1.3.1, researchers have proposed several MIP models for modeling the BIM and buffer mechanisms. In this section, we underline the main differences between our proposed $SSFU_{PS}$ model and those proposed in the literature. Table 4.3 compares the BIM modeling approaches in surgical case scheduling. For each model, we provide the MIP formulation regarding the BIM, achievable strategies, advantages, and disadvantages. For the MIP formulation, we present the decision variables, constraints, and objective function. In the literature, MIP formulations for scheduling problems are often classified based on the choice of the decision variables [130]. After reviewing the literature, we find that three types of decision variables are commonly used, including completion time variables, assignment position variables, and time index variables. The first two types are used to model the continuous duration of surgery, while the last type is used to model the discrete duration of surgery. A detailed explanation of these variables can be found in [130].

To the best of our knowledge, Van Essen et al. [58] was the first to propose a MIP model for BIMs by using the concept of *global sequence*, where surgeries are programmed without dead times in between. Figure 4.4(a) shows an example of 7 surgeries assigned to 2 blocks, with 7 BIMs and 6 BIIs. Apart from the BII_1, all other BIIs can be given by the difference of the completion times of two consecutive surgeries in the *global sequence*, i.e., $BII_2 = f_5 - f_1$, $BII_3 = f_2 - f_5$, $BII_4 = f_6 - f_2$, $BII_5 = f_3 - f_6$, $BII_6 = f_7 - f_3$. Thus, they only minimize the maximum BII among $BII_2, BII_3, \cdots, BII_6$. Later, Vandenberghe et al. [56] also employed the concept of *global sequence* to model the BIMs considering uncertain surgery durations. However, these two studies do not insert buffers. This results in a certain lower limit for BII (> 0), thus leading to a lack of flexibility.

Recently, some researchers have proposed MIP models for BIMs with insertion buffers. However, the calculation of the BII length may be inaccurate in some situations. To illustrate this, we first provide an example of a surgery schedule with buffers in Figure 4.4(b). In this example, 7 surgeries were in 3 blocks, with 6 BIMs, 3 BIIs, and 3 buffers. The global sequence is $1, 2, 3, 4, 5, 6, 7$, and each length of BII can only be calculated by the difference of the completion time and start time of two overlapping surgeries which satisfy the conditions given in Theorem 3, i.e., $BII_1 = f_1 - z_3$, $BII_2 = f_2 - z_4$, $BII_3 = f_5 - z_7$. In the literature, Schulz and Fliedner [59] studied the BIMs mechanism in surgical case

Table 4.3 – Comparison of BIM modeling approaches in surgical case scheduling

Research	MIP formulation for BIM			Advantages	Disadvantages
	Variables	Constraints	Objective		
[58, 56]	Completion time variables	The BII is given by the difference of the completion times of two consecutive surgeries in the *global sequence*.	$\min \max BII$	Scheduling with irregular surgical times, the resources can be utilized efficiently.	Without allowing insertion buffers, this results in a certain lower limit for BII (> 0), thus leading to a lack of flexibility.
[57]	Completion time variables	The BII is given by the difference of the completion time and the start time of two overlapping surgeries. Restrict the BIM to be present within the given time duration.	-	Scheduling with irregular surgical times, the resources can be utilized efficiently.	The overlapping cases are not fully considered when the buffers are inserted. As a result, the model may be inaccurate in some situations.
[59]	Completion time variables	The BII is given by the difference of the completion times of two consecutive surgeries in the *global sequence*.	$\min(\max BII - \min BII)$	Scheduling with irregular surgical times, the resources can be utilized efficiently.	The calculation of BII may be inaccurate when adopting the concept of *global sequence* with allowing insertion buffers.
[61, 60] [71]	Time index variables Assignment position variables	The surgery day is divided into several time periods, and restrict the BIM to exist in the given periods	-	Divide the surgery day into several equally time periods, simplify the modeling and solve the problem easily.	The length of the time period greatly affects the performance of the model. If the time period is too short, it may lead to a large number of variables and constraints.
Ours	Completion time variables	The BII is given by the difference of the completion time and the start time of two overlapping surgeries, which satisfy conditions as provided in Theorem 3.	$\min \max BII$	Scheduling with irregular surgical times, the resources can be utilized efficiently. All overlapping cases are well considered, ensuring the accuracy of the model.	The model is complex and requires a large number of variables and constraints.



(a) Surgery schedule without buffer

(b) Surgery schedule with buffers

Figure 4.4 – Demonstration of surgery schedule.

scheduling, where idle time is allowed between surgeries. They adopted the concept of *global sequence* and calculated the length of BII by the difference of the completion times of two consecutive surgeries in the *global sequence*. However, it is easy to know that the above BIIs, in Figure 4.4(b), cannot be accurately calculated by their method. Moreover, Latorre-Nunez et al. [57] calculated the BII by the difference between the completion time and the start time of two overlapping surgeries. However, when the buffers are inserted, the overlapping cases are not fully accounted for. As a result, the model may be inaccurate in some situations. Appendix B provides a detailed discussion on the calculation of BII for the two models above.

Unlike the above models, Jung et al. [61] and Wang et al. [60] used the time index variable formulation, where the surgery day is discretized into several periods, and the surgery duration is also discretized. Similarly, Freeman et al. [71] used assignment position variables formulation, where the surgery day is discretized into several periods, but the surgery duration is continuous. All three models restricted the BIM from existing in given periods. The above method simplifies the modeling of the BIMs and buffer mechanism, which makes the problem easier to solve. However, the length of the period greatly affects the performance of the model. If the period is too long, the medical resource utilization may be inefficient, and the length of BII will be large. If the time period is too short, it may lead to a large number of variables and constraints, which will increase the complexity of the model.

4.4 Computational experiments

4.4.1 Instances design and experimental protocol

Our computational study is based on anonymized real-world surgery data presented by [131] and [132]. This data set is collected from a hospital in Oslo, Norway, and involves daily surgery data from 2006 to 2008, including 10,390 surgeries. These surgeries belong to six different surgical specialties, namely General Cardiology (CARD), Gastroenterology (GASTRO), Gynecology (GYN), Medicine (MED), Orthopedics (ORTH), and Urology (URO). Durations of these surgeries follow a Lognormal distribution, in which mean and standard deviation are based on the surgery type, as presented in Table 4.4. The percentage of each surgery type is also presented in the Table.

The problem instances consist of 70, 100, 140, and 200 surgeries of different sizes,

Table 4.4 – Distribution of surgery duration (in minutes) for different surgery types.

Surgery type	Mean (minute)	Standard deviation (minute)	Percent (%)
CARD	99	53	14.01
GASTRO	132	76	17.79
GYN	78	52	27.81
MED	75	72	4.41
ORTH	142	58	17.81
URO	72	38	17.98

16, 25, and 32 surgery blocks with 3 different block layouts, resulting in 36 instances. In detail, the number of ORs of these 3 surgery blocks is 5, 8, and 10, respectively. The block layouts are generated according to three policies: (1) Random allocation policy (RAP), assigning the surgery block to the planning horizon randomly; (2) Uniform allocation policy (UAP), assigning each type of surgery block to different days as uniformly as possible; (3) Concentrated allocation policy (CAP), assigning the same type of surgery block to the same day as much as possible. Figure 4.5 shows the block layout of these three policies in the amount of 32 surgery blocks. In addition, for emergency surgeries, we assume that the arrival rate follows a Poisson distribution, and the duration follows a uniform distribution according to [133, 61]. Specifically, the surgery arrival rate $\lambda = |\mathcal{E}_d|/L$ is related to the number of emergency surgeries $|\mathcal{E}_d|$, and the range of the duration of emergency surgeries is in $[60, 180]$.

OR	Mon	Tue	Wed	Thu	Fri
OR1	URO	CARD	URO	URO	ORTH
OR2	ORTH	GYN	GYN		GASTRO
OR3		CARD		GYN	
OR4	GYN	CARD		GYN	
OR5	GYN	ORTH			GASTRO
OR6				GASTRO	GASTRO
OR7	GYN	GYN		GASTRO	
OR8	CARD	GASTRO		ORTH	URO
OR9			CARD	URO	URO
OR10	ORTH			MED	ORTH

(a) RAP

OR	Mon	Tue	Wed	Thu	Fri
OR1	GYN	CARD	URO	ORTH	GASTRO
OR2	CARD		GYN		URO
OR3		URO		GASTRO	
OR4	GASTRO	GYN	CARD	URO	
OR5	URO	ORTH	URO		ORTH
OR6		GASTRO		MED	CARD
OR7	GYN			ORTH	GYN
OR8		GYN	GASTRO	CARD	
OR9	ORTH		ORTH		
OR10	GASTRO		GYN	GYN	

(b) UAP

OR	Mon	Tue	Wed	Thu	Fri
OR1	GYN				MED
OR2	GYN	GASTRO	ORTH	URO	CARD
OR3	GYN	GASTRO	ORTH	URO	CARD
OR4		GASTRO	ORTH	URO	CARD
OR5		GASTRO	ORTH		
OR6	GYN				CARD
OR7	GYN	GASTRO	ORTH	URO	CARD
OR8	GYN		ORTH	URO	
OR9	GYN	GASTRO			
OR10	GYN			URO	

(c) CAP

Figure 4.5 – Three block layout policies for 32 surgery blocks.

We consider the following cost structure for the objective function. First, the cost of overtime is set to $C^O = \$26/\text{min}$, and the cost of idle time is $C^I = 26/1.5 \approx \$17/\text{min}$, which same as the previous studies [134, 45]. Second, the cost of performing surgery is set to $C_{ib} = 0.75 * C^O * \bar{D}_i * \alpha \approx \$20/\text{min} * \bar{D}_i * \alpha$. Where α is a coefficient set up to differentiate individual surgeries, randomly selected within the range [0.8,1.2]. Third, the

cost of postponing surgery and canceling surgery is set to be the same, $C_i^P = C_i^Q = 1.5 * C_{ib}$. Finally, the cost of tardiness of elective surgery is set to $C^T = C^O/2 = \$13/\text{min}$, and the cost of waiting time of emergency surgery is set to $C^W = 2 * C^T = \$26/\text{min}$.

Our approach was coded in Python and the Gurobi Optimizer 9.0.3 was called to solve the model. Experiments are run on a 6 nodes computing cluster where each node is equipped with double Inter(R) Xeon(R) Gold 6226R (16 cores) 2.90GHz CPUs and 256Gb RAM. The time limits are set to 1h, 5h, 10h, and 10h for the instances with 70, 100, 140, and 200 surgeries, respectively. The performance of the solution approach is dependent on the number of samples available. More samples lead to a better approximation of the objective function, but also increase the computational efforts. We are inspired by the approach of [71] to use a relatively small number of scenarios to provide information regarding the surgery duration and the arrival of emergency surgeries. In detail, we generate a certain number of scenarios $|\mathcal{S}|$, where the first scenario is constructed using the expected values of all random parameters to represent the central tendencies of the associated distributions, capturing the overall average behavior. Subsequently, the remaining scenarios are generated by randomly sampling from the specified distributions of the random parameters, allowing us to capture the variations and dispersion of the random parameters. There is a trade-off between computational time and solution quality when choosing the number of scenarios in our solution approach. To achieve the best performance, we consider a set of scenarios $|\mathcal{S}| = \{5, 10, 15, 20, 30, 40, 50\}$. In preliminary experiments, we found that the number of scenarios is related to the surgical demand and the supply of surgery blocks, which can be represented by the supply-demand ratio $SDR = ((O+L) \times |\mathcal{B}|)/\sum_{i \in \mathcal{I}} \bar{D}_i$. In detail, the SDR values of each instance are presented in Table 4.5, where the instances are divided into three groups: low SDR, medium SDR, and high SDR. High SDR values suggest that the availability of surgery blocks exceeds the demand for surgical procedures, whereas low SDR values indicate that the demand for surgical procedures surpasses the availability of surgery blocks. In our preliminary experiments, we found that the best number of scenarios decreases from 50 to 5 as the SDR increases from 0.48 to 1.37, while the best number of scenarios increases from 5 to 50 as the SDR increases from 1.37 to 2.70. For different SDR values, the optimal number of scenarios is as follows: 50 for SDR = 0.19 and 0.32, 40 for SDR = 0.75 and 0.96, 15 for SDR = 1.06, 5 for SDR = 1.34, 10 for SDR = 1.49, 40 for SDR = 1.92, 50 for SDR = 2.13 and 2.70.

Table 4.5 – Supply-demand ratio (SDR) of all instances

Group	Low SDR					Medium SDR				High SDR		
Instance	100-16	140-16	200-16	200-25	200-32	70-16	100-25	140-25	140-32	70-25	70-32	100-32
SDR value	0.96	0.68	0.48	0.75	0.96	1.37	1.49	1.06	1.37	2.13	2.70	1.92

4.4.2 Experimental results

To assess the performance of our proposed models, we use a discrete-event simulation (DES) approach to evaluate the surgery plan generated by solving the $SSFU_{PS}$ model. To quickly reschedule unstarted surgeries, we adopt simple rescheduling rules rather than solving the $SSFU_{RS}$ model. Thus, we refer to this approach as the sample average approximation (SAA) approach. We use the value of the stochastic solution (VSS) to measure the quality gained by considering the uncertain information when solving the SSFU problem with known distributions of random parameters. To compute the VSS, we first define the expected value problem (EVP). Specifically, we replace the random parameters of the $SSFU_{PS}$ model with the expected values to create the scenario for the EVP. Table 4.6 shows the VSS values of the SAA and EVP approaches for the 36 instances. The VSS value is computed as $VSS(\%) = (Obj_{EVP} - Obj_{SAA})/Obj_{EVP} \times 100$, where Obj_{EVP} and Obj_{SAA} are the objective values of the EVP and SAA approaches evaluated by the DES under the same 500 scenarios, respectively.

Table 4.6 – Computational results of EVP and SAA approaches.

Elective surgeries	Blocks	CAP			RAP			UAP		
		Obj_{SAA}	Obj_{EVP}	VSS (%)	Obj_{SAA}	Obj_{EVP}	VSS (%)	Obj_{SAA}	Obj_{EVP}	VSS (%)
70	16	215,751.81	215,195.67	-0.26	213,912.03	212,923.74	-0.46	211,463.54	210,811.49	-0.31
100	16	303,227.58	300,930.64	-0.76	294,692.89	297,763.74	1.03	298,443.49	299,885.05	0.48
140	16	418,047.32	423,368.08	1.26	411,027.26	419,996.71	2.14	421,940.52	423,481.13	0.36
200	16	594,298.57	597,424.88	0.52	596,761.63	596,947.85	0.03	601,744.70	601,764.49	0.00
70	25	196,101.30	196,184.00	0.04	196,149.66	199,560.53	1.71	195,406.24	204,291.12	4.35
100	25	287,608.20	296,109.06	2.87	266,267.04	292,673.96	9.02	281,549.32	300,679.83	6.36
140	25	410,631.40	408,653.07	-0.48	397,949.04	409,181.15	2.75	408,809.67	429,861.92	4.90
200	25	596,961.09	595,002.35	-0.33	587,478.59	609,583.10	3.63	591,621.29	615,773.72	3.92
70	32	197,235.03	196,727.59	-0.26	194,564.66	199,096.62	2.28	194,541.47	200,216.32	2.83
100	32	281,060.78	289,974.14	3.07	255,222.25	281,610.18	9.37	263,153.17	286,097.79	8.02
140	32	398,143.34	401,993.95	0.96	384,627.21	415,723.10	7.48	372,220.76	398,703.81	6.64
200	32	588,268.34	589,237.53	0.16	579,675.90	610,642.77	5.07	575,083.94	603,768.52	4.75

Table 4.6 shows that the majority of VSS values exhibit an upward trend as the number of ORs increases for a specific block layout and a given number of elective surgeries.

For instance, the VSS values increase from 2.14% to 7.48% as the number of ORs rises from 16 to 32 for the RAP block layout with 140 elective surgeries. It can be observed that the majority of the VSS values exhibit an initial increase as the number of elective surgeries increases from 70 to 100, followed by a subsequent decline as the number of elective surgeries increases from 100 to 140 for a specific block layout and a given number of operating rooms (ORs). In particular, the VSS values increased from 2.28% to 9.37% as the number of elective surgeries increased from 70 to 100, and then decreased from 9.37% to 5.07% as the number of elective surgeries increased from 100 to 200 for the RAP block layout with 32 ORs. The reason for the former is that as the value of SDR decreases, the uncertainty of the SSFU problem increases, and the SAA outperforms the EVP, resulting in a higher VSS value. The reason for the latter is that as the number of elective surgeries continues to increase, the complexity of the problem increases significantly, and the performance of the SAA approach deteriorates, which leads to a lower VSS value. Furthermore, it can be observed that the VSS values are significantly influenced by the block layout. In particular, the VSS values are less than 1% for nine instances with CAP block layout, three instances are between 1% and 3.5%. This is because the same type of surgery block is assigned on the same day under the CAP, which allows elective surgeries to be rescheduled to other blocks rather than being canceled when uncertain events occur. In conclusion, the results demonstrate that our proposed method can effectively address the uncertainty in the SSFU problem.

4.4.3 Impact of the buffer mechanism

To reveal the impact of the buffer mechanism on the surgical scheduling plan, we perform experiments with different values of C^W, which significantly affect the size and number of buffers. We set C^W as {0,26,50,100,200,400} and perform experiments on the above 36 instances. Figure 4.6 shows the daily average idle time (minute) for all ORs as the value of C^W increases. There is a clear trend that larger values of C^W result in longer idle times in the ORs to accommodate the buffers for emergency surgeries.

From Figure 4.6, we can observe that with a fixed C^W, as the SDR increases, the idle time in the operating room decreases. The scarcer the surgical resources, the shorter the idle time in the operating room. Specifically, the idle times are 484.1 minutes for high SDR, 367.2 minutes for medium SDR, and 303.7 minutes for low SDR when C^W=0. With the increase of the C^W, the average daily idle time in the operating room increases. Especially, the idle time reaches the longest opening time of a whole operating room, 600

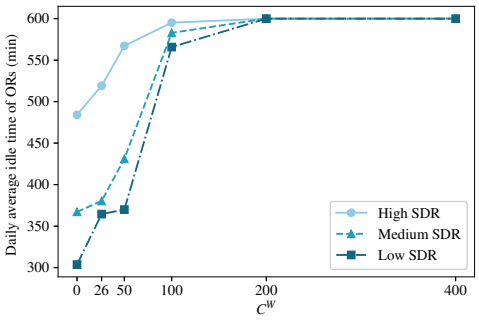
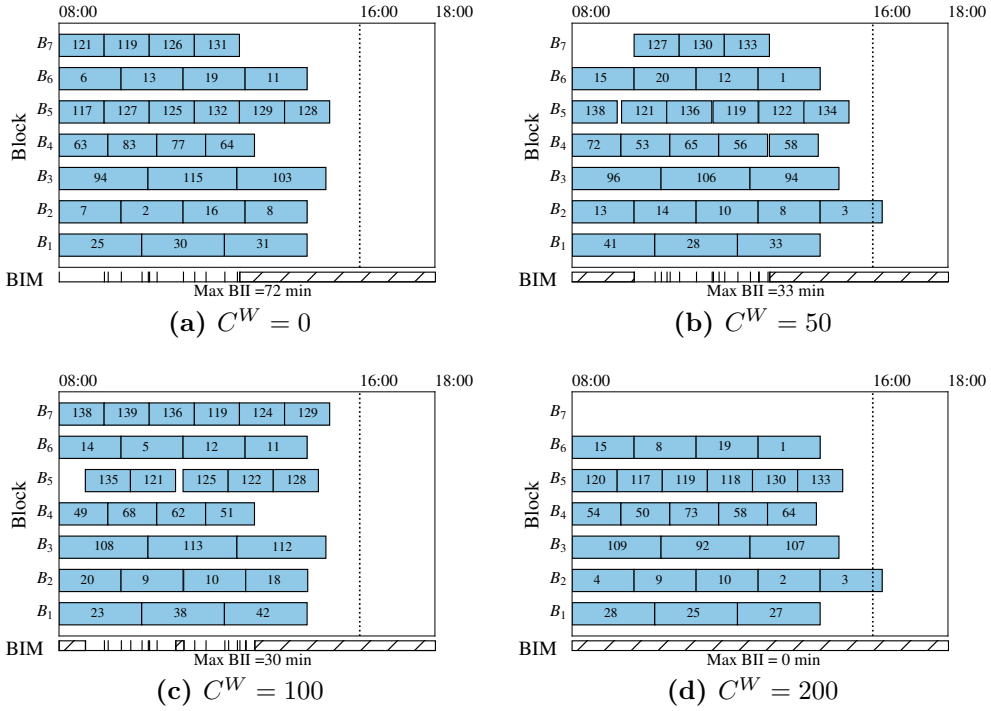


Figure 4.6 – The daily average idle time (minute) in different values of C^W

minutes, when $C^W \geq 200$. This indicates that the size of the buffer can be controlled by adjusting the value of C^W, but the effective value of C^W is limited to 200.

To gain a more intuitive understanding of the surgical plan obtained by our solution approach under different C^W settings, we display initial surgical plans for instance 140-32-UAP for Monday under C^W=0, 50, 100, 200 as shown in Figure 4.7. In each subfigure, the x-axis represents the time horizon, and the y-axis represents the surgery block number. In addition, the bottom of each subfigure shows the BIM and BII, where the former is represented by vertical lines and the latter is represented by the shaded boxes. From this figure, we can observe that when C^W=0, the maximal BII is 72 minutes, with virtually no buffer between surgeries in the resulting schedule, which can be seen as the "BIMs only" strategy. As the C^W increases from 0 to 100, there exist buffers in block 5 and block 7, and the maximal BII is decreased to 33 minutes. When C^W=100, there are also two buffers in block 5, which are located before surgery 135 and between surgery 121 and 125, respectively, and the maximal BII is decreased to 30 minutes. When C^W=200, OR is entirely freed up, and no buffer is added for other blocks. In addition, the maximal BII is decreased to 0 minutes, which can be seen as the "BIMs + extra OR" strategy. Therefore, this experiment demonstrates how the maximal BII is reduced by adding buffers controlled by adjusting the value of C^W. Moreover, the results also show that our proposed modeling approach for the BII bridges the "BIMs only" strategy, the "BIMs + extra OR" strategy, and the "BIMs + buffers" strategy. Our approach allows for balancing the trade-off between these three strategies within the same mathematical programming model.

As the size of the buffer directly impacts the waiting time for elective and emergency surgeries, we have collected the average waiting times for elective and emergency surgeries across 10 groups of instances with different supply-demand ratios under varying C^W settings, as displayed in Figure 4.8. From this figure, we can see that as the SDR increases,



(a) $C^W = 0$ **(b)** $C^W = 50$

(c) $C^W = 100$ **(d)** $C^W = 200$

Figure 4.7 – Surgery plans (Monday) for instance 140-32-UAP under different C^W

the waiting times for both elective and emergency surgeries decrease under all three C^W settings, indicating that the scarcer the surgical resources, the longer the patient waiting times. Moreover, we can see that as C^W increases, the average waiting times for both elective and emergency surgeries decrease, especially for emergency surgeries, whose waiting times decrease significantly. The reason for reducing the delay time of elective surgeries is that buffers can be inserted a few minutes before the surgery to ensure that the surgery can be performed on time. Especially when C^W=200, the waiting time for emergency surgeries can be decreased to one minute. These results demonstrate that the strategy of setting buffers considering BIM can reduce patient waiting times, especially effectively reducing the waiting time for emergency surgeries.

4.5 Chapter conclusion

In this chapter, we studied a surgical case scheduling problem in flexible ORs under uncertainty (SSFU), which consists of operational decisions of assignment (assign patients to OR blocks in a given time horizon) and sequencing (determine the start time of the

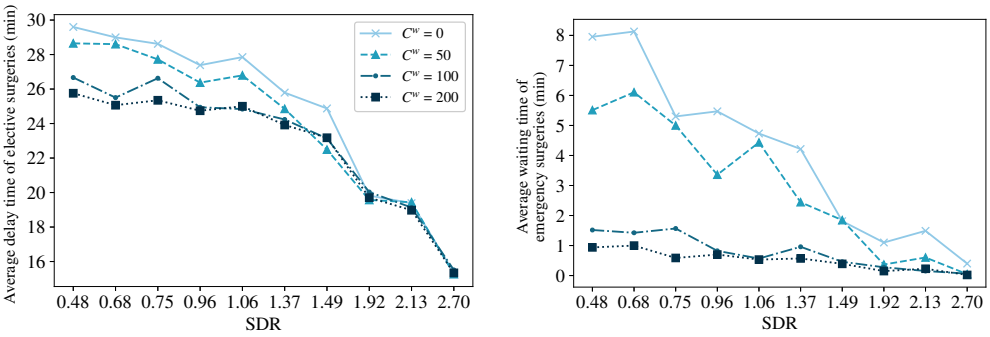


Figure 4.8 – The tardiness of elective surgery and waiting time of emergency surgery with different values of C^W

assigned surgeries in each OR block) while taking into account the uncertainty of elective surgery duration, emergency surgeries arrival, and their duration. We adopt the integrated proactive and reactive strategies, where a proactive SSFU problem is first solved to generate an initial elective surgery plan, and then a reactive SSFU problem is solved to dynamically adjust the plan based on actual surgery durations and emergency arrivals. To deal with the uncertainty of surgery durations and emergency arrivals in the proactive SSFU problem, we implemented three mechanisms — reserving capacity, BIM, and buffer — to improve the robustness of the plan.

We formulated the proactive SSFU problem as a two-stage stochastic programming problem, where the first stage assigns elective surgeries to OR blocks and determines the start time of each surgery, and the second stage evaluates the quality of the obtained plan under different durations of elective/emergency surgeries. We proposed a scenario-based model $SSFU_{PS}$, where minimizing the longest possible waiting time of emergency surgeries by inserting buffers is different from the traditional surgical case scheduling problem. We also proposed a MIP model $SSFU_{RS}$ to formulate the proactive SSFU problem.

We conducted extensive computational experiments to evaluate the performance of the proposed models. Extensive computational experiments demonstrated that our proposed models significantly outperform the traditional deterministic approach under different block layout strategies. Moreover, we discussed the impact of the buffer mechanism on the performance of the surgery plan, and analyzed the parameter sensitivity of the buffer mechanism.

Chapter 5

A THREE-PHASE SIMULATION-OPTIMIZATION APPROACH FOR SURGICAL CASE SCHEDULING IN FLEXIBLE OPERATING ROOMS UNDER UNCERTAINTY

In this chapter, we present a hybrid simulation and optimization approach for the surgical case scheduling problem in flexible operating rooms (ORs) under uncertainty (SSFU). An initial surgery plan obtained by solving the $SSFU_{PS}$ model may not be good enough since it does not consider the possible impact of dynamic rescheduling. To obtain a high-quality initial elective surgery plan, we propose an innovative three-phase simulation-optimization (TPSO) approach where the result of dynamic rescheduling in the planning horizon is considered. To decrease the complexity of the proactive SSFU problem, we decompose it into an elective surgery assignment subproblem considering emergency demand ($ESAP_{REC}$), and multiple elective surgery sequencing subproblems with BIMs & buffers ($ESSP_{B\&B}$). We apply a discrete-event simulation algorithm to evaluate the plan quality. To effectively feedback the evaluation results to $ESAP_{REC}$ and $ESSP_{B\&B}$ models, we develop a set of novel constraints, including sequence feedback and assignment feedback. Especially, for the assignment feedback, we propose local best assignment (LBA) feedback constraints to reduce the search space. We conducted extensive computational experiments to evaluate the performance of the proposed TPSO approach. The content of this chapter is based on an article submitted to *Production and Operations Management* and *Journal of Systems Engineering*.

5.1 Introduction

As demonstrated in Section 4.2, the elective surgery plan is initially generated by solving the proactive SSFU model, which is dynamically modified when unexpected events occur during the surgery day. Thus, a high-quality initial elective surgery plan is essential to improve the efficiency of the surgical case scheduling process. However, after multiple dynamic rescheduling for a given initial plan, the actual OR overtime, idle time, waiting time for emergency surgeries and cancellation of scheduled surgeries may be higher or lower than expected, which may lead to a suboptimal plan. In this chapter, we present a three-phase simulation-optimization (TPSO) approach, which further considers the result of dynamic rescheduling for the given plan. The distinctive features of our proposed TPSO approach are summarized as follows:

(1) To decrease the complexity of the proactive SSFU problem, we decompose it into an elective surgery assignment subproblem considering emergency demand ($ESAP_{REC}$), and multiple elective surgery sequencing subproblems with BIMs & buffers ($ESSP_{B\&B}$). By sequentially solving these two subproblems, an initial surgery plan can be obtained.

(2) We apply a discrete-event simulation algorithm to evaluate the plan quality. To accelerate the simulation process, we propose simple rescheduling rules to quickly reschedule the unstarted surgeries rather than solving the $SSFU_{RS}$ model.

(3) To effectively feedback the evaluation results to $ESAP_{REC}$ and $ESSP_{B\&B}$ models, we develop a set of novel constraints. One of them is the local best feedback constraints, which only retain one solution from the given assignment and remove all other solutions to reduce the search space.

Moreover, we carry out extensive experimental analysis using challenging test instances with different characteristics. Extensive computational experiments demonstrate that our proposed TPSO approach significantly outperforms the sample average approximation approach, and the typical deterministic approach. In addition, experimental analysis is carried out to identify the impact of feedback mechanisms of the proposed TPSO approach, and the effect of the buffer mechanism.

Section 5.2 describes the proposed TPSO approach in detail. Section 5.3 presents the computational results and analyzes the critical components of the TPSO approach. Section 5.4 draws the conclusions.

5.2 Three-phase simulation-optimization approach

5.2.1 General scheme

Given the complexity of the proactive SSFU problem, our proposed TPSO approach decomposes it into two subproblems: the elective surgery assignment subproblem with reserving emergency capacity, denoted as $ESAP_{REC}$, and the elective surgery sequencing subproblem with BIMs and buffers, denoted as $ESSP_{B\&B}$. The $ESAP_{REC}$ generates a new elective surgery assignment by selecting surgeries from the waiting list, allocating them to the surgery blocks, and reserving capacity for emergencies. Since the surgery assignment decision is made without considering the sequence of surgeries, $ESSP_{B\&B}$ re-optimizes the given day's surgery assignment to obtain an improved assignment and determines the start times of surgeries to generate an updated plan. Thus, the main idea of the TPSO approach is to iteratively optimize these two subproblems and evaluate the obtained elective surgery plan in each iteration. In addition, the evaluation result is also used to guide the search for an improved plan.

Algorithm 2 presents the general scheme of our proposed TPSO approach. It starts with a scenario generation procedure to generate the scenarios S applied in both optimization and simulation procedures. Note that the scenario generation method is the same as the one used in Section 4.4. Then, it creates the $ESAP_{REC}$ model by using the scenarios S as the input data, and initializes the necessary parameters. In the following, it iteratively solves $ESAP_{REC}$ and $ESSP_{B\&B}$ models to generate an elective surgery plan, and evaluates the obtained plan to generate feedback. The $ESAP_{REC}^{(k)}$ model is solved within a time limit TL_1 (see Section 5.2.2). According to the new obtained surgery assignment $\mathcal{I}_b^{(k)}$, a total of $|\mathcal{D}|$ $ESSP_{B\&B}^{(k,j)}$s are generated (see Section 5.2.3). After solving these models, the obtained plan is subsequently evaluated by simulation, employing the discrete-event simulation (DES) method to simulate the rescheduling process under scenarios S. Consequently, record relevant measures, such as OR overtime and idle time, emergency surgery waiting time, tardiness, and cancellation of elective surgeries (see Section 5.2.4). The sequencing feedback is generated based on the evaluated values of the measures, and then feedback to $ESSP_{B\&B}^{(k,j+1)}$ model to assist in finding a higher quality plan (see Section 5.2.5). Optimizing $ESSP_{B\&B}^{(k,j)}$ model, DES simulation, and result feedback to $ESSP_{B\&B}^{(k,j+1)}$ are repeated for a given number of iterations $MaxIter$. The assignment feedback is generated and passed back to $ESAP_{REC}^{(k+1)}$ to find a higher quality assignment (see Section 5.2.5). The TPSO algorithm terminates when the elapsed time reaches the given time limit. The above

Algorithm 2: Outline of the TPSO algorithm

Input : A given problem instance NP, the time limit TL_1 and TTL, the maximum number of iterations $MaxIter$, the number of scenarios $|\mathcal{S}|$.
Output : The best elective surgery plan $(\mathcal{I}_b^*, \boldsymbol{f}^*)$ with the corresponding objective value Obj^* found during the optimization process.

$\mathcal{S} \leftarrow$ ScenarioGeneration(NP, $|\mathcal{S}|$);
Create $ESAP_{REC}^{(0)}$ model according to $\mathcal{S}$, the best objective value $Obj^* \leftarrow +\infty$, the iteration count $k \leftarrow 0$;
while *the elapsed time does not reach* TTL **do**
  $\mathcal{I}_b^{(k)} \leftarrow$ SolveModel($ESAP_{REC}^{(k)}$, TL_1);
  $ESSP_{B\&B}^{(k,0)} = \{ESSP_{B\&B,d}^{(k,0)} | d \in \mathcal{D}\} \leftarrow$ Create $ESSP_{B\&B}^{(k,0)}$ models according to $\mathcal{I}_b^{(k)}$;
  Set the iteration count $j \leftarrow 0$, the local best solution $Obj^{(k)} \leftarrow +\infty$;
  while $j < MaxIter$ **do**
    $(\bar{\mathcal{I}}_b^{(k,j)}, \boldsymbol{f}) \leftarrow$ SolveModel($ESSP_{B\&B}^{(k,j)}$);
    $(Obj, \eta) \leftarrow$ DESEvaluatePlan($\bar{\mathcal{I}}_b^{(k,j)}$, $\boldsymbol{f}$, $\mathcal{S}$);
    if $Obj < Obj^{(k)}$ **then**
      $\dot{\mathcal{I}}_b^{(k)} \leftarrow \bar{\mathcal{I}}_b^{(k,j)}$, $\boldsymbol{f}^{(k)} \leftarrow \boldsymbol{f}$, $Obj^{(k)} \leftarrow Obj$, $\eta^{(k)} \leftarrow \eta$;
    if $Obj < Obj^*$ **then**
      $\mathcal{I}_b^* \leftarrow \bar{\mathcal{I}}_b^{(k,j)}$, $\boldsymbol{f}^* \leftarrow \boldsymbol{f}$, $Obj^* \leftarrow Obj$;
    $ESSP_{B\&B}^{(k,j+1)} \leftarrow$ AddSequenceFeedback($ESSP_{B\&B}^{(k,j)}$, $\bar{\mathcal{I}}_b^{(k,j)}$, $\boldsymbol{f}$, η);
    $j \leftarrow j + 1$;
  $ESAP_{REC}^{(k+1)} \leftarrow$ AddAssignmentFeedback($ESAP_{REC}^{(k)}$, $\eta^{(k)}$, $\mathcal{I}_b^{(k)''}$, $\mathcal{I}_b^{(k)}$);
  $k \leftarrow k + 1$;
return $(\mathcal{I}_b^*, \boldsymbol{f}^*, Obj^*)$;

mentioned procedure is illustrated in Figure 5.1. In the following sections, we present the main components of our TPSO approach.

5.2.2 $ESAP^{REC}$ model

The $ESAP_{REC}$ is used to generate an elective surgery assignment and reserve capacity for emergencies. The $ESAP_{REC}$ model can be derived by extracting the assignment decisions and constraints from the first-stage $SSFU_{PS}$ model. In addition, we introduce a new auxiliary variable η_d^A to represent the expected total cost of the given assignment, including the costs of idle time, overtime, tardiness, cancellation of elective surgeries, and

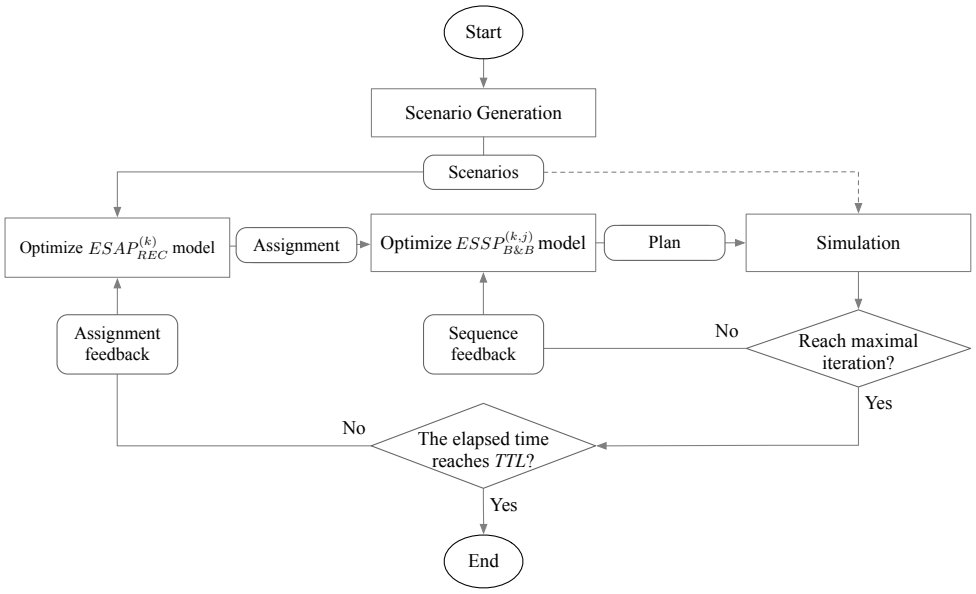


Figure 5.1 – Schematic of the TPSO approach.

waiting times of emergency surgery. Thus, the formulation of $ESAP_{REC}$ in iteration k is as follows:

$$\boldsymbol{ESAP^{(k)}_{REC}}: \quad \min \sum_{b\in\mathcal{B}} \sum_{i\in\mathcal{I}_b} C_{ib}x_{ib} + \sum_{i\in\mathcal{I}} C^P_i \left(1 - \sum_{b\in\mathcal{B}|i\in\mathcal{I}_b} x_{ib}\right) + \sum_{d\in\mathcal{D}} \eta^A_d \tag{5.1}$$

s.t. Constraints (4.2), (4.21), (4.31) – (4.36)

$$\eta^A_d \geq \sum_{s\in\mathcal{S}} P(s) \left(\sum_{i\in\mathcal{I}} \sum_{b\in\mathcal{B}_d|i\in\mathcal{I}_b} C^Q_i u_{ibs} + \sum_{b\in\mathcal{B}_d} C^O o^I_{bs}\right) \quad d\in\mathcal{D} \tag{5.2}$$

$$\eta^A_d \geq 0 \quad d\in\mathcal{D} \tag{5.3}$$

The objective function (5.1) minimizes the cost of performing and postponing surgeries, and the estimation of costs related to idle time, overtime, cancellation of elective surgeries, and emergency surgery waiting times. Constraint (5.2) estimates the expected total cost of canceling elective surgeries and overtime. Although constraint (5.2) just provides an estimation of the cost of canceling surgeries and overtime, it is beneficial to the optimization process by providing a guideline for the assignment decisions. Moreover, the expected total cost of the assignment is evaluated by the latter simulation procedure in the third phase, and will be feedback to the $ESAP_{REC}$ model. Constraint (5.3) defines

the domain of the variables η_d^A.

The $ESAP_{REC}^{(k)}$ solution is denoted as $\boldsymbol{x}^{(k)}$. The sets of surgeries that $ESAP_{REC}^{(k)}$ assigns to surgery block b and day d are denoted as $\mathcal{I}_b^{(k)}$ and $\mathcal{I}_d^{(k)}$, respectively. Two stopping criteria are used for the $ESAP_{REC}^{(k)}$: (1) a specific optimality gap and (2) maximum time per iteration, which is 90% of the total remaining CPU time. These two above criteria are similar to [135].

5.2.3 $ESSP^{B\&B}$ model

The $ESSP_{B\&B}$ model is created based on the $ESAP_{REC}$ model outputs to optimize daily surgical case scheduling while minimizing the waiting time of emergency surgeries. The scheduling is bound by the surgery assignment of $ESAP_{REC}$, which was made without consideration of sequencing decisions. Note that allowing the modification of the previous assignments in scheduling may lead to better solutions, as demonstrated in [136, 135, 137]. The trade-off here is between the quality of the solution, which improves with more reassignments, and the complexity of the problem, which also increases correspondingly. The main features of the $ESSP_{B\&B}$ model include employing the BIMs and buffers mechanisms, whose performance highly depends on the reserving capacity of surgery assignment. To leverage the performance of the BIMs and buffers without introducing excessive reassignment variables, the $ESSP_{B\&B}$ model only allows postponing some assigned surgeries to the next planning horizon, thus enhancing the reserving capacity.

Elective surgeries are assigned to surgery blocks with the same surgical specialty as the elective surgery itself, and these surgeries have the same distribution of surgery duration, which is the same as [131, 134]. This indicates that the only difference between surgeries within the same block is the cost of performing and postponing. Therefore, there is a symmetry issue in $ESSP_{B\&B}$, which is assigning multiple surgeries to a single surgery block can lead to numerous alternative optimal solutions generated by simply renumbering the surgeries. To break this symmetry and accelerate the solving process, we can determine the sequence for each surgery block before optimizing according to a deterministic sequence rule SR: *the sequence for each surgery block can be obtained by sorting the surgeries in descending order based on the postponing cost* C_i^P. *If the postponing costs of two surgeries are equal, the one with the lower performing cost* C_{ib} *is operated first. Moreover, in any surgery block, surgery with a lower ranking will be postponed first if necessary.*

The idea of symmetry-breaking in sequence rule SR is inspired by [138]. The optimal sequence of any surgery block can be obtained according to the sequence rule SR (see the

proof A.4).

Theorem 4 *The optimal sequence for each surgery block can be obtained according to the sequence rule SR.*

Given the surgery assignment of day d, the sequence of surgeries in each surgery block can be determined according to Theorem 1. Specifically, we use (n,b) to represent the n-th surgery in block b. Given k-th assignment, the notation of $ESSP_{B\&B}^{(k,j)}$ model in iteration j is given in Table 5.1, which is formulated as follows:

Table 5.1 – Notation used for the $ESSP_{B\&B}$ model.

Symbol	Description
Sets	
$\mathcal{N}_b^{(k)}$	Set of patients assigned to block b in iteration k $(n = 1, ..., \lvert\mathcal{N}_b^{(k)}\rvert)$
Parameters	
C_{nb}	The cost of performing the n-th surgery assigned to block b
C_{nb}^P	The cost of postponing the n-th surgery assigned to block b
Variables	
x_{nb}	1 if the n-th surgery in block b is operated, 0 otherwise
z_{nb}	Start time of the n-th surgery in block b
f_{nb}	Finish time of the n-th surgery in block b
$w_{nbn'b'}$	1 if the finish time of the n'-th surgery in block b' is the closest one and strictly later than the start time of the n-th surgery in block b, 0 otherwise
l	Continuous variable for idle time
$g_{nbn'b'}$	1 if the start time of the n-th surgery in block b is no less than the start time of the n'-th surgery in block b', 0 otherwise
$\phi_{nbn'b'}$	1 if the finish time of the n-th surgery in block b is strictly greater than the start time of the n'-th surgery in block b', 0 otherwise
q^{MAX}	The maximum length of all BIIs
η	Estimation of costs related to idle time, overtime, tardiness and cancellation of elective surgeries, and emergency surgery waiting times

$$\boldsymbol{ESSP}_{\boldsymbol{B\&B}}^{(\boldsymbol{k},\boldsymbol{j})}: \quad \min \sum_{b\in\mathcal{B}_d}\sum_{n\in\mathcal{N}_b^{(k)}} C_{nb}x_{nb} + \sum_{b\in\mathcal{B}_d}\sum_{n\in\mathcal{N}_b^{(k)}} C_{nb}^P(1-x_{nb}) + \eta \tag{5.4}$$

$$\text{s.t.} \qquad \eta \geq \sum_{b\in\mathcal{B}_d} C^O o_b + C^I l + C^W q^{MAX} \tag{5.5}$$

$$x_{nb} \geq x_{n+1,b} \quad \forall b \in \mathcal{B}_d; n \in \mathcal{N}_b^{(k)} \setminus \{|\mathcal{N}_b^{(k)}| - 1\} \tag{5.6}$$

$$f_{1,b} \geq \bar{D}_b x_{1,b} \quad \forall b \in \mathcal{B}_d \tag{5.7}$$

$$f_{n+1,b} \geq f_{nb} + \bar{D}_b - (O+L)(2 - x_{n+1,b} - x_{nb}) \quad \forall b \in \mathcal{B}_d; n \in \mathcal{N}_b^{(k)} \setminus \{|\mathcal{N}_b^{(k)}| - 1\} \tag{5.8}$$

$$f_{nb} \leq (O+L)x_{nb} \quad \forall b \in \mathcal{B}_d; n \in \mathcal{N}_b^{(k)} \tag{5.9}$$

$$z_{nb} = f_{nb} - \bar{D}_b x_{nb} \quad \forall b \in \mathcal{B}_d; n \in \mathcal{N}_b^{(k)} \tag{5.10}$$

$$v_b \geq f_{nb} \quad \forall b \in \mathcal{B}_d; n \in \mathcal{N}_b^{(k)} \tag{5.11}$$

$$l \geq \sum_{b \in \mathcal{B}_d} (v_b - \sum_{n \in \mathcal{N}_b^{(k)}} \bar{D}_b x_{nb}) \tag{5.12}$$

$$o_b \geq v_b - L \quad \forall b \in \mathcal{B}_d \tag{5.13}$$

$$z_{nb} - z_{n'b'} \leq M(2 + g_{nbn'b'} - x_{nb} - x_{n'b'}) \quad \forall b, b' \in \mathcal{B}_d | b \neq b'; n \in \mathcal{N}_b^{(k)}; n' \in \mathcal{N}_{b'}^{(k)} \tag{5.14}$$

$$z_{n'b'} - z_{nb} - \epsilon \leq M(3 - g_{nbn'b'} - x_{nb} - x_{n'b'}) \forall b, b' \in \mathcal{B}_d | b \neq b'; n \in \mathcal{N}_b^{(k)}; n' \in \mathcal{N}_{b'}^{(k)} \tag{5.15}$$

$$f_{n'b'} - z_{nb} \leq M(2 + \phi_{nbn'b'} - x_{nb} - x_{n'b'}) \quad \forall b, b' \in \mathcal{B}_d | b \neq b'; n \in \mathcal{N}_b^{(k)}; n' \in \mathcal{N}_{b'}^{(k)} \tag{5.16}$$

$$z_{nb} - f_{n'b'} + \epsilon \leq M(3 - \phi_{nbn'b'} - x_{nb} - x_{n'b'}) \forall b, b' \in \mathcal{B}_d | b \neq b'; n \in \mathcal{N}_b^{(k)}; n' \in \mathcal{N}_{b'}^{(k)} \tag{5.17}$$

$$2g_{nbn'b'} \leq x_{nb} + x_{n'b'} \quad \forall b, b' \in \mathcal{B}_d | b \neq b'; n \in \mathcal{N}_b^{(k)}; n' \in \mathcal{N}_{b'}^{(k)} \tag{5.18}$$

$$2\phi_{nbn'b'} \leq x_{nb} + x_{n'b'} \quad \forall b, b' \in \mathcal{B}_d | b \neq b'; n \in \mathcal{N}_b^{(k)}; n' \in \mathcal{N}_{b'}^{(k)} \tag{5.19}$$

$$\sum_{b' \in \mathcal{B}_d} \sum_{n' \in \mathcal{N}_{b'}^{(k)}} w_{nbn'b'} \leq x_{nb} \quad \forall b \in \mathcal{B}_d; n \in \mathcal{N}_b^{(k)} \tag{5.20}$$

$$\sum_{b' \in \mathcal{B}_d} \sum_{n' \in \mathcal{N}_{b'}^{(k)}} w_{nbn'b'} \geq \sum_{b' \in \mathcal{B}_d | b \neq b'} \sum_{n' \in \mathcal{N}_{b'}^{(k)}} (g_{nbn'b'} + \phi_{nbn'b'} - x_{n'b'}) - |\mathcal{B}_d| + 2 \quad \forall b \in \mathcal{B}_d; n \in \mathcal{N}_b^{(k)} \tag{5.21}$$

$$2w_{nbn'b'} \leq g_{nbn'b'} + \phi_{nbn'b'} \quad \forall b, b' \in \mathcal{B}_d; n \in \mathcal{N}_b^{(k)}; n' \in \mathcal{N}_{b'}^{(k)} \tag{5.22}$$

$$q^{MAX} \geq f_{n'b'} - z_{nb} - M(1 - w_{nbn'b'}) \quad \forall b, b' \in \mathcal{B}_d; n \in \mathcal{N}_b^{(k)}; n' \in \mathcal{N}_{b'}^{(k)} \tag{5.23}$$

$$w_{nbn'b} = 0 \quad \forall b \in \mathcal{B}_d; n, n' \in \mathcal{N}_b^{(k)} | n \neq n' \tag{5.24}$$

$$x_{nb} \in \{0,1\} \quad \forall b \in \mathcal{B}_d; n \in \mathcal{N}_b^{(k)} \tag{5.25}$$

$$g_{nbn'b'}, \phi_{nbn'b'} \in \{0,1\} \quad \forall b, b' \in \mathcal{B}_d | b \neq b'; n \in \mathcal{N}_b^{(k)}; n' \in \mathcal{N}_{b'}^{(k)} \tag{5.26}$$

$$w_{nbn'b'} \in \{0,1\} \quad \forall b, b' \in \mathcal{B}_d; n \in \mathcal{N}_b^{(k)}; n' \in \mathcal{N}_{b'}^{(k)} \tag{5.27}$$

$$z_{nb} \in [0, L + O - \bar{D}_b] \quad \forall b \in \mathcal{B}_d; n \in \mathcal{N}_b^{(k)} \tag{5.28}$$

$$f_{nb} \in [0, L+O] \quad \forall b \in \mathcal{B}_d; n \in \mathcal{N}_b^{(k)} \tag{5.29}$$

$$o_b \in [0, O] \quad \forall b \in \mathcal{B}_d \tag{5.30}$$

$$l \in [0, (O+L) \cdot |\mathcal{B}_d|] \tag{5.31}$$

$$v_b \in [0, L+O] \quad \forall b \in \mathcal{B}_d \tag{5.32}$$

$$\eta, q^{MAX} \geq 0 \tag{5.33}$$

The objective function (5.4) minimizes the cost of performing and postponing surgeries, and the expected total cost related to OR overtime, idle time, surgery waiting times, and cancellation of scheduled surgeries. Constraint (5.5) defines an estimation of the cost of performing surgeries, including the cost of idle time, overtime, and the waiting time of emergency surgeries. Constraint (5.6) is the symmetry-breaking constraint, which ensures that the surgery with a lower ranking will be postponed first. Constraints (5.7) and (5.8) compute the finish time of each surgery. Constraint (5.9) ensures that a surgery is scheduled if and only if the surgery is assigned. Constraint (5.10) computes the start time of each surgery. Constraint (5.11) computes the finish time of each block. Constraint (5.12) computes the total idle time. Constraint (5.13) computes the overtime for each block. Constraints (5.14) - (5.23) are related to the BIM. Constraint (5.14) ensures that if the start time of surgery (n, b) is no less than the start time of (n', b') then the variable $g_{nbn'b'}$ is set to 1. Constraint (5.16) ensures that if the finish time of surgery (n', b') is no less than the start time of surgery (n', b'), then the variable $\phi_{nbn'b'}$ is set to 1. Constraints (5.18) and (5.19) ensure that the variables $g_{nbn'b'}$ and $\phi_{nbn'b'}$ can measure the overlap between two surgeries if and only if these two surgeries are operated. Constraint (5.20) states that there is only one surgery (n', b') whose finish time is the closest to the start time of surgery (n, b) if and only if the surgeries (n, b) is performed. Constraint (5.21) ensures that there is a BII whose start time equals the start time of surgery (n, b) if all ORs are occupied at the beginning of the surgery (n, b). Constraint (5.22) ensures that for surgery (n', b'), which finish time is the closest to the start time of surgery (n, b), its finish time must be strictly later than the start time of surgery (n, b). Constraint (5.23) computes the largest BII. Moreover, constraint (5.24) sets $w_{nbn'b'}$ to 0 when two distinct surgeries operated in the same block. This is because the surgery (n', b), whose finish time is the closest one and strictly later than the start time of surgery (n, b) in the same block b, corresponds to surgery (n, b) itself. Finally, constraints (5.25)-(5.33) define the domain of the variables. Parameter ϵ is used to avoid the numerical issue when the finish time of surgery (n, b) is equal to the start time of surgery (n', b').

Given k-th assignment, the $ESSP_{B\&B}^{(k,j)}$ solution in iteration j is denoted as $(\boldsymbol{x}^{(k,j)}, \boldsymbol{f}^{(k,j)})$. The set of surgeries that $ESSP_{B\&B}^{(k,j)}$ assigns to surgery block b are denoted as $\bar{N}_b^{(k,j)}$. Stopping criteria for the $ESSP_{B\&B}^{(k,j)}$ are same to the $ESAP_{REC}^{(k)}$. The difference is that the maximum time per iteration is set to be 20% of the total remaining CPU time.

5.2.4 Discrete-event simulation algorithm for surgery rescheduling

As we mentioned in Section 4.3.2, five types of events may occur during the surgery process, and it is necessary to reschedule the surgeries when the *Late finish* and *Emergency arrive* events occur. Thus, to evaluate the performance of the initial surgery plan, we use a discrete-event simulation (DES) algorithm to simulate the dynamic surgery rescheduling process.

Algorithm 3 presents the framework of the DES algorithm to evaluate a given initial surgery plan $\mathcal{L}_d = \{(i, B_i, Z_i, F_i) | i \in \mathcal{I}_d^{(k,j)}\}$ for the surgery day d in a given scenario s. After initializing the necessary parameters, it creates a future event list FEL according to the initial surgery plan $\mathcal{L}_d$, arrival time of emergency surgeries A_{is} and surgery duration D_{is}. The core of the DES is to maintain this FEL, which consists of a sequence of event notices ordered by nondecreasing simulation time. The event notices here are the descriptions of future events, and each of them includes the surgery ID, surgery block ID, event type, and the time at which it will occur, denoted as (i, B, ET, T). Noting that any elective surgery i, with planned finish time $F_i > O + L$, will be canceled in the surgery plan $\mathcal{L}_d^{(n)}$ and the corresponding event notices will not be generated in the FEL. After initialization, the DES starts to handle the event notices in the FEL one by one in order. To handle an event notice, the adopted operation depends on the event type ET, which can described as follows. If the event type is *Surgery start*, the corresponding surgery i will be added to the set $\hat{\mathcal{I}}_{ds}$ and its actual start time $\hat{Z}_i$ will be recorded. If the event type is *Finish on time* or *Early finish*, the actual finish time $\hat{F}_i$ of the corresponding surgery i will be recorded. If the event type is *Late finish* or *Emergency arrive*, the rescheduling procedure will be triggered to generate a new surgery plan $\mathcal{L}_d^{(n+1)}$, and based on which a new FEL will be created. Moreover, the handled event notices will be removed from the FEL. The above procedures are terminated when all events in the future event list are processed ($FEL = \emptyset$). The last step of the DES is to calculate the value of $\hat{O}_{ds}$ and $\hat{L}_{ds}$ according to the $\hat{Z}_{is}$ and $\hat{F}_{is}$, for $i \in \hat{\mathcal{I}}_{ds}$.

It is worth noting that the DES algorithm is the most frequently called component within the TPSO algorithm, and each invocation instigates the rescheduling procedure multiple times. Thus, the computing efficiency of rescheduling surgeries is critical to the overall efficiency of the proposed TPSO algorithm. We adopt simple rescheduling rules to quickly reschedule the unstarted surgeries. The rescheduling rule is as follows:

Algorithm 3: Outline of the DES algorithm

Input : A given problem instance NP, a scenario s, a surgery day d, an initial surgery plan $\mathcal{L}_d$, arrive time A_{is} for each emergency surgery i, surgery duration D_{is} for each surgery i.

Output : Set of actual performed elective surgeries $\hat{\mathcal{I}}_{ds}$, actual overtime $\hat{O}_{ds}$ and idle time $\hat{L}_{ds}$, actual start time $\hat{Z}_{is}$ and finish time $\hat{F}_{is}$ for each surgery i.

Initialize the set of the performed surgeries $\hat{\mathcal{I}}_{ds} \leftarrow \emptyset$, the current elective surgery plan $\mathcal{L}_d^{(0)} \leftarrow \mathcal{L}_d$, set the iteration counter $n \leftarrow 0$;
$FEL = \{(i_1, B_1, ET_1, T_1), ..., (i_n, B_n, ET_n, T_n)\} \leftarrow$CreateFutureEventList($\mathcal{L}_d^{(n)}$, D_{is}, A_{is});
while $FEL \neq \emptyset$ **do**
    Select the first event notice (i_1, B_1, ET_1, T_1) from FEL;
    if ET_1 = *Surgery start* **then**
        $\hat{\mathcal{I}}_{ds} \leftarrow \hat{\mathcal{I}}_{ds} \cup \{i_1\}$;
        $\hat{Z}_{is} = T_1$;
    else if $ET_1 \in$ *{Finish on time, Early finish}* **then**
        $\hat{F}_{is} = T_1$;
    if $ET_1 \in$ *{Late finish, Emergency arrive}* **then**
        $\mathcal{L}_d^{(n+1)} \leftarrow$ Rescheduling($NP, \mathcal{L}_d^{(n)}, D_{is}, A_{is}$);
        $FEL \leftarrow$CreateFutureEventList($\mathcal{L}_d^{(n+1)}$, D_{is}, A_{is});
        $n \leftarrow n + 1$;
    $FEL \leftarrow FEL \setminus \{(i_1, B_1, ET_1, T_1)\}$;
Calculate $\hat{O}_{ds}$ and $\hat{L}_{ds}$ according to $\hat{Z}_{is}$ and $\hat{F}_{is}$, $\forall i \in \hat{\mathcal{I}}_{ds}$;

Rule 1 For emergency surgeries, they will wait and be inserted into the surgery block with the earliest available BIM, when the *Emergency arrive* event occurs. Note that the emergency surgeries will not be canceled even if the operating time exceeds the maximum allowable time of the surgery block.

Rule 2 For elective surgeries, they will be postponed if their planned start time is later than the actual end time of the preceding surgeries. Furthermore, any surgery whose planned finish time is beyond the maximum allowable time of the surgery block will be canceled.

After evaluating the given initial surgery plan $\mathcal{L}_d^{(k,j)}$ in all scenarios, the expected total costs related to idle time, overtime, tardiness, and cancellation of elective surgeries, and

waiting times of emergency surgery of this plan can be calculated as follows:

$$\hat{\eta}_d^{(k,j)} = \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\left(\sum_{i\in\mathcal{I}_d^{(k,j)}\setminus\hat{\mathcal{I}}_{ds}} C_i^Q + C^O\hat{O}_{ds}^{(k,j)} + C^L\hat{L}_{ds}^{(k,j)} + \sum_{i\in\hat{\mathcal{I}}_d} C_i^T|\hat{Z}_{is}^{(k,j)} - Z_i^{(k,j)}| + \sum_{i\in\mathcal{E}_{ds}} C^W(\hat{Z}_{is}^{(k,j)} - A_{is})\right) \tag{5.34}$$

5.2.5 Feedback mechanism

In order to accurately estimate the objective value of the obtained solutions in the subsequent iterations, two types of feedback are developed, including sequence feedback for $ESSP_{B\&B}$ and assignment feedback for $ESAP_{REC}$.

Sequence feedback

Given the j-th iteration, the $ESSP_{B\&B}^{(k,j)}$ assigns a certain number of surgeries to surgery block b, denoted as $\dot{\mathcal{N}}_b^{(k,j)}$, with a simulated cost value of $\hat{\eta}_d^{(k,j)}$. To formulate the sequence feedback constraints, we introduce binary variables and constraints. For variable, θ_j, has value 1 if the new assignment is the same as the previous assignment j, and 0 otherwise. Big M parameters are introduced to ensure the constraints are satisfied. Therefore, the sequence feedback constraints can be formulated as follows:

$$\eta \geq \sum_{b\in\mathcal{B}_d} C^O o_b + C^I l + C^W q^{MAX} - \sum_{j=1}^{J} M\theta_j \tag{5.35}$$

$$\eta \geq \hat{\eta}_d^{(k,j)} - M(1-\theta_j) \quad j = 1,2,...,J \tag{5.36}$$

$$\theta_j \geq 1 - \left(\sum_{b\in\mathcal{B}_d}\sum_{n\in\dot{\mathcal{N}}_b^{(k,j)}}(1-x_{nb}) + \sum_{b\in\mathcal{B}_d}\sum_{n=\mathcal{N}_b^{(k,j)}\setminus\dot{\mathcal{N}}_b^{(k,j)}} x_{nb}\right) \quad j = 1,2,...,J \tag{5.37}$$

$$M(1-\theta_j) \geq \sum_{b\in\mathcal{B}_d}\sum_{n\in\dot{\mathcal{N}}_b^{(k,j)}}(1-x_{nb}) + \sum_{b\in\mathcal{B}_d}\sum_{n=\mathcal{N}_b^{(k,j)}\setminus\dot{\mathcal{N}}_b^{(k,j)}} x_{nb} \quad j = 1,2,...,J \tag{5.38}$$

$$\theta_j \in \{0,1\} \quad j = 1,2,...,J \tag{5.39}$$

Constraints (5.35) and (5.36) are either-or constraints, which estimate the cost of the assignment. Note that constraint (5.5) is replaced by constraint (5.35). Constraints (5.37) and (5.38) ensure that the binary variable θ_j is set to 1 if the assignment is the same as

the previous, and 0 otherwise. Constraint (5.39) defines the variable domain.

Assignment feedback

The goals of assignment feedback are reducing the search space and improving the accuracy of the objective value of the obtained solution. Since the $ESSP_{B\&B}$ allows postponing some assigned surgeries of the $ESAP_{REC}$ to the next planning horizon, we consider the best reassignment $\mathcal{I}_b^{(k)'}$ obtained by the $ESSP_{B\&B}$ after multiple iterations as the best subset of the given assignment $\mathcal{I}_b^{(k)}$. Thus, we propose feedback constraints (5.40) and (5.41) to only retain the best subset $\dot{\mathcal{I}}_b^{(k)}$, and remove all other subsets of the given assignment $\widetilde{\mathcal{I}}_b^{(k)}, \widetilde{\mathcal{I}}_b^{(k)} \subset \mathcal{I}_b^{(k)} \wedge \widetilde{\mathcal{I}}_b^{(k)} \neq \mathcal{I}_b^{(k)'}$ to reduce the search space. Considering the ability of constraints (5.40) and (5.41), we refer to them as the local best assignment (LBA) feedback constraints.

$$\sum_{(i,b)\in\mathcal{H}_d^{(k)}\setminus\dot{\mathcal{H}}_d^{(k)}} x_{ib} - \sum_{(i,b)\in\dot{\mathcal{H}}_d^{(k)}} \left(\left|\mathcal{H}_d^{(k)} \setminus \dot{\mathcal{H}}_d^{(k)}\right| + 1\right)(1 - x_{ib}) \leq \sum_{(i,b)\in\mathcal{H}_d\setminus\mathcal{H}_d^{(k)}} |\mathcal{H}_d|^2 x_{ib} \tag{5.40}$$
$$\forall d \in \mathcal{D}; k = 1, 2, ..., K$$

$$\sum_{(i,b)\in\mathcal{H}_d^{(k)}\setminus\dot{\mathcal{H}}_d^{(k)}} x_{ib} - \sum_{(i,b)\in\dot{\mathcal{H}}_d^{(k)}} \left(\left|\mathcal{H}_d^{(k)} \setminus \dot{\mathcal{H}}_d^{(k)}\right| + 1\right)(1 - x_{ib}) \geq - \sum_{(i,b)\in\mathcal{H}_d\setminus\mathcal{H}_d^{(k)}} |\mathcal{H}_d|^2 x_{ib} \tag{5.41}$$
$$\forall d \in \mathcal{D}; k = 1, 2, ..., K$$

where $\mathcal{H}_d = \{(i,b)|b \in \mathcal{B}_d, i \in \mathcal{I}_b\}$, $\mathcal{H}_d^{(k)} = \{(i,b)|b \in \mathcal{B}_d, i \in \mathcal{I}_b^{(k)}\}$ and $\dot{\mathcal{H}}_d^{(k)} = \{(i,b)|b \in \mathcal{B}_d, i \in \dot{\mathcal{I}}_b^{(k)}\}$. $\dot{\mathcal{I}}_b^{(k)} \subseteq \mathcal{I}_b^{(k)}$ is the best subset of the assigned surgeries of the $ESAP_{REC}$ in the k-th iteration. The validation of the LBA feedback constraints is shown in Theorem 5 (see the proof A.5).

Theorem 5 *The LBA feedback constraints (5.40) and (5.41) are valid.*

Similar to the sequence feedback, given the k-th assignment, we define new binary variables and constraints to correct the cost estimation $\hat{\eta}_d^{(k)} = \min_{j=0}^{J} \hat{\eta}_d^{(k,j)}$ related to the surgery assignment $\dot{\mathcal{H}}_d^{(k)}$ which has been simulated in the previous iteration. The definition of variables θ_{kd} are similar to θ_j, and constraints (5.42)-(5.46) are similar to constraints (5.35)-(5.39). Note that constraint (5.2) is replaced by constraint (5.42).

$$\eta_d \geq \sum_{s\in\mathcal{S}} P(s)\left(\sum_{i\in\mathcal{I}}\sum_{b\in\mathcal{B}_d|i\in\mathcal{I}_b} C_i^Q u_{ibs} + \sum_{b\in\mathcal{B}_d} C^O o_{bs}\right) - \sum_{k=1}^{K} M\theta_{kd} \quad \forall d\in\mathcal{D} \tag{5.42}$$

$$\eta_d \geq \hat{\eta}_d^{(k)} - M\left(1-\theta_{kd}\right) \quad \forall d\in\mathcal{D}; k=1,2,...,K \tag{5.43}$$

$$\theta_{kd} \geq 1 - \left(\sum_{(i,b)\in\dot{\mathcal{H}}_d^{(k)}} (1-x_{ib}) + \sum_{(i,b)\in\mathcal{H}_d\setminus\dot{\mathcal{H}}_d^{(k)}} x_{ib}\right) \quad \forall d\in\mathcal{D}; k=1,2,...,K \tag{5.44}$$

$$M(1-\theta_{kd}) \geq \sum_{(i,b)\in\dot{\mathcal{H}}_d^{(k)}} (1-x_{ib}) + \sum_{(i,b)\in\mathcal{H}_d\setminus\dot{\mathcal{H}}_d^{(k)}} x_{ib} \quad \forall d\in\mathcal{D}; k=1,2,...,K \tag{5.45}$$

$$\theta_{kd} \in \{0,1\} \quad \forall d\in\mathcal{D}; k=1,2,...,K \tag{5.46}$$

5.3 Computational experiments

In this section, we use instances same as Section 4.4.1 for our experiments and compare the performance of three approaches, including our TPSO approach and the SAA approach proposed in Section 4. Our TPSO approach was coded in Python, and the Gurobi Optimizer 9.0.3 was called to solve the model. The maximal number of iterations $MaxIter$ is set to 10. The time limits for our TPSO approach are set to be the same as the SAA approach, which are 1h, 5h, 10h and 10h for the instances with 70, 100, 140 and 200 surgeries, respectively. Moreover, the number of scenarios is also set to be the same as the SAA approach, which are 50 for SDR = 0.19 and 0.32, 40 for SDR = 0.75 and 0.96, 15 for SDR = 1.06, 5 for SDR = 1.34, 10 for SDR = 1.49, 40 for SDR = 1.92, 50 for SDR = 2.13 and 2.70.

5.3.1 Experimental results and comparisons

To assess the performance of our TPSO approach, we carried out computational experiments on the 36 instances, comparing it with the SAA approach. The SAA approach is implemented by solving the $SSFU_{SB}$ model using the Gurobi Optimizer 9.0.3 with the same number of scenarios and the same time limit as the TPSO approach. To fairly compare the two approaches, the obtained solutions are evaluated by simulating the surgery schedule in 500 scenarios. The simulation process is the same as the DES algorithm in Section 5.2.4.

Figure 5.2 displays box plots and median values for the gap between the best objective

value obtained by the TPSO and SAA approaches for different SDRs. In detail, the gap is calculated as $Gap = (Obj_M - Obj^*)/Obj^* \times 100\%$ where Obj^* is the best objective value obtained by the TPSO and SAA approaches, and Obj_M is either the best objective value obtained by the TPSO or SAA approach. The results highlight the sensitivity of the SAA approaches to different SDRs, while the TPSO approach is more robust. In particular, the majority of solutions found by the TPSO approach yield values within 0.30% of the best solution for low SDR, 0.40% for medium SDR, and 0.00% for high SDR. In contrast, the SAA approach yields values within 2.60% of the best solution for low SDR, 7.40% for medium SDR, and 3.80% for high SDR. Moreover, the median gaps of the TPSO approach are significantly lower than that of the SAA approach for all SDRs, such as 0.00% vs. 0.36% for low SDR, 0.00% vs. 1.97% for medium SDR, and 0.00% vs. 3.29% for high SDR. Lastly, for the medium SDR, the worst-case gap of 13.42% for the SAA approach is significantly higher than that of 0.77% for the TPSO approach. In conclusion, the surgical plans obtained by our proposed TPSO approach can perform a larger number of surgeries at a lower cost, which is superior to the traditional SAA approach.

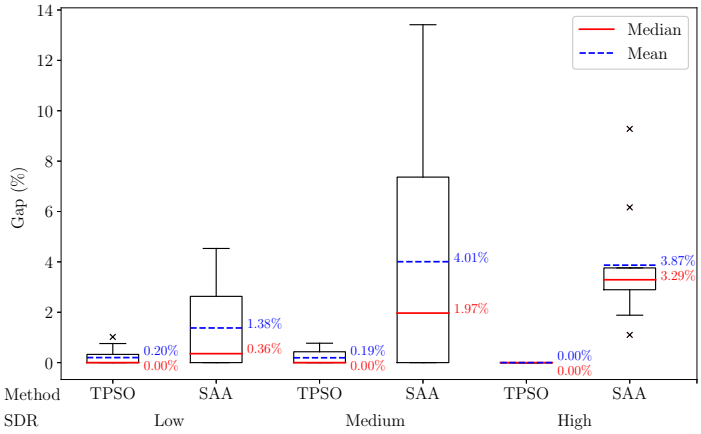
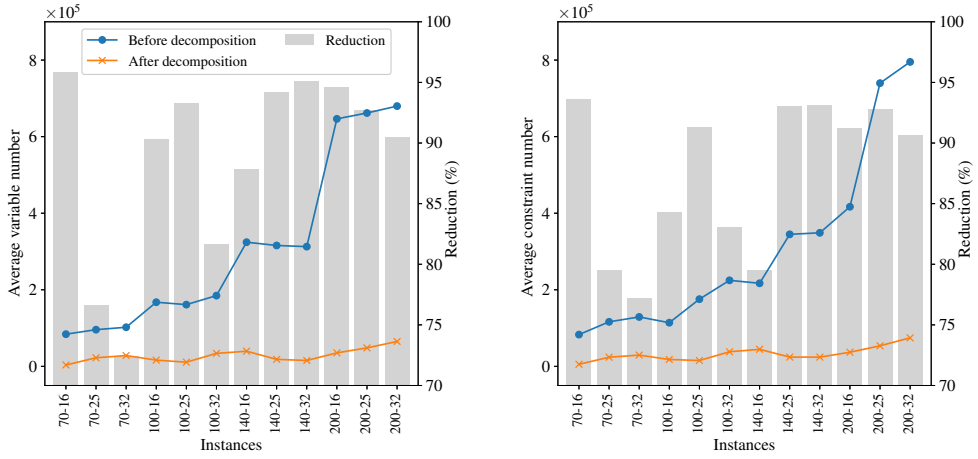


Figure 5.2 – Computational results of TPSO and SAA.

5.3.2 Analysis of decomposition and feedback mechanisms

The consistent performance of the TPSO approach as the SDR value increases is somehow due to the decomposition of the SSFU problem into $ESAP_{REC}$ and $ESSP_{B\&B}$ subproblems. In general, a smaller model (with fewer variables and constraints) would be easier to solve. Thus, we first compare the size of the models before and after decomposition. Figure 5.3 plots the average number of variables and constraints solved by the TPSO

and SAA approaches, along with the percentage of the reduction, as the problem size increases. The significant reduction in variables and constraints reduces the computational complexity of solving the proactive SSFU problem.



(a) Variable reduction **(b)** Constraint reduction

Figure 5.3 – Model size reduction via decomposition.

Apart from the decomposition of the proactive SSFU problem, the feedback mechanism in the TPSO approach plays a crucial role in enhancing the quality of the solution. To confirm this, we additionally perform experiments to compare the proposed TPSO approach, denoted as V0, with the following two variants: Variant 1, denoted as V1, without feedback mechanism, which solves SSFU problem in three subproblems; Variant 2, denoted as V2, which replaces the constraints (5.40) and (5.41) with the classical no-good cut [139]. Figure 5.4 shows the box plots for the gap between the best objective value obtained by the TPSO and its variants, as well as the SAA approach for different SDRs. In general, we observe that the average gap of V0 is the smallest, which indicates that the feedback constraints (5.40) and (5.41) are effective. Moreover, the average gap of V1 is slightly smaller than that of V2 in the instances with low and medium SDR. This is because, in these cases, the demand is less than the supply, and the cost estimation in $ESAP_{REC}$ and $ESSP_{B\&B}$ is relatively accurate. The information provided by the classical no good cut is insufficient to improve the solution quality. However, in the instances with high SDR, the demand exceeds the supply, and the cost estimation is inaccurate. Thus, the no-good cut can provide information to improve the quality of the solution.

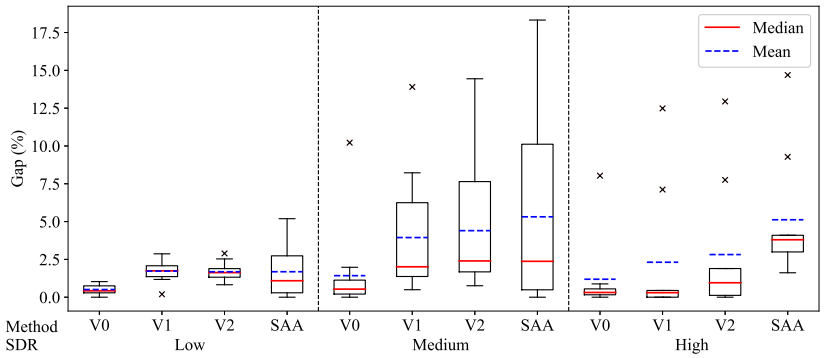


Figure 5.4 – Comparison of different variants of TPSO and SAA.

5.4 Chapter conclusion

In this chapter, we proposed an innovative three-phase simulation-optimization (TPSO) approach to obtain a high-quality initial elective surgery plan, where the result of dynamic rescheduling in the planning horizon is considered. To decrease the complexity of the proactive SSFU problem, we decomposed it into an elective surgery assignment subproblem considering emergency demand ($ESAP_{REC}$) and multiple elective surgery sequencing subproblems with BIMs & buffers ($ESSP_{B\&B}$). We applied a discrete-event simulation (DES) algorithm to evaluate the plan quality. To effectively feedback the evaluation results to $ESAP_{REC}$ and $ESSP_{B\&B}$ models, we developed a set of novel constraints, including sequence feedback and assignment feedback. Especially, for the assignment feedback, we proposed local best assignment (LBA) feedback constraints to reduce the search space, which has been proven to validate.

We conducted extensive computational experiments to evaluate the performance of the proposed TPSO approach. We compared our TPSO approach with the SAA approach proposed in Chapter 4. The results show that our proposed TPSO approach can perform a larger number of surgeries at a lower cost. Moreover, we analyzed the effectiveness of the decomposition and feedback mechanism.

Part III

Conclusions

CONCLUSIONS

This thesis investigates the static and stochastic variants of the well-known patient admission scheduling (PAS) problem and the surgical case scheduling problem in flexible operating rooms under uncertainty (SSFU). As the literature review shown in Chapter 1, for these two NP-hard problems, and many methods have been proposed to solve them. In this thesis, we proposed several advanced mathematical optimization models and solution methods to solve these problems efficiently. Extensive experiments on benchmark instances show the effectiveness of the proposed methods.

In Chapter 2, we focused on how to improve the efficiency of solving the IP model of the static PAS problem using better model formulations. We employed a two-stage exact method that decomposes the PAS problem into two subproblems, including the patient-room assignment (PRA) subproblem and the patient-bed assignment (PBA) subproblem. To solve the PRA subproblem, we applied a warm start approach in which we solve the APRA$^{\mathrm{WT}}$ model to generate a high-quality feasible solution and then use the obtained solution as a warm start to the APRA model. We proposed two aggregated gender policy constraints AGC_0, AGC_1, and aggregated patient transfer constraint ATC, and generated 4 APRA$^{\mathrm{WT}}$ models and 8 APRA models. Experimental results on the 13 benchmark instances in the literature indicate that our method can obtain new improved solutions (new upper bounds) for 6 instances, including one proven optimal solution. In addition, we apply our method to the original PAS problem, which has the maximum number of hard constraints, and perform computational experiments on the same 13 benchmark instances. Our method yields 5 new best solutions and proves optimality for 6 instances.

In Chapter 3, we proposed a new stochastic PAS (SPAS) problem that aims to assign patients to rooms during their planned hospitalization periods while considering the uncertainty of the overstay days. We considered the SPAS problem as a two-stage stochastic programming problem, and built a scenario-based model $SPAS_{SB}$ and its equivalent state-variable model $SPAS_{SV}$. To solve the $SPAS_{SV}$ model efficiently, we elaborated a solution method SAA-SV in which we solve the $SPAS_{SAA}$ model to generate a high-quality feasible solution and then use it as an initial solution to solve the $SPAS_{SV}$ model. The results show that the SAA-SV method effectively improves the solution quality and

computational time. Especially, the *SAA-SV* method is capable of finding solutions with an average optimality gap of 1.73% for large instances reaching 500 patients and 3.3×10^{150} scenarios in 1 hour.

In Chapter 4, we presented a study on SSFU, which consists of operational decisions of assignment (assign patients to OR blocks in a given time horizon) and sequencing (determine the start time of the assigned surgeries in each OR block), while considering the uncertainties associated with the durations of elective surgeries, the arrivals of emergency surgeries, and their durations. To solve this problem, we adopt an integrated proactive and reactive strategy, where a proactive SSFU model is first solved to generate an initial elective surgery plan, and then a reactive SSFU model is solved to dynamically adjust the plan based on actual surgery durations and emergency arrivals. Moreover, we implemented three mechanisms — reserving capacity, Break-In-Moment, and buffer — to improve the robustness of the plan. Extensive computational experiments were conducted to evaluate the performance of the proposed models and mechanisms.

In Chapter 5, we proposed an innovative three-phase simulation-optimization (TPSO) approach to solve the proactive SSFU problem to obtain a high-quality solution, where the result of dynamic rescheduling in the planning horizon is considered. To decrease the complexity of the proactive SSFU problem, we decomposed it into an elective surgery assignment subproblem considering emergency demand ($ESAP_{REC}$) and multiple elective surgery sequencing subproblems with BIMs & buffers ($ESSP_{B\&B}$). We applied a discrete-event simulation (DES) algorithm to evaluate the plan quality. To effectively feedback the evaluation results to $ESAP_{REC}$ and $ESSP_{B\&B}$ models, we developed a set of novel constraints, including sequence feedback and assignment feedback. Especially, for the assignment feedback, we proposed local best assignment (LBA) feedback constraints to reduce the search space, which have been proven to validate. Extensive experiments show the effectiveness of the proposed TPSO approach.

Perspectives

For future research, several directions could be explored.

For the static PAS problem, we can investigate the following directions: (1) Design dedicated branch-and-bound algorithms to improve solving performance while guaranteeing optimality. (2) Design matheuristic algorithms that exploit mathematical programming techniques in a metaheuristic framework. (3) Investigate the dynamic PAS problem by

appropriately adjusting the proposed models to solve real-world situations.

Considering that our proposed SAA-SV method is a general solution framework for two-stage stochastic programming problems, we can investigate its effectiveness in solving the project scheduling problem and the operating room scheduling problem, among others. However, despite its broad applicability, the challenge arises when we solve the state-variable models using general-purpose solvers like CPLEX or Gurobi, especially as the size of the instances increases. To address this issue, designing dedicated Benders decomposition algorithms based on the structure of the state-variable model could be a promising direction.

Moreover, for the SSFU problem, according to the BII existence theorem, we can calculate the length of one BII multiple times, which is inefficient. Therefore, avoiding the repeated calculation of the length of the BII is a potential research direction. Moreover, our modeling approach for the BIMs and buffer mechanism can be extended to other problems, such as the job shop scheduling problem. In addition, the performance of the TPSO approach is significantly dependent on the number of scenarios. To improve the solution performance, we can formulate the scenario-based stochastic programming model $ESAP_{REC}$ to its equivalent state-variable model [98], whose size may be significantly reduced with a pseudo-polynomial number of variables and constraints.

Reference

[1] Fabian Röthlisberger et al., « Challenges and Potential Improvements in the Admission Process of Patients with Spinal Cord Injury in a Specialized Rehabilitation Clinic an Interview Based Qualitative Study of an Interdisciplinary Team », *in*: *BMC Health Services Research* 17.*1* (Dec. 2017), p. 443 (cit. on p. 9).

[2] Berta Ortiga et al., « Standardizing Admission and Discharge Processes to Improve Patient Flow: A Cross Sectional Study », *in*: *BMC Health Services Research* 12.*1* (Dec. 2012), p. 180 (cit. on p. 9).

[3] Eugene Litvak and Maureen Bisognano, « More patients, less payment: increasing hospital efficiency in the aftermath of health reform », *in*: *Health Affairs* 30.*1* (2011), pp. 76–80 (cit. on p. 9).

[4] Somayeh Ghazalbash et al., « Operating Room Scheduling in Teaching Hospitals », *in*: *Advances in Operations Research* 2012 (Mar. 2012), ed. by Ching-Jong Liao, p. 548493 (cit. on p. 9).

[5] E. W. Hans and P. T. Vanberkel, « Operating Theatre Planning and Scheduling », *in*: *Handbook of Healthcare System Scheduling* 168 (2012), pp. 105–130 (cit. on p. 9).

[6] Peter Demeester, Patrick De Causmaecker, and G Vanden Berghe, « Applying a local search algorithm to automatically assign patients to beds », *in*: *Proceedings of the 22nd Conference on Quantitative Methods for Decision Making (Orbel 22)*, 2008, pp. 35–36 (cit. on pp. 9, 16, 46).

[7] Haichao Liu, Yang Wang, and Jin-Kao Hao, « Solving the Patient Admission Scheduling Problem Using Constraint Aggregation », *in*: *European Journal of Operational Research* (2024) (cit. on pp. 9, 11, 64, 75).

[8] Leonardo S.L. Bastos et al., « A mixed integer programming approach to the patient admission scheduling problem », *in*: *European Journal of Operational Research* 273.*3* (2019), pp. 831–840 (cit. on pp. 9, 17, 18, 20, 32, 35, 38, 40, 41, 48, 49, 51–55, 58, 75, 164).

[9] Rosita Guido, Maria Carmela Groccia, and Domenico Conforti, « An Efficient Matheuristic for Offline Patient-to-Bed Assignment Problems », *in*: *European Journal of Operational Research* 268.*2* (July 2018), pp. 486–503 (cit. on pp. 9, 17–20, 35, 41, 49, 51–53, 75, 164).

[10] Troels Martin Range, Richard Martin Lusby, and Jesper Larsen, « A Column Generation Approach for Solving the Patient Admission Scheduling Problem », *in*: *European Journal of Operational Research* 235.*1* (2014), pp. 252–264 (cit. on pp. 9, 17, 18, 20, 25, 55).

[11] Sara Ceschia and Andrea Schaerf, « Local Search and Lower Bounds for the Patient Admission Scheduling Problem », *in*: *Computers & Operations Research* 38.*10* (Oct. 2011), pp. 1452–1463 (cit. on p. 9).

[12] Sara Ceschia and Andrea Schaerf, « Modeling and Solving the Dynamic Patient Admission Scheduling Problem under Uncertainty », *in*: *Artificial Intelligence in Medicine* 56.*3* (Nov. 2012), pp. 199–205 (cit. on pp. 9, 58, 59, 76, 77).

[13] Shuwan Zhu et al., « Operating Room Planning and Surgical Case Scheduling: A Review of Literature », *in*: *Journal of Combinatorial Optimization* 37.*3* (Apr. 2019), pp. 757–805 (cit. on pp. 9, 10, 16, 21).

[14] 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 (Dec. 2015), pp. 52–69 (cit. on pp. 10, 86).

[15] Yang Wang et al., « A Three-Phase Matheuristic Algorithm for the Multi-Day Task Assignment Problem », *in*: *Computers & Operations Research* 159 (2023), p. 106313 (cit. on p. 12).

[16] Vaishali Choudhary et al., « A Comprehensive Review of Patient Scheduling Techniques with Uncertainty », *in*: *Handbook of Formal Optimization*, ed. by Anand J. Kulkarni and Amir H. Gandomi, Singapore: Springer Nature Singapore, 2024, pp. 1–21 (cit. on p. 16).

[17] J. Zhang, M. Dridi, and A. E. Moudni, « An approximate dynamic programming approach to the admission control of elective patients », *in*: *Computers & Operations Research* 132.*12* (2021), p. 105259 (cit. on p. 16).

[18] Zhipeng Lü and Jin Kao Hao, « Adaptive Neighborhood Search for Nurse Rostering », *in*: *European Journal of Operational Research* 218.*3* (2012), pp. 865–876 (cit. on p. 16).

[19] Andreas T Ernst et al., « Staff Scheduling and Rostering: A Review of Applications, Methods and Models », *in*: *European journal of operational research* 153.*1* (2004), pp. 3–27 (cit. on p. 16).

[20] Renato Bruni and Paolo Detti, « A Flexible Discrete Optimization Approach to the Physician Scheduling Problem », *in*: *Operations Research for Health Care* 3.*4* (2014), pp. 191–199 (cit. on p. 16).

[21] Mustafa Demirbilek, Juergen Branke, and Arne Strauss, « Dynamically Accepting and Scheduling Patients for Home Healthcare », *in*: *Health care management science* 22 (2019), pp. 140–155 (cit. on p. 16).

[22] S Ayca Erdogan, Tracey L Krupski, and Jennifer Mason Lobo, « Optimization of telemedicine appointments in rural areas », *in*: *Service Science* 10.*3* (2018), pp. 261–276 (cit. on p. 16).

[23] Zahraa A. Abdalkareem et al., « Healthcare Scheduling in Optimization Context: A Review », *in*: *Health and Technology* 11.*3* (May 2021), pp. 445–469 (cit. on p. 16).

[24] W Vancroonenburg, D Goossens, and F Spieksma, « On the Complexity of the Patient Assignment Problem », *in*: *Tech. rep., KAHO Sint-Lieven, Gebroeders De Smetstraat 1, Gent, Belgium* (2011) (cit. on p. 17).

[25] Peter Demeester et al., « A hybrid tabu search algorithm for automatically assigning patients to beds », *in*: *Artificial Intelligence in Medicine* 48.*1* (2010), pp. 61–70 (cit. on pp. 17, 18, 20, 33, 35, 38, 162).

[26] Abdelaziz Hammouri and Mohammed Alweshah, « Biogeography Based Optimization with Guided Bed Selection Mechanism for Patient Admission Scheduling Problems », *in*: *International Journal of Soft Computing* 12 (2017), pp. 103–111 (cit. on pp. 17, 18, 20).

[27] Sara Ceschia and Andrea Schaerf, « Local search and lower bounds for the patient admission scheduling problem », *in*: *Computers & Operations Research* 38.*10* (2011), pp. 1452–1463 (cit. on pp. 17–20, 32, 35, 37, 38, 41, 51, 54).

[28] Aykut Melih Turhan and Bilge Bilgen, « Mixed integer programming based heuristics for the Patient Admission Scheduling problem », *in*: *Computers & Operations Research* 80 (2017), pp. 38–49 (cit. on pp. 17, 18, 20).

[29] Burak Bilgin et al., « One hyper-heuristic approach to two timetabling problems in health care », *in*: *Journal of Heuristics* 18.*3* (2012), pp. 401–434 (cit. on pp. 18, 20).

[30] Saif Kifah and Salwani Abdullah, « An adaptive non-linear great deluge algorithm for the patient-admission problem », *in*: *Information Sciences* 295 (2015), pp. 573–585 (cit. on pp. 18, 20).

[31] Asaju La'aro Bolaji, Akeem Femi Bamigbola, and Peter Bamidele Shola, « Late Acceptance Hill Climbing Algorithm for Solving Patient Admission Scheduling Problem », *in*: *Knowledge-Based Systems* 145 (2018), pp. 197–206 (cit. on pp. 18, 20).

[32] Abdelaziz I Hammouri and Basem Alrifal, « Investigating biogeography-based optimisation for patient admission scheduling problems. », *in*: *Journal of Theoretical & Applied Information Technology* 70.*3* (2014), pp. 413–421 (cit. on pp. 18, 20).

[33] Abdelaziz I. Hammouri, « A Modified Biogeography-Based Optimization Algorithm with Guided Bed Selection Mechanism for Patient Admission Scheduling Problems », *in*: *Journal of King Saud University - Computer and Information Sciences* 34.*3* (2022), pp. 871–879 (cit. on pp. 18, 20).

[34] Iyad Abu Doush et al., « Harmony Search Algorithm for Patient Admission Scheduling Problem », *in*: *Journal of Intelligent Systems* 29.*1* (2018), pp. 540–553 (cit. on pp. 18, 20).

[35] Asaju La'aro Bolaji et al., « A Room-Oriented Artificial Bee Colony Algorithm for Optimizing the Patient Admission Scheduling Problem », *in*: *Computers in Biology and Medicine* 148 (2022), p. 105850 (cit. on pp. 18, 20).

[36] Zahraa A. Abdalkareem et al., « Discrete Flower Pollination Algorithm for Patient Admission Scheduling Problem », *in*: *Computers in Biology and Medicine* 141 (2022), p. 105007 (cit. on pp. 18, 20).

[37] Sara Ceschia and Andrea Schaerf, « Dynamic Patient Admission Scheduling with Operating Room Constraints, Flexible Horizons, and Patient Delays », *in*: *Journal of Scheduling* 19.*4* (2016), pp. 377–389 (cit. on pp. 19, 20).

[38] Yi-Hang Zhu et al., « Compatibility of Short and Long Term Objectives for Dynamic Patient Admission Scheduling », *in*: *Computers & Operations Research* 104 (2019), pp. 98–112 (cit. on pp. 19, 20).

[39] Rosita Guido, « Patient admission scheduling problems with uncertain length of stay: optimization models and an efficient matheuristic approach », *in*: *International Transactions in Operational Research* (2023) (cit. on pp. 19, 20).

[40] Sara Ceschia and Andrea Schaerf, « Modeling and solving the dynamic patient admission scheduling problem under uncertainty », *in*: *Artificial intelligence in medicine* 56.*3* (2012), pp. 199–205 (cit. on pp. 19, 20).

[41] Richard Martin Lusby et al., « An Adaptive Large Neighborhood Search Procedure Applied to the Dynamic Patient Admission Scheduling Problem », *in*: *Artificial intelligence in medicine* 74 (2016), pp. 21–31 (cit. on pp. 19, 20, 77).

[42] Wim Vancroonenburg, Patrick De Causmaecker, and Greet Vanden Berghe, « A study of decision support models for online patient-to-room assignment planning », *in*: *Annals of Operations Research* 239.*1* (2016), pp. 253–271 (cit. on pp. 19, 20).

[43] Hongru Miao and Jian-Jun Wang, « Scheduling Elective and Emergency Surgeries at Shared Operating Rooms with Emergency Uncertainty and Waiting Time Limit », *in*: *Computers & Industrial Engineering* 160 (Oct. 2021), p. 107551 (cit. on pp. 21, 22).

[44] Najla Omrane Aissaoui, Hejer Hachicha Khlif, and Farah Mansour Zeghal, « Integrated Proactive Surgery Scheduling in Private Healthcare Facilities », *in*: *Computers & Industrial Engineering* 148 (Oct. 2020), p. 106686 (cit. on pp. 21, 22).

[45] Karmel S. Shehadeh and Rema Padman, « Stochastic Optimization Approaches for Elective Surgery Scheduling with Downstream Capacity Constraints: Models, Challenges, and Opportunities », *in*: *Computers & Operations Research* 137 (Jan. 2022), p. 105523 (cit. on pp. 21, 103).

[46] Seokjun Youn, H. Neil Geismar, and Michael Pinedo, « Planning and Scheduling in Healthcare for Better Care Coordination: Current Understanding, Trending Topics, and Future Opportunities », *in*: *Production and Operations Management* 31.*12* (Dec. 2022), pp. 4407–4423 (cit. on p. 21).

[47] Michael Samudra et al., « Scheduling Operating Rooms: Achievements, Challenges and Pitfalls », *in*: *Journal of Scheduling* 19.*5* (Oct. 2016), pp. 493–525 (cit. on p. 21).

[48] Jerrold H. May et al., « The Surgical Scheduling Problem: Current Research and Future Opportunities: The Surgical Scheduling Problem », *in*: *Production and Operations Management* 20.*3* (May 2011), pp. 392–405 (cit. on p. 21).

[49] Brecht Cardoen, Erik Demeulemeester, and Jeroen Beliën, « Operating Room Planning and Scheduling: A Literature Review », *in*: *European Journal of Operational Research* 201.*3* (Mar. 2010), pp. 921–932 (cit. on p. 21).

[50] Mehdi Lamiri et al., « A Stochastic Model for Operating Room Planning with Elective and Emergency Demand for Surgery », *in*: *European Journal of Operational Research* 185.*3* (Mar. 2008), pp. 1026–1037 (cit. on pp. 21, 25, 86).

[51] Jose M. Molina-Pariente, Erwin W. Hans, and Jose M. Framinan, « A Stochastic Approach for Solving the Operating Room Scheduling Problem », *in*: *Flexible Services and Manufacturing Journal* 30.*1-2* (June 2018), pp. 224–251 (cit. on pp. 21, 25).

[52] Jiafu Tang and Yu Wang, « An Adjustable Robust Optimisation Method for Elective and Emergency Surgery Capacity Allocation with Demand Uncertainty », *in*: *International Journal of Production Research* 53.*24* (2015), pp. 7317–7328 (cit. on pp. 21, 25).

[53] Hongru Miao and Jian-Jun Wang, « Distributed Surgical Scheduling across Collaborating Hospitals Considering Stochastic Duration and Emergency Demand », *in*: *Computers & Industrial Engineering* 183 (Sept. 2023), p. 109462 (cit. on pp. 21, 24, 25).

[54] Davide Duma and Roberto Aringhieri, « The Management of Non-Elective Patients: Shared vs. Dedicated Policies », *in*: *Omega* 83 (Mar. 2019), pp. 199–212 (cit. on pp. 22, 24, 25, 86).

[55] Mathieu Vandenberghe et al., « Stochastic Surgery Selection and Sequencing under Dynamic Emergency Break-Ins », *in*: *Journal of the Operational Research Society* 72.*6* (2021), pp. 1309–1329 (cit. on p. 22).

[56] Mathieu Vandenberghe et al., « Surgery Sequencing to Minimize the Expected Maximum Waiting Time of Emergent Patients », *in*: *European Journal of Operational Research* 275.*3* (June 2019), pp. 971–982 (cit. on pp. 22, 25, 100, 101).

[57] Guillermo Latorre-Núñez et al., « Scheduling Operating Rooms with Consideration of All Resources, Post Anesthesia Beds and Emergency Surgeries », *in*: *Computers & Industrial Engineering* 97 (July 2016), pp. 248–257 (cit. on pp. 22, 25, 101, 102, 161, 162).

[58] J.T. van Essen et al., « Minimizing the Waiting Time for Emergency Surgery », *in*: *Operations Research for Health Care* 1.*2-3* (June 2012), pp. 34–44 (cit. on pp. 22, 25, 86, 100, 101).

[59] Arne Schulz and Malte Fliedner, « Minimizing the Expected Waiting Time of Emergency Jobs », *in*: *Journal of Scheduling* 26.*2* (Apr. 2023), pp. 147–167 (cit. on pp. 22, 25, 86, 100, 101, 161).

[60] Jian-Jun Wang et al., « Operating Room Scheduling for Non-Operating Room Anesthesia with Emergency Uncertainty », *in*: *Annals of Operations Research* 321.*1-2* (Feb. 2023), pp. 565–588 (cit. on pp. 22, 24, 25, 101, 102).

[61] Kyung Sung Jung et al., « Scheduling Elective Surgeries with Emergency Patients at Shared Operating Rooms », *in*: *Production and Operations Management* 28.*6* (2019), pp. 1407–1430 (cit. on pp. 22–25, 101–103).

[62] 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 (Dec. 2022), p. 100366 (cit. on pp. 22, 25).

[63] Yao Xiao and Reena Yoogalingam, « Reserved Capacity Policies for Operating Room Scheduling », *in*: *Operations Management Research* 14.*1-2* (June 2021), pp. 107–122 (cit. on pp. 22, 25).

[64] Shing Chih Tsai, Yingchieh Yeh, and Chen Yun Kuo, « Efficient Optimization Algorithms for Surgical Scheduling under Uncertainty », *in*: *European Journal of Operational Research* 293.*2* (Sept. 2021), pp. 579–593 (cit. on pp. 22, 25).

[65] Ergin Erdem, Xiuli Qu, and Jing Shi, « Rescheduling of Elective Patients upon the Arrival of Emergency Patients », *in*: *Decision Support Systems* 54.*1* (2012), pp. 551–563 (cit. on pp. 22, 25).

[66] Roberto Bargetto, Thierry Garaix, and Xiaolan Xie, « Dynamic Insertion of Emergency Surgeries with Different Waiting Time Targets », *in*: *IEEE Transactions on Automation Science and Engineering* 16.*1* (2019), pp. 87–99 (cit. on pp. 23, 25).

[67] Thiago A.O. Silva and Mauricio C. De Souza, « Surgical Scheduling under Uncertainty by Approximate Dynamic Programming », *in*: *Omega* 95 (Sept. 2020), p. 102066 (cit. on pp. 23, 25).

[68] Jian-Jun Wang, Hongru Miao, and Ran Xu, « Surgical Rescheduling Problem with Emergency Patients Considering Participants' Dissatisfaction », *in*: *Soft Computing* 25.*16* (Aug. 2021), pp. 10749–10769 (cit. on pp. 23, 25).

[69] Masoud Eshghali et al., « Machine Learning Based Integrated Scheduling and Rescheduling for Elective and Emergency Patients in the Operating Theatre », *in*: *Annals of Operations Research* 332.*1* (2024), pp. 989–1012 (cit. on pp. 24, 25).

[70] Yu Wang, Jiafu Tang, and Richard Y.K. Fung, « A Column-Generation-Based Heuristic Algorithm for Solving Operating Theater Planning Problem under Stochastic Demand and Surgery Cancellation Risk », *in*: *International Journal of Production Economics* 158 (2014), pp. 28–36 (cit. on p. 25).

[71] Nickolas K. Freeman, Sharif H. Melouk, and John Mittenthal, « A Scenario-Based Approach for Operating Theater Scheduling Under Uncertainty », *in*: *Manufacturing & Service Operations Management* 18.*2* (May 2016), pp. 245–261 (cit. on pp. 25, 75, 77, 83, 101, 102, 104).

[72] Balasubramanian Ram, Mark H. Karwan, and A.J.G. Babu, « Aggregation of Constraints in Integer Programming », *in*: *European Journal of Operational Research* 35.*2* (1988), pp. 216–227 (cit. on pp. 24, 25).

[73] Pierre-Louis Poirion, « Optimal Constraints Aggregation Method for ILP », *in*: *Discrete Applied Mathematics* 262 (2019), pp. 148–157 (cit. on p. 24).

[74] Archana Khurana and Katta G. Murty, « How Effective Is Aggregation for Solving 01 Models? », *in*: *OPSEARCH* 49.*1* (Mar. 2012), pp. 78–85 (cit. on pp. 25, 45).

[75] Bahram Alidaee, « Zero Duality Gap in Surrogate Constraint Optimization: A Concise Review of Models », *in*: *European Journal of Operational Research* 232.*2* (2014), pp. 241–248 (cit. on p. 25).

[76] Andrew C. Trapp and Oleg A. Prokopyev, « A Note on Constraint Aggregation and Value Functions for Two-Stage Stochastic Integer Programs », *in*: *Discrete Optimization* 15 (2015), pp. 37–45 (cit. on pp. 25, 32).

[77] Yuri M. Ermoliev, Arkadii V. Kryazhimskii, and Andrzej Ruszczyski, « Constraint Aggregation Principle in Convex Optimization », *in*: *Mathematical Programming* 76.*3* (1997), pp. 353–372 (cit. on p. 25).

[78] David F. Rogers et al., « Aggregation and Disaggregation Techniques and Methodology in Optimization », *in*: *Operations Research* 39.*4* (1991), pp. 553–582 (cit. on p. 25).

[79] Daniel Porumbel and François Clautiaux, « Constraint Aggregation in Column Generation Models for Resource-Constrained Covering Problems », *in*: *INFORMS Journal on Computing* 29.*1* (Jan. 2017), pp. 170–184 (cit. on p. 25).

[80] Pascal Benchimol, Guy Desaulniers, and Jacques Desrosiers, « Stabilized Dynamic Constraint Aggregation for Solving Set Partitioning Problems », *in*: *European Journal of Operational Research* 223.*2* (2012), pp. 360–371 (cit. on pp. 25, 32).

[81] Issmail Elhallaoui et al., « Dynamic Aggregation of Set-Partitioning Constraints in Column Generation », *in*: *Operations Research* 53.*4* (2005), pp. 632–645 (cit. on p. 25).

[82] James R Evans, « A Network Decomposition/Aggregation Procedure for a Class of Multicommodity Transportation Problems », *in*: *Networks. An International Journal* 13.*2* (1983), pp. 197–205 (cit. on p. 25).

[83] Ke-Shi Zhang et al., « Constraint Aggregation for Large Number of Constraints in Wing Surrogate-Based Optimization », *in*: *Structural and Multidisciplinary Optimization* 59.*2* (2019), pp. 421–438 (cit. on p. 25).

[84] Mohammed Saddoune et al., « Integrated Airline Crew Scheduling: A Bi-Dynamic Constraint Aggregation Method Using Neighborhoods », *in*: *European Journal of Operational Research* 212.*3* (2011), pp. 445–454 (cit. on p. 25).

[85] Issmail Elhallaoui et al., « Bi-Dynamic Constraint Aggregation and Subproblem Reduction », *in*: *Computers & Operations Research* 35.*5* (2008), pp. 1713–1724 (cit. on p. 25).

[86] Fred Glover, « Tutorial on Surrogate Constraint Approaches for Optimization in Graphs », *in*: *Journal of Heuristics* 9 (2003), pp. 175–227 (cit. on p. 25).

[87] Jian Zhang, Mahjoub Dridi, and Abdellah El Moudni, « A Two-Phase Optimization Model Combining Markov Decision Process and Stochastic Programming for Advance Surgery Scheduling », *in*: *Computers & Industrial Engineering* 160 (Oct. 2021), p. 107548 (cit. on pp. 26, 58).

[88] Maryam Khatami et al., « Inpatient discharge planning under uncertainty », *in*: *IISE Transactions* 54.*4* (2022), pp. 332–347 (cit. on pp. 26, 58).

[89] Anne Van Den Broek d'Obrenan et al., « Minimizing Bed Occupancy Variance by Scheduling Patients under Uncertainty », *in*: *European Journal of Operational Research* 286.*1* (Oct. 2020), pp. 336–349 (cit. on p. 26).

[90] Daiki Min and Yuehwern Yih, « Scheduling Elective Surgery under Uncertainty and Downstream Capacity Constraints », *in*: *European Journal of Operational Research* 206.*3* (2010), pp. 642–652 (cit. on pp. 26, 58, 73, 74).

[91] Ridong Wang et al., « Robust Elective Hospital Admissions With Contextual Information », *in*: *IEEE Transactions on Automation Science and Engineering* (2023), pp. 1–19 (cit. on p. 26).

[92] Karmel S. Shehadeh and Rema Padman, « A Distributionally Robust Optimization Approach for Stochastic Elective Surgery Scheduling with Limited Intensive Care Unit Capacity », *in*: *European Journal of Operational Research* 290.*3* (May 2021), pp. 901–913 (cit. on p. 26).

[93] Fanwen Meng et al., « A Robust Optimization Model for Managing Elective Admission in a Public Hospital », *in*: *Operations Research* 63.*6* (Dec. 2015), pp. 1452–1467 (cit. on p. 26).

[94] Álvaro Porras et al., « Tight and Compact Sample Average Approximation for Joint Chance-Constrained Problems with Applications to Optimal Power Flow », *in*: *INFORMS Journal on Computing* 35.*6* (Nov. 2023), pp. 1454–1469 (cit. on pp. 26, 75).

[95] Roya Karimi, Jianqiang Cheng, and Miguel A. Lejeune, « A Framework for Solving Chance-Constrained Linear Matrix Inequality Programs », *in*: *INFORMS Journal on Computing* 33.*3* (July 2021), pp. 1015–1036 (cit. on p. 26).

[96] Renaud Chicoisne, Fernando Ordóñez, and Daniel Espinoza, « Risk Averse Shortest Paths: A Computational Study », *in*: *INFORMS Journal on Computing* 30.*3* (Aug. 2018), pp. 539–553 (cit. on p. 26).

[97] Camilo Mancilla and Robert Storer, « A Sample Average Approximation Approach to Stochastic Appointment Sequencing and Scheduling », *in*: *IIE Transactions* 44.*8* (Aug. 2012), pp. 655–670 (cit. on p. 26).

[98] Hossein Hashemi Doulabi, Shabbir Ahmed, and George Nemhauser, « State-Variable Modeling for a Class of Two-Stage Stochastic Optimization Problems », *in*: *INFORMS Journal on Computing* 34.*1* (Jan. 2022), pp. 354–369 (cit. on pp. 26, 133).

[99] Hossein Hashemi Doulabi and Soheyl Khalilpourazari, « Stochastic weekly operating room planning with an exponential number of scenarios », *in*: *Annals of Operations Research* 328.*1* (2023), pp. 643–664 (cit. on p. 26).

[100] Satyajith Amaran et al., « Simulation Optimization : A Review of Algorithms and Applications », *in*: (2019), pp. 1–28 (cit. on pp. 26, 27).

[101] John R. Birge and François Louveaux, *Introduction to Stochastic Programming*, Springer Series in Operations Research and Financial Engineering, New York, NY: Springer New York, 2011 (cit. on p. 26).

[102] Chenhao Zhou et al., « Classification and Literature Review on the Integration of Simulation and Optimization in Maritime Logistics Studies », *in*: *IISE TRANSACTIONS* 53.*10* (Oct. 2021), pp. 1157–1176 (cit. on p. 27).

[103] Huai Tein Lim and Razamin Ramli, « Recent Advancements of Nurse Scheduling Models and a Potential Path », *in*: *Proceedings of the 6th IMT-GT Conference on Mathematics, Statistics and Its Applications*, Citeseer, 2010, pp. 395–409 (cit. on p. 27).

[104] Vanda De Angelis, Giovanni Felici, and Paolo Impelluso, « Integrating Simulation and Optimisation in Health Care Centre Management », *in*: *European Journal of Operational Research* 150.*1* (Oct. 2003), pp. 101–114 (cit. on p. 27).

[105] Michael Ferris et al., « Breast Cancer Epidemiology: Calibrating Simulations via Optimization », *in*: *Oberwolfach Reports* 2 (Jan. 2005), pp. 9023–9027 (cit. on p. 27).

[106] Damien Ernst et al., « The Cross-Entropy Method for Power System Combinatorial Optimization Problems », *in*: *2007 IEEE Lausanne Power Tech*, 2007, pp. 1290–1295 (cit. on p. 27).

[107] Geiza Cristina Da Silva et al., « The Dynamic Space Allocation Problem: Applying Hybrid GRASP and Tabu Search Metaheuristics », *in*: *Computers & Operations Research* 39.*3* (Mar. 2012), pp. 671–677 (cit. on pp. 51, 53, 164).

[108] Rosita Guido, « Patient Admission Scheduling Problems with Uncertain Length of Stay: Optimization Models and an Efficient Matheuristic Approach », *in*: *International Transactions in Operational Research* 31.*1* (2024), pp. 53–87 (cit. on pp. 58, 77).

[109] Ana Batista, David Pozo, and Jorge Vera, « Managing the Unknown: A Distributionally Robust Model for the Admission Planning Problem under Uncertain Length of Stay », *in*: *Computers & Industrial Engineering* 154 (Apr. 2021), p. 107041 (cit. on p. 58).

[110] Jian Zhang, Mahjoub Dridi, and Abdellah El Moudni, « Column-Generation-Based Heuristic Approaches to Stochastic Surgery Scheduling with Downstream Capacity Constraints », *in*: *International Journal of Production Economics* 229 (Nov. 2020), p. 107764 (cit. on p. 58).

[111] Jian Zhang, Mahjoub Dridi, and Abdellah El Moudni, « A Two-Level Optimization Model for Elective Surgery Scheduling with Downstream Capacity Constraints », *in*: *European Journal of Operational Research* 276.*2* (July 2019), pp. 602–613 (cit. on p. 58).

[112] Wim Vancroonenburg, Patrick De Causmaecker, and Greet Vanden Berghe, « Chance-Constrained Admission Scheduling of Elective Surgical Patients in a Dynamic, Uncertain Setting », *in*: *Operations Research for Health Care* 22 (Sept. 2019), p. 100196 (cit. on p. 58).

[113] Aida Jebali and Ali Diabat, « A Chance-Constrained Operating Room Planning with Elective and Emergency Cases under Downstream Capacity Constraints », *in*: *Computers & Industrial Engineering* 114 (Dec. 2017), pp. 329–344 (cit. on p. 58).

[114] Aida Jebali and Ali Diabat, « A Stochastic Model for Operating Room Planning under Capacity Constraints », *in*: *International Journal of Production Research* 53.*24* (Dec. 2015), pp. 7252–7270 (cit. on p. 58).

[115] Lu He et al., « A Systematic Review of Research Design and Modeling Techniques in Inpatient Bed Management », *in*: *Computers & Industrial Engineering* 127 (Jan. 2019), pp. 451–466 (cit. on p. 58).

[116] Chin-Sheng Yang et al., « Predicting the Length of Hospital Stay of Burn Patients: Comparisons of Prediction Accuracy among Different Clinical Stages », *in*: *Decision Support Systems* 50.*1* (Dec. 2010), pp. 325–335 (cit. on p. 59).

[117] Marianna De Santis et al., « A Penalty Branch-and-Bound Method for Mixed Binary Linear Complementarity Problems », *in*: *INFORMS Journal on Computing* 34.*6* (Nov. 2022), pp. 3117–3133 (cit. on p. 73).

[118] Mingjie Li, Jin-Kao Hao, and Qinghua Wu, « A Flow Based Formulation and a Reinforcement Learning Based Strategic Oscillation for Cross-Dock Door Assignment », *in*: *European Journal of Operational Research* 312.*2* (2024), pp. 473–492 (cit. on p. 74).

[119] Yang Wang et al., « A Three-Phase Matheuristic Algorithm for the Multi-Day Task Assignment Problem », *in*: *Computers & Operations Research* 159 (2023), p. 106313 (cit. on p. 74).

[120] Zhen Shang et al., « Multi-Wave Tabu Search for the Boolean Quadratic Programming Problem with Generalized Upper Bound Constraints », *in*: *Computers & Operations Research* 150 (Feb. 2023), p. 106077 (cit. on p. 74).

[121] Maria Albareda-Sambola, Elena Fernández, and Francisco Saldanha-da-Gama, « Heuristic Solutions to the Facility Location Problem with General Bernoulli Demands », *in*: *INFORMS Journal on Computing* 29.*4* (Nov. 2017), pp. 737–753 (cit. on p. 74).

[122] Hartmut Stadtler and Nikolai Heinrichs, « Multi-Period Descriptive Sampling for Scenario Generation Applied to the Stochastic Capacitated Lot-Sizing Problem », *in*: *OR Spectrum* (Jan. 2024) (cit. on p. 74).

[123] Thomas Reiten Bovim et al., « Stochastic Master Surgery Scheduling », *in*: *European Journal of Operational Research* 285.*2* (Sept. 2020), pp. 695–711 (cit. on p. 74).

[124] Sujin Kim, Raghu Pasupathy, and Shane G Henderson, « A guide to sample average approximation », *in*: *Handbook of simulation optimization* (2015), pp. 207–243 (cit. on p. 74).

[125] Xuewen Mu, Yaling Zhang, and Sanyang Liu, « A New Branch and Bound Method with Pretreatment for the Binary Quadratic Programming », *in*: *Applied Mathematics and Computation* 192.*1* (Sept. 2007), pp. 252–259 (cit. on p. 74).

[126] Sujin Kim and Jong Hyun Ryu, « The Sample Average Approximation Method for Multi-Objective Stochastic Optimization », *in*: *Proceedings - Winter Simulation Conference* 12.*2* (2011), pp. 4021–4032 (cit. on p. 83).

[127] Yann Ferrand, Michael Magazine, and Uday Rao, « Comparing Two Operating-Room-Allocation Policies for Elective and Emergency Surgeries », *in*: *Proceedings of the 2010 Winter Simulation Conference*, 2010, pp. 2364–2374 (cit. on p. 86).

[128] Gerhard Wullink et al., « Closing Emergency Operating Rooms Improves Efficiency », *in*: *Journal of medical systems* 31 (2007), pp. 543–546 (cit. on p. 86).

[129] Jtv Essen et al., « Minimizing the waiting time for emergency surgery », *in*: *Operations Research for Health Care* 1.*2-3* (2012), pp. 34–44 (cit. on p. 90).

[130] Ahmet B. Keha, Ketan Khowala, and John W. Fowler, « Mixed Integer Programming Formulations for Single Machine Scheduling Problems », *in*: *Computers & Industrial Engineering* 56.*1* (Feb. 2009), pp. 357–367 (cit. on p. 100).

[131] Karmel S. Shehadeh, « Data-Driven Distributionally Robust Surgery Planning in Flexible Operating Rooms over a Wasserstein Ambiguity », *in*: *Computers & Operations Research* 146 (Oct. 2022), p. 105927 (cit. on pp. 102, 116).

[132] Carlo Mannino, Eivind J Nilssen, and Tomas Eric Nordlander, « Sintef ict: Mss-adjusts surgery data », `https://www.sintef.no/Projectweb/Health-care-optimization/Testbed/`, 2010 (cit. on p. 102).

[133] Thomas Reiten Bovim et al., « Stochastic Master Surgery Scheduling », *in*: *European Journal of Operational Research* 285.*2* (2020), pp. 695–711 (cit. on p. 103).

[134] Daiki Min and Yuehwern Yih, « Scheduling elective surgery under uncertainty and downstream capacity constraints », *in*: *European Journal of Operational Research* 206.*3* (2010), pp. 642–652 (cit. on pp. 103, 116).

[135] Vahid Roshanaei et al., « Collaborative operating room planning and scheduling », *in*: *INFORMS Journal on Computing* 29.*3* (June 2017), pp. 558–580 (cit. on p. 116).

[136] Bahman Naderi et al., « Increased Surgical Capacity without Additional Resources: Generalized Operating Room Planning and Scheduling », *in*: *Production and operations management* 30.*8* (2021), pp. 2608–2635 (cit. on p. 116).

[137] Aïda Jebali, Atidel B. Hadj Alouane, and Pierre Ladet, « Operating Rooms Scheduling », *in*: *International Journal of Production Economics* 99.*1-2* (Jan. 2006), pp. 52–62 (cit. on p. 116).

[138] Raf Jans, « Solving Lot-Sizing Problems on Parallel Identical Machines Using Symmetry-Breaking Constraints », *in*: *INFORMS Journal on Computing* 21.*1* (2009), pp. 123–136 (cit. on p. 116).

[139] Andre A. Ciré, Elvin Çoban, and John N. Hooker, « Logic-Based Benders Decomposition for Planning and Scheduling: A Computational Analysis », *in*: *Knowledge Engineering Review* 31.*5* (2016), pp. 440–451 (cit. on p. 126).

[140] Peter Demeester et al., « A hyper-heuristic approach to the patient admission scheduling problem », *in*: *35th Annual ORAHS conference*, 2009, pp. 65–65.

[141] Alain Guinet, Nadine Meskens, and Tao Wang, « A Multi-objective Patient Admission Planning Improving Resources Utilisation Under Bed Capacity Constraints », *in*: *Health Care Systems Engineering for Scientists and Practitioners*, Springer, 2016, pp. 13–24.

[142] He He, Hal Daume III, and Jason M Eisner, « Learning to search in branch and bound algorithms », *in*: *Advances in neural information processing systems* 27 (2014), pp. 3293–3301.

[143] Stefano Lucidi et al., « A derivative-free approach for a simulation-based optimization problem in healthcare », *in*: *Optimization Letters* 10.*2* (2016), pp. 219–235.

[144] Gregor Hendel et al., « Estimating the Size of Branch-and-Bound Trees », *in*: *INFORMS Journal on Computing* 34.*2* (2022), pp. 934–952.

[145] M. Davidson, « PrimalDual Constraint Aggregation with Application to Stochastic Programming », *in*: *Annals of Operations Research* 99.*1* (2000), pp. 41–58.

[146] Daniel Villeneuve, *Logiciel de Génération de Colonnes*, École Polytechnique de Montréal, 1999.

[147] Denis C. Onyekwelu, « Technical NoteComputational Viability of a Constraint Aggregation Scheme for Integer Linear Programming Problems », *in*: *Operations Research* 31.*4* (Aug. 1983), pp. 795–801.

[148] Gianmauro Numico et al., « The Hospital Care of Patients with Cancer: A Retrospective Analysis of the Characteristics of Their Hospital Stay in Comparison with Other Medical Conditions », *in*: *European Journal of Cancer* 139 (Nov. 2020), pp. 99–106.

[149] Gianmauro Numico et al., « Organizational Determinants of Hospital Stay: Establishing the Basis of a Widespread Action on More Efficient Pathways in Medical Units », *in*: *Internal and Emergency Medicine* 15.*6* (Sept. 2020), pp. 1011–1019.

[150] Jingui Xie et al., « The Analytics of Bed Shortages: Coherent Metric, Prediction, and Optimization », *in*: *Operations Research* 71.*1* (2023), pp. 23–46.

[151] Ana Batista, David Pozo, and Jorge Vera, « Managing the Unknown: A Distributionally Robust Model for the Admission Planning Problem under Uncertain Length of Stay », *in*: *Computers & Industrial Engineering* 154 (2021), p. 107041.

[152] Daniel Baena, Jordi Castro, and Antonio Frangioni, « Stabilized benders methods for large-scale combinatorial optimization, with application to data privacy », *in*: *Management Science* 66.*7* (2020), pp. 3051–3068.

[153] Hossein Hashemi Doulabi et al., « Exploiting the structure of two-stage robust optimization models with exponential scenarios », *in*: *INFORMS Journal on Computing* 33.*1* (2021), pp. 143–162.

[154] Jonathan E. Helm, Shervin AhmadBeygi, and Mark P. Van Oyen, « Design and Analysis of Hospital Admission Control for Operational Effectiveness », *in*: *Production and Operations Management* 20.*3* (May 2011), pp. 359–374.

[155] Jiekun Feng and Pengyi Shi, « Steady-state Diffusion Approximations for Discrete-time Queue in Hospital Inpatient Flow Management », *in*: *Naval Research Logistics (NRL)* 65.*1* (Feb. 2018), pp. 26–65.

[156] J. G. Dai and Pengyi Shi, « Inpatient Overflow: An Approximate Dynamic Programming Approach », *in*: *Manufacturing & Service Operations Management* 21.*4* (Oct. 2019), pp. 894–911.

[157] Yanping Jiang et al., « Admission Control of Hospitalization with Patient Gender by Using Markov Decision Process », *in*: *International Transactions in Operational Research* 30.*1* (Jan. 2023), pp. 70–98.

[158] Mao-Te Chuang, Ya-han Hu, and Chia-Lun Lo, « Predicting the Prolonged Length of Stay of General Surgery Patients: A Supervised Learning Approach », *in*: *International Transactions in Operational Research* 25.*1* (Jan. 2018), pp. 75–90.

[159] Aya Awad, Mohamed BaderElDen, and James McNicholas, « Patient Length of Stay and Mortality Prediction: A Survey », *in*: *Health Services Management Research* 30.*2* (May 2017), pp. 105–120.

[160] Fei Ma et al., « Length-of-Stay Prediction for Pediatric Patients With Respiratory Diseases Using Decision Tree Methods », *in*: *IEEE Journal of Biomedical and Health Informatics* 24.*9* (Sept. 2020), pp. 2651–2662.

[161] Junde Chen et al., « A Deep Learning Approach for Inpatient Length of Stay and Mortality Prediction », *in*: *Journal of Biomedical Informatics* 147 (Nov. 2023), p. 104526.

[162] Yi-Hang Zhu et al., « Compatibility of Short and Long Term Objectives for Dynamic Patient Admission Scheduling », *in*: *Computers & Operations Research* 104 (Apr. 2019), pp. 98–112.

[163] Aykut Melih Turhan and Bilge Bilgen, « Mixed Integer Programming Based Heuristics for the Patient Admission Scheduling Problem », *in*: *Computers and Operations Research* 80 (2017), pp. 38–49.

[164] Wim Vancroonenburg, Patrick De Causmaecker, and Greet Vanden Berghe, « A Study of Decision Support Models for Online Patient-to-Room Assignment Planning », *in*: *Annals of Operations Research* 239.*1* (Apr. 2016), pp. 253–271.

[165] R. Henrion and W. Römisch, « Problem-Based Optimal Scenario Generation and Reduction in Stochastic Programming », *in*: *Mathematical Programming* 191.*1* (Jan. 2022), pp. 183–205.

[166] M.A Borg, « Bed Occupancy and Overcrowding as Determinant Factors in the Incidence of MRSA Infections within General Ward Settings », *in*: *Journal of Hospital Infection* 54.*4* (2003), pp. 316–318.

[167] Hadhemi Saadouli et al., « A Stochastic Optimization and Simulation Approach for Scheduling Operating Rooms and Recovery Beds in an Orthopedic Surgery Department », *in*: *Computers & Industrial Engineering* 80 (2015), pp. 72–79.

[168] Matteo Fischetti and Michele Monaci, « Exploiting Erraticism in Search », *in*: *Operations Research* 62.*1* (2014), pp. 114–122.

[169] Cristian Ramírez-Pico, Ivana Ljubi, and Eduardo Moreno, « Benders Adaptive-Cuts Method for Two-Stage Stochastic Programs », *in*: *Transportation Science* 57.*5* (Sept. 2023), pp. 1252–1275.

[170] G. Somayeh et al., « Operating Room Scheduling in Teaching Hospitals », *in*: *Advances in Operations Research* 2012 (2012), p. 16.

[171] Thiago A.O. Silva et al., « Surgical scheduling with simultaneous employment of specialised human resources », *in*: *European Journal of Operational Research* 245.*3* (2015), pp. 719–730.

[172] W. M. Hancock et al., « Operating room scheduling data base analysis for scheduling », *in*: *Journal of Medical Systems* 12.*6* (1988), pp. 397–409.

[173] David P Strum, Jerrold H May, and Luis G Vargas, « Modeling the uncertainty of surgical procedure times: comparison of log-normal and normal models », *in*: *The Journal of the American Society of Anesthesiologists* 92.*4* (2000), pp. 1160–1167.

[174] Serhat Gul et al., « Bi-criteria scheduling of surgical services for an outpatient procedure center », *in*: *Production and Operations management* 20.*3* (2011), pp. 406–417.

[175] Carlo Mannino, Eivind J Nilssen, and Tomas Eric Nordlander, « Sintef ict: Mss-adjusts surgery data », https://www.sintef.no/Projectweb/Health-care-optimization/Testbed/, 2010.

[176] Serhat Gul, « A Stochastic Programming Approach for Appointment Scheduling Under Limited Availability of Surgery Turnover Teams », *in*: *Service Science* 10.*3* (Sept. 2018), pp. 277–288.

[177] Dinh Nguyen Pham and Andreas Klinkert, « Surgical Case Scheduling as a Generalized Job Shop Scheduling Problem », *in*: *European Journal of Operational Research* 185.*3* (2008), pp. 1011–1025.

[178] Erwin Hans et al., « Robust Surgery Loading », *in*: *European Journal of Operational Research* 185.*3* (Mar. 2008), pp. 1038–1050.

[179] Brecht Cardoen, Erik Demeulemeester, and Jeroen Beliën, « Optimizing a Multiple Objective Surgical Case Sequencing Problem », *in*: *International Journal of Production Economics* 119.*2* (June 2009), pp. 354–366.

[180] Atle Riise and Edmund K. Burke, « Local Search for the Surgery Admission Planning Problem », *in*: *Journal of Heuristics* 17.*4* (Aug. 2011), pp. 389–414.

[181] Vahid Roshanaei et al., « Collaborative Operating Room Planning and Scheduling », *in*: *INFORMS Journal on Computing* 29 (Aug. 2017), pp. 558–580.

[182] Walton Hancock et al., « Operating room scheduling data base analysis for scheduling », *in*: *Journal of medical systems* 12 (Jan. 1989), pp. 397–409.

[183] David P Strum, Jerrold H May, and Luis Vargas, « Modeling the Uncertainty of Surgical Procedure Times: Comparison of Log-normal and Normal Models », *in*: *Anesthesiology* 92 (May 2000), pp. 1160–7.

[184] Serhat Gul et al., « BiCriteria Scheduling of Surgical Services for an Outpatient Procedure Center », *in*: *Production and Operations Management* 20 (May 2011), pp. 406–417.

[185] Ahmad Ghasemkhani, S. Ali Torabi, and Mahdi Hamid, « A Hybrid Simulation and Optimisation Approach for Capacity Allocation of Operating Rooms under Uncertainty », *in*: *International Journal of Systems Science: Operations & Logistics* 10.*1* (Dec. 2023), p. 2244423.

[186] Oussama Masmoudi, Xavier Delorme, and Paolo Gianessi, « Job-Shop Scheduling Problem with Energy Consideration », *in*: *International Journal of Production Economics* 216 (Oct. 2019), pp. 12–22.

[187] Amir Ahmadi-Javid, Zahra Jalali, and Kenneth J Klassen, « Outpatient Appointment Systems in Healthcare: A Review of Optimization Studies », *in*: *European Journal of Operational Research* 258.*1* (Apr. 2017), pp. 3–34.

[188] Ludovica Adacher and Christos G. Cassandras, « Lot Size Optimization in Manufacturing Systems: The Surrogate Method », *in*: *International Journal of Production Economics* 155 (Sept. 2014), pp. 418–426.

APPENDIX

A Proof

A.1 Proof of Theorem 1

Proof. The difference between the models $APRA_{AGC_1\&TC}$ and $APRA_{GC_1\&TC}$ is the constraints (2.26)-(2.27) and (2.37)-(2.38). Thus, the other constraints are omitted in the following proof. Note that since the variables of (2.26) and (2.27) are consistent, and the structure of these two constraints is similar, the method used to prove the equivalence between (2.26) and (2.37) can be applied to prove the equivalence between (2.27) and (2.38). Consequently, we will only present the proof of the equivalence of constraints (2.26) and (2.37).

Proof of condition (i): By summing up the inequalities indexed by p in constraint (2.26), we get inequality (A.1).

$$|\mathcal{F}_d|(b_{rd}+u_{rd}) \geq \sum_{p\in\mathcal{F}|d\in\mathcal{D}_p} x_{prd} \quad \forall r\in\mathcal{R}_D^M, d\in\mathcal{D} \tag{A.1}$$

According to the domains of the variables b_{rd}, u_{rd}, x_{prd}, we consider the following two cases for the inequality (A.1):
Case 1: If $b_{rd}+u_{rd}=0$, then $\sum_{p\in\mathcal{F}|d\in\mathcal{D}_p} x_{prd}=0$, which implies that constraint (2.37) is satisfied.
Case 2: If $b_{rd}+u_{rd}>0$, then we can rewrite (A.1) as:

$$\frac{\sum_{p\in\mathcal{F}|d\in\mathcal{D}_p} x_{prd}}{b_{rd}+u_{rd}} \leq |\mathcal{F}_d| \quad \forall r\in\mathcal{R}_D^M, d\in\mathcal{D} \tag{A.2}$$

Since the number of female elective patients assgined to room r on day d is no more than the capacity of room r, and the number of female elective patients on day d, i.e., $\sum_{p\in\mathcal{F}|d\in\mathcal{D}_p} x_{prd} \leq \min\{Q_r, |\mathcal{F}_d|\} = \lambda_{rd}^F, \forall r\in\mathcal{R}_D^M, d\in\mathcal{D}$, it follows that the left-hand side of (A.2) is no more than λ_{rd}^F. As a result, constraint (2.37) is also satisfied in this case. Thus, condition (i) is proved.

Proof of condition (ii): By rewriting the constraint (2.37), we have:

$$b_{rd} + u_{rd} \geq \frac{\sum_{p \in \mathcal{F} | d \in \mathcal{D}_p} x_{prd}}{\lambda^F_{rd}} \geq x_{prd} \quad \forall p \in \mathcal{F}, d \in \mathcal{D}_p, r \in \mathcal{R}^M_D \tag{A.3}$$

Obviously, the constraint (2.26) is satisfied. Thus, we have proved condition (ii). Therefore, we have proved the equivalence of the two models. ■

A.2 Proof of Theorem 2

Proof. Given two APRA models, i.e. $APRA_{GC_1\&TC}$ and $APRA_{GC_1\&ATC}$. In order to prove the equivalence of the two models, we need to prove the following two conditions: (i) if $(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{t})$ is a feasible solution to the $APRA_{GC_1\&TC}$, then, there exists a vector $\boldsymbol{z}$ such that $(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{z})$ is feasible to the $APRA_{GC_1\&ATC}$ with objective value $S_{APRA_{GC_1\&ATC}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{z}) = S_{APRA_{GC_1\&TC}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{t})$. (ii) if $(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{z})$ is a feasible solution to the $APRA_{GC_1\&ATC}$, then, there exists a vector $\boldsymbol{t}$ such that $(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{t})$ is feasible to the $APRA_{GC_1\&TC}$ with objective value $S_{APRA_{GC_1\&TC}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{t}) = S_{APRA_{GC_1\&ATC}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{z})$. The difference between the two models is the objective function and the constraints TC and ATC. Thus, the other constraints are omitted in the following proof. For $\forall p \in \mathcal{P}^E | LOS_p \geq 2, d \in \mathcal{D}$, let $a(p, d)$ be the room index r where $x_{prd} = 1$.

Proof of condition (i): Since complete assignment constraint (2.15) holds, the right-hand side of (2.42) gives $RN_{a(p,d)} - RN_{a(p,d+1)} \in [1 - |R|, |R| - 1]$ and the right-hand side of (2.43) gives $RN_{a(p,d+1)} - RN_{a(p,d)} \in [1 - |R|, |R| - 1]$. If $a(p, d) \neq a(p, d + 1)$, then $z_{pd} = 1$. If $a(p, d) = a(p, d + 1)$, then both $z_{pd} = 0$ and $z_{pd} = 1$ can satisfy (2.42) and (2.43). Hence, $\boldsymbol{z}$ can be determined. $S_{APRA_{GC_0\&ATC}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{z}) = S_{APRA_{GC_1\&TC}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{t})$ is proved by observing that each patient can only change room on two consecutive days once. Thus, $\sum_{r \in \mathcal{R}} t_{prd} = z_{pd}$, $\forall p \in \mathcal{P}^E | LOS_p \geq 2, d \in \mathcal{D}_p \setminus \{DD_p - 1\}$. Therefore, the objectives of the two models are the same. This proves the condition (i).

Proof of condition (ii): We discuss the proof of condition (ii) by considering two cases. Case 1: $a(p, d) = a(p, d + 1)$. In this case, based on the domain of the variable x_{prd}, the right-hand side of (2.20) gives 0. Therefore, both $t_{prd} = 0$ and $t_{prd} = 1$ can satisfy the constraint (2.20). Case 2: $a(p, d) \neq a(p, d + 1)$. In this case, $t_{p,a(p,d),d} = 1$, and for $\forall r \in \mathcal{R} \backslash \{a(p, d)\}$, both $t_{prd} = 0$ and $t_{prd} = 1$ can satisfy the constraint (2.20). Hence, $\boldsymbol{t}$ can be determined. Similar to the proof of condition (i), the two models have the same objectives. Thus, we have proved condition (ii).

Therefore, we have proved the equivalence of the two models. ■

A.3 Proof of Theorem 3

Proof. We use proof by contradiction, assuming that no BII exists when the given two conditions are satisfied, and show that this assumption leads to a contradiction, thus proving the theorem. Suppose that surgery i starts at time z_i, surgery i' ends at time $f_{i'}$, which is the closest completion time to the start time of surgery i, i.e., $\text{argmin}_{i' \in \mathcal{I} | f_{i'} > z_i} f_{i'} - z_i$. Moreover, all ORs are fully occupied at the start time z_i of surgery i. If no BII exists, this means one of the following three conditions must be satisfied: (1) the OR is not fully occupied at the time $z_{i'}$; (2) the OR is fully occupied at the time z_i, but $f_{i'} \leq z_{i'}$; (3) the OR is fully occupied at the time z_i, but there exists a surgery i'' which completion time $f_{i''}$ is closer to the time z_i than $f_{i'}$, i.e., $z_i < f_{i''} < f_{i'}$. However, all these three conditions contradict the assumption. Since the assumption that no BII exists leads to a contradiction, we conclude that a BII must exist when the given conditions are satisfied. Therefore, we have proved the theorem 3. ■

A.4 Proof of Theorem 4

Proof. Let's assume there are two surgeries, i and i', with postponing costs C_i^P and $C_{i'}^P$, and performing costs C_{ib} and $C_{i'b}$, respectively. Without loss of generality, we can assume $C_i^P > C_{i'}^P$. Consider the two possible sequences: $i - i'$ and $i' - i$. The total costs are given by $C_{ib} + C_{i'}^P$ and $C_{i'b} + C_i^P$, respectively. To show that the first sequence is optimal, we must prove that $C_{ib} + C_{i'}^P \leq C_i^P + C_{i'b}$. Rearranging the terms, we get $C_{ib} - C_i^P \leq C_{i'b} - C_{i'}^P$. Since $C_i^P > C_{i'}^P$, we can rewrite the inequality as $C_i^P - C_{ib} \geq C_{i'}^P - C_{i'b}$.

As the postponing cost is always greater than the performing cost for each surgery, this inequality holds true. Therefore, scheduling i before i' minimizes the total cost, proving that sorting the surgeries in descending order based on the postponing cost is indeed optimal. Moreover, according to the above proof, one can easily prove that if two surgeries have the same postponing cost, the one with the lower performing cost should be scheduled first. ■

A.5 Proof of Theorem 5

Proof. Let's assume that after the K-th iteration, constraints (5.40)-(5.41) are added to the $ESAP_{REC}$ concerning the sets $\dot{\mathcal{H}}_d^{(k)}$ and $\mathcal{H}_d^{(k)}$, $k = 1, 2, ..., K$. During the n-th iteration ($n > K$) of solving $ESAP_{REC}$, let's consider the index set of the variables

$x_{ib} = 1, (i,b) \in \widetilde{\mathcal{H}}_d$ and $\widetilde{\mathcal{H}}_d \neq \emptyset$. Consequently, all possible relations among $\widetilde{\mathcal{H}}_d$, $\dot{\mathcal{H}}_d^{(k)}$ and $\mathcal{H}_d^{(k)}$ are as depicted in Figure A.1.

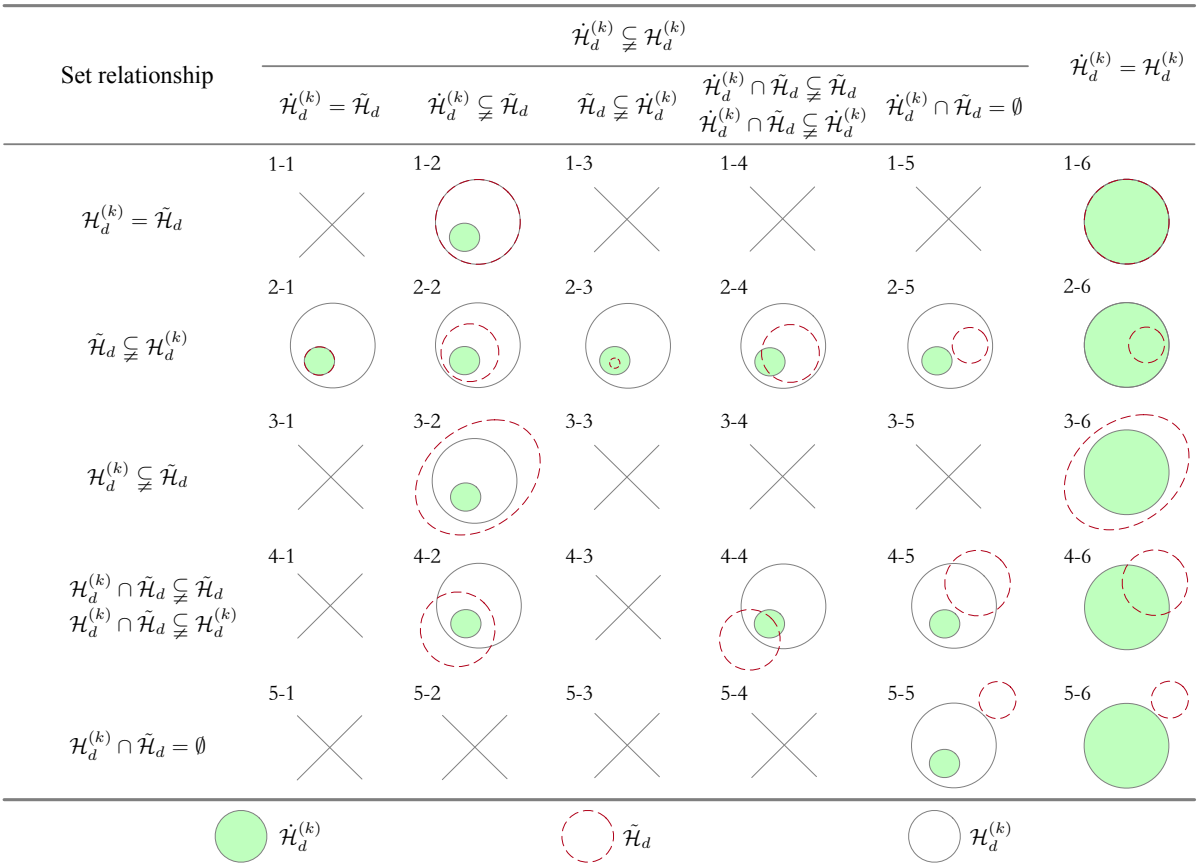


Figure A.1 – All possible relationships among $\dot{\mathcal{H}}_d^{(k)}$, $\widetilde{\mathcal{H}}_d$ and $\mathcal{H}_d^{(k)}$

The figure illustrates that under the condition $\dot{\mathcal{H}}_d^{(k)} \subsetneqq \mathcal{H}_d^{(k)}$, columns 2-6 respectively illustrate conditions where $\widetilde{\mathcal{H}}_d$ equals, contains, is encompassed by, intersects with, or is disjoint from $\dot{\mathcal{H}}_d^{(k)}$. Column 7 represents the condition where $\dot{\mathcal{H}}_d^{(k)} = \mathcal{H}_d^{(k)}$. Each row respectively illustrates the conditions where $\widetilde{\mathcal{H}}_d$ is equal to, includes, is included in, intersects with, or is disjoint from $\mathcal{H}_d^{(k)}$. This results in a total of 30 possible relationships.

Nonetheless, given that $\dot{\mathcal{H}}_d^{(k)} \subseteq \mathcal{H}_d^{(k)}$, the relationships 1-1, 1-3, 1-4, 1-5, 3-1, 3-3, 3-4, 3-5, 4-1, 4-3, 5-1, 5-2, 5-3, and 5-4 are not feasible, accounting for 14 non-existing scenarios. Owing to the proof provided by $ESSP_{B\&B}$, which establishes that $\theta_d^1\left(x\left(\dot{\mathcal{H}}_d^{(k)}\right)\right) \leq \theta_d^1\left(x\left(\widetilde{\mathcal{H}}_d\right)\right)$ while $\widetilde{\mathcal{H}}_d \subseteq \mathcal{H}_d^{(k)}$, the task of proving the validation of the constraints (5.40)-(5.41) becomes equivalent to showing that these constraints are capable of excluding relationships 1-2, 2-2, 2-3, 2-4, 2-5, 2-6, but unable to exclude the relationships 1-6, 2-1, 3-2, 3-6, 4-2, 4-4, 4-5, 4-6, 5-5, and 5-6.

For each of the potential relationships outlined in Figure A.1, let the variables in the constraints (5.40)-(5.41) be $x_{ib} = 1, x_{i'b'} = 0$ where $(i,b) \in \widetilde{\mathcal{H}}_d$ and $(i',b') \notin \widetilde{\mathcal{H}}_d$. By calculating the values of these two constraints under these relationships, one can determine whether a particular relationship is excluded. Specifically, if the calculated value does not satisfy the inequality relationship, then the relationship is excluded. Conversely, if the calculated value does satisfy the inequality relationship, the relationship cannot be excluded.

To facilitate discussion, let us divide constraint (5.40) into three parts, as shown in constraints (A.4).

$$\left| \overbrace{\sum_{(i,b)\in\mathcal{H}_d^{(k)}\setminus\dot{\mathcal{H}}_d^{(k)}} x_{ib}}^{A} - \overbrace{\sum_{(i,b)\in\dot{\mathcal{H}}_d^{(k)}} \left(\left|\mathcal{H}_d^{(k)} \setminus \dot{\mathcal{H}}_d^{(k)}\right| + 1\right)(1 - x_{ib})}^{B} \right| \leq \overbrace{\sum_{(i,b)\in\mathcal{H}_d\setminus\mathcal{H}_d^{(k)}} |\mathcal{H}_d|^2 x_{ib}}^{C} \tag{A.4}$$
$$\forall d \in \mathcal{D}; k = 1, 2, ..., K$$

Table A.1 presents the values of the two constraints under each relationship. According to this Table, relationships 1-2 and 2-2 do not satisfy $A - B \leq C$, and relationships 2-3, 2-4, 2-5, and 2-6 do not satisfy $A - B \geq -C$. That is, the constraints (5.40) can exclude 2 relationships, and the constraints (5.41) can exclude 4 relationships. Therefore, these two constraints can exclude these 6 relationships. The remaining 10 relationships all satisfy these two constraints, implying that these two constraints will not exclude these relationships.

In summary, LBA feedback constraints (5.40) and (5.41) are valid. ■

A.6 Proof of validity of the constraint (4.17)

Proof. Given day d and surgery i, the validity of constraint (4.17) can be divided into two cases for discussion, including $\sum_{b\in\mathcal{B}_d} x_{ib} = 0$ and $\sum_{b\in\mathcal{B}_d} x_{ib} = 1$.

Case 1: $\sum_{b\in\mathcal{B}_d} x_{ib} = 0$, which means surgery i is not operated on day d.

Since constraints (4.15) and (4.16) hold, we have $g_{ii'd} = \phi_{ii'd} = 0$. In this case, if $\sum_{b\in\mathcal{B}_d|b\neq b'} x_{i'b} = 1$, we have $g_{ii'd} + \phi_{ii'd} - \sum_{b\in\mathcal{B}_d|b\neq b'} x_{i'b} = -1$. Thus, $\max \sum_{b'\in\mathcal{B}_d|b\neq b'}(g_{ii'd} + \phi_{ii'd} - x_{i'b}) - |\mathcal{B}_d| + 2 \leq 1 - |\mathcal{B}_d| \leq 0$, which means there is no BII when surgery i is not operated on day d. Therefore, in this case, the expression of the formula is consistent with the meaning of the constraint.

Table A.1 – Values of the constraints (5.40)-(5.41) under different relationships

Relationship	A	B	C	$A-B\leq C$	$A-B\geq -C$
1-2	α	0	0	×	✓
1-6	0	0	0	✓	✓
2-1	0	0	0	✓	✓
2-2	$[1,\alpha-1]$	0	0	×	✓
2-3	0	$[\alpha+1,(\alpha+1)(\lvert\dot{\mathcal{H}}_d^{(k)}\rvert-1)]$	0	✓	×
2-4	$[1,\alpha]$	$[\alpha+1,(\alpha+1)(\lvert\dot{\mathcal{H}}_d^{(k)}\rvert-1)]$	0	✓	×
2-5	$[1,\alpha]$	$(\alpha+1)\lvert\dot{\mathcal{H}}_d^{(k)}\rvert$	0	✓	×
2-6	0	$[1,(\lvert\dot{\mathcal{H}}_d^{(k)}\rvert-1)]$	0	✓	×
3-2	α	0	$[\lvert\mathcal{H}_d\rvert^2,\beta\lvert\mathcal{H}_d\rvert^2]$	✓	✓
3-6	0	0	$[\lvert\mathcal{H}_d\rvert^2,\beta\lvert\mathcal{H}_d\rvert^2]$	✓	✓
4-2	$[0,\alpha-1]$	0	$[\lvert\mathcal{H}_d\rvert^2,\beta\lvert\mathcal{H}_d\rvert^2]$	✓	✓
4-4	$[0,\alpha]$	$[\alpha+1,(\alpha+1)(\lvert\dot{\mathcal{H}}_d^{(k)}\rvert-1)]$	$[\lvert\mathcal{H}_d\rvert^2,\beta\lvert\mathcal{H}_d\rvert^2]$	✓	✓
4-5	$[1,\alpha]$	$(\alpha+1)\lvert\dot{\mathcal{H}}_d^{(k)}\rvert$	$[\lvert\mathcal{H}_d\rvert^2,\beta\lvert\mathcal{H}_d\rvert^2]$	✓	✓
4-6	0	$[1,(\lvert\dot{\mathcal{H}}_d^{(k)}\rvert-1)]$	$[\lvert\mathcal{H}_d\rvert^2,\beta\lvert\mathcal{H}_d\rvert^2]$	✓	✓
5-5	0	$(\alpha+1)\lvert\dot{\mathcal{H}}_d^{(k)}\rvert$	$[\lvert\mathcal{H}_d\rvert^2,\beta\lvert\mathcal{H}_d\rvert^2]$	✓	✓
5-6	0	$\lvert\dot{\mathcal{H}}_d^{(k)}\rvert$	$[\lvert\mathcal{H}_d\rvert^2,\beta\lvert\mathcal{H}_d\rvert^2]$	✓	✓

$^*\alpha=\left|\mathcal{H}_d^{(k)}\setminus\dot{\mathcal{H}}_d^{(k)}\right|,\beta=\left|\mathcal{H}_d\setminus\mathcal{H}_d^{(k)}\right|$

Case 2: $\sum_{b\in\mathcal{B}_d}x_{ib}=1$, which means surgery i is operated on day d.

Since constraints (4.15) and (4.16) hold, if $\sum_{b\in\mathcal{B}_d|b\neq b'}x_{i'b}=0$, we have $g_{ii'd}=\phi_{ii'd}=0$, $g_{ii'd}+\phi_{ii'd}-\sum_{b\in\mathcal{B}_d|b\neq b'}x_{i'b}=0$. In contrast, if $\sum_{b\in\mathcal{B}_d|b\neq b'}x_{i'b}=1$, there are two possible situations: $g_{ii'd}+\phi_{ii'd}-\sum_{b\in\mathcal{B}_d|b\neq b'}x_{i'b}=0$ or $g_{ii'd}+\phi_{ii'd}-\sum_{b\in\mathcal{B}_d|b\neq b'}x_{i'b}=1$. The former indicates that surgery i' is not operated when surgery i starts, while the latter indicates that surgery i' is being operated when surgery i starts. Furthermore, since there are at most $|B_d|-1$ surgeries being operated when surgery i starts, we have $\max\sum_{b'\in\mathcal{B}_d|b\neq b'}(g_{ii'd}+\phi_{ii'd}-x_{i'b})\leq|\mathcal{B}_d|-1$. Therefore, $\max\sum_{b'\in\mathcal{B}_d|b\neq b'}(g_{ii'd}+\phi_{ii'd}-x_{i'b})-|\mathcal{B}_d|+2\leq 1$, which means there is at most one BII when surgery i is operated on day d, and the BII exists if and only if ORs are fully occupied when surgery i' starts. Thus, in this case, the expression of the formula is consistent with the meaning of the constraint.

In summary, the constraint (4.17) is valid for both cases. ■

B Discussion on constraints system of computing the BII in the literature

In order to better explain the problem of computing the length of the BII in the literature, we extract the related constraints from the existing literature and analyze them in detail. Specifically, Schulz and Fliedner [59] proposed the following constraints to compute the BII:

$$s_i + p_i - (s_j + p_j) \leq b_j + \left(x_{ij} \cdot (|I| + 1) + \left(\sum_{k \in I: k \neq i,j} x_{ki} - x_{kj} \right) - 1 \right) \cdot M \quad \forall i, j \in I : i \neq j \tag{B.5}$$

$$s_i + p_i - (s_j + p_j) \geq b_j - \left(x_{ij} \cdot (|I| + 1) + \left(\sum_{k \in I: k \neq i,j} x_{ki} - x_{kj} \right) - 1 \right) \cdot M \quad \forall i, j \in I : i \neq j \tag{B.6}$$

where I is the set of surgeries, i, j are indices of surgeries in I. s_i and p_i are the start time and duration of surgery i, respectively. b_j is the duration of the BII that follows surgery j. x_{ij} is a binary variable that equals 1 if surgery i is starts before j and 0 otherwise. M is a large positive number. Constraints (B.5) and (B.6) fix the BII, i.e., the time between the ending time of surgery j and the ending time of surgery i which starts next b_j, for all jobs which have a successor. As we showed in Figure 4.4(b), the BII length in the surgery schedule with buffer can only calculated by the difference of the completion time and start time of two overlapping surgeries, which satisfy the conditions as provided in Theorem 3. Thus, the constraints (B.5) and (B.6) fail to compute the BII in the surgical case scheduling with buffer.

Moreover, Latorre-Núñez et al. [57] calculated the BII by the difference between the completion time and the start time of two overlapping surgeries, which are shown as follows:

$$C_i^1 + L_i \leq C_j^1 + L_j + G \cdot (1 - z_{ij}^2) \quad \forall i, j \in J | i \neq j \tag{B.7}$$

$$z_{ij}^2 + z_{ji}^2 = 1 \quad i, j \in J | i \neq j \tag{B.8}$$

$$E_{max} \geq (C_j^1 + L_j) - (C_i^1 - P_i^1 - S_i) - z_{ji}^1 \cdot G \quad \forall i, j \in J | i \neq j : P_j^1 + S_j + L_j > E_{max} \tag{B.9}$$

$$z_{ij}^2 + z_{ij}^1 \leq z_{ij}^3 + 1 \quad \forall i, j \in J | i \neq j \tag{B.10}$$

$$\sum_{j \in J | j \neq i} z_{ij}^3 \leq m^1 - 2 \quad i \in J : P_i^1 + S_i + L_i > E_{max} \tag{B.11}$$

where J is the set of surgeries, i, j are the indices of surgeries in J. Parameters C_i^1, L_i, P_i^1, S_i are the completion time, the clean time, the duration, and the setup time of surgery i, respectively. m^1 is the number of ORs. G is a large positive number. Parameter E_{max} is the maximum allowable length of BII. z_{ij}^1, z_{ij}^2, z_{ij}^3 are binary variables. Specifically, z_{ij}^1 equals 1 if the gap between the start of surgery i and the end of surgery j is greater than E_{max}, and 0 otherwise. z_{ij}^2 equals 1 if the completion of cleaning of surgery i precedes the completion of the cleaning of surgery j, and 0 otherwise. z_{ij}^3 equals 1 if the parallel performing time of surgery i and j is greater than E_{max}, 0 otherwise. Constraint (B.7) determines the precedence between the completion of the cleaning activities after the surgeries. For each combination of surgeries, constraint (B.8) states that there is only one predecessor. Constraint (B.9) requires that the difference between the end of surgery j and the start of surgery i to be less than or equal to E_{max}. Constraint (B.10) determines whether the time of the surgeries scheduled in parallel is greater than E_{max}. The author proposed constraint (B.11) to ensure that E_{max} is met. However, this constraint may not be correct. For example, if there are 2 ORs and 10 surgeries, the right side of the constraint (B.11) is zero, which means that the time to perform surgery i and any other surgery j is less than E_{max}. This is incorrect if E_{max} is a very small number, such as 1 minute. Therefore, Latorre-Núñez et al. [57] did not carefully consider overlapping cases, which may lead to incorrect results.

C Application to the original patient admission scheduling problem

As mentioned in Section 1.2, various static PAS problems have been studied in the literature. The differences are the treatment of SC1-SC4 and SC9 constraints. Our proposed method can solve these static variants by decreasing the number of soft constraints and adjusting the domain of the patient-room assignment variables according to the specific problem definition. Different from the standard PAS problem we solved, in the original PAS problem proposed by [25], the former four constraints are hard constraints, which are not allowed to be violated. In order to solve the original PAS problem, we use our pro-

posed two-stage optimization approach and modify the APRA$^{\mathrm{WT}}$ and APRA models by limiting the set of rooms that can be assigned to each patient. Specifically, we use $\mathcal{R}_p \in \mathcal{R}$, which is defined as the set of rooms that can be assigned to patient p without violating the constraints SC1-SC4. The modified $APRA^{WT}$ and $APRA$ models are formulated as follows:

$$\text{Modified } \boldsymbol{APRA^{WT}}\text{: Min } S = \sum_{p\in\mathcal{P}^E}\sum_{r\in\mathcal{R}_p} C'_{pr}x_{pr} \tag{C.12}$$

s.t. Constraints (2.30), (2.34), where $\mathcal{R}_p$ is used instead of $\mathcal{R}$

Constraints (2.23)

$$\sum_{p\in\mathcal{P}|d\in\mathcal{D}_p,r\in\mathcal{R}_p} x_{pr} \leq Q_r, \quad \forall d\in\mathcal{D}, r\in\mathcal{R} \tag{C.13}$$

$$\lambda^F_{rd} f_{rd} \geq \sum_{p\in\mathcal{F}|d\in\mathcal{D}_p,r\in\mathcal{R}_p} x_{pr} \quad \forall d\in\mathcal{D}, r\in\mathcal{R}^M_D \tag{C.14}$$

$$\lambda^M_{rd}(1-f_{rd}) \geq \sum_{p\in\mathcal{M}|d\in\mathcal{D}_p,r\in\mathcal{R}_p} x_{pr} \quad \forall d\in\mathcal{D}, r\in\mathcal{R}^M_D \tag{C.15}$$

$$\text{Modified } \boldsymbol{APRA}\text{: Min } S = \sum_{p\in\mathcal{P}^E,r\in\mathcal{R}_p,d\in\mathcal{D}_p} C_{prd}\cdot x_{prd} + \sum_{p\in\mathcal{P}^E|LOS_p\geq 2,r\in\mathcal{R}_p,d\in\mathcal{D}_p\setminus\{DD_p-1\}} W_{Tr}\cdot t_{prd} \tag{C.16}$$

s.t. Constraints (2.15), (2.20), (2.21), (2.25), where $\mathcal{R}_p$ is used instead of $\mathcal{R}$

Constraints (C.13), (C.14), (C.15), where x_{prd} is used instead of x_{pr}

Constraints (2.23)

The computational results are summarized in Table C.2. Since SC1-SC4 are considered to be hard constraints, the corresponding penalties (Gen., Age, Sng. and Ned. Pref.) are equal to zero and are not reported. Note that most studies treat SC1-SC4 as soft constraints, and therefore, the corresponding problems are relaxations of the original PAS problem. Consequently, the best lower bounds in those studies can be used as the best known lower bounds BLB of the original PAS problem. The best known solutions BKS are similarly derived from the results in the literature without incurring penalties of SC1-SC4. The symbol "-" is used to indicate that the instance is infeasible for the original PAS

problem or the result is not available in the literature.

Table C.2 Results on the benchmark instances for the original PAS problem (new best solutions and new best lower bounds in **bold**, proven optimal solutions in star *).

Instance	Literature results		Two-stage optimization approach							Breakdown of the *Obj* components				
	BKS (Time to end**)	*BLB* (Time to end**)	*Obj*	Time to best	Time to end	*LB*	*GAP* (%)	Node of APRA^{WT}	Node of APRA	Room pref.	Dept.	Spec.	Pref. prop.	Trs.
1	651.20 (21,226[8])	651.20 (21,226[8])	651.20*	272	5,370	651.20	0.00	5,872	6,732	651.2	0.0	0.0	0.0	0.0
2	1,128.00 (44,258[8], 2,577[9])	1,115.80 (44,258[8])	**1,125.60***	7,638	24,372	**1,125.60**	0.00	10,447	8,356	1,113.6	0.0	12.0	0.0	0.0
3	761.60 (44,258[8], 2,577[9])	758.60 (44,258[8])	761.60*	2,021	12,864	**761.60**	0.00	46,752	62,533	753.6	0.0	8.0	0.0	0.0
4	1,151.60 (44,258[8])	1,143.20 (44,258[8])	**1,151.00**	35,818	86,400	**1,150.00**	0.09	144,679	553,384	1040.0	0.0	75.0	36.0	0.0
5	624.00 (4,227[8], 62[9])	624.00 (4,227[8])	624.00*	199	752	624.00	0.00	670	5,603	624.0	0.0	0.0	0.0	0.0
6	792.60 (10,082[8], 2,577[9])	792.60 (10,082[8])	792.60*	462	1,019	792.60	0.00	45	1	789.6	0.0	3.0	0.0	0.0
7	1,176.40 (6,209[8], 2,577[9])	1,176.40 (4,209[8])	1,176.40*	36	486	1,176.40	0.00	321	5,206	730.4	20.0	158.0	268.0	0.0
8	4,063.00 (44,258[8])	4,024.41 (44,258[8])	**4,058.60**	9,655	86,400	**4,039.60**	0.47	9,137	3,020	1,433.6	214.0	871.0	1,518.0	22.0
9	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10	7,804.60 (2,577[9])	7,687.33 (44,258[8])	**7,793.80**	43,200	86,400	**7,719.60**	0.96	1,298	1	2,948.8	4.0	481.0	4,360.0	0.0
11	11,536.20 (654[9])	10,987.72 (44,258[8])	11,836.60	43,200	86,400	10727.05	7.73	268	0	4,361.6	16.0	963.0	6,496.0	0.0
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	8,842.80 (44,258[8])	**9,093.60**	67,866	86,400	**8,912.40**	2.03	66,457	441	2,061.6	627.0	1,695.0	4,556.0	154.0

** Total computation time reported by the corresponding reference, adjusted following the procedure from [107]

From Table C.2, we observe that our approach computed 5 out of 13 new best solutions on the tested benchmark instances (2, 4, 8, 10, 13, solutions obtained for instances 10, 13 are better than our new found solutions in Table 2.8). Our approach proved the optimality of 6 out of 13 solutions (1, 2, 3, 5, 6, 7). Moreover, our approach improved the best lower bound for 6 out of the instances (2, 3, 4, 8, 10, 13, lower bounds obtained for instances 8, 10, 13 are better than our new found lower bounds in Table 2.8). Note that instances 9 and 12 are infeasible in the original PAS problem. The reason is that the number of elective patients exceeds the capacity of the rooms allowed for them.

List of Figures

List of Tables

List of Abbreviation

The following abbreviations are used in this thesis.

ABC	Artificial bee colony
AGC	Aggregated gender policy constraint
ANLGD	Adaptive non-linear great deluge
APRA	Advanced patient-room assignment
ATC	Aggregated patient transfer constraint
B&B	Branch-and-Bound
BBO	Biogeography based optimization
BBO-GBS	BBO algorithm with guided bed selection mechanism
BIM	Break-in-Moment
BII	Break-in-Interval
B&C	Branch-and-cut
CA	Constraint aggregation
CAP	Concentrated allocation policy
CARD	General cardiology
CG	Column generation
CM	Complete model
Con.	Constraint
DCA	Dynamic constraint aggregation
DES	Discrete-event simulation
DFP	Discrete flower pollination
DPAS	Dynamic patient admission scheduling
DRO	Distributionally robust optimization
DSR	Demand-to-supply ratio
EVP	Expected value problem
$ESAP_{REC}$	Elective surgery assignment subproblem considering emergency demand
$ESSP_{B\&B}$	elective surgery sequencing subproblems with BIMs & buffers
F&O	Fix-and-Optimize

F&R	Fix-and-Relax
GASTRO	Gastroenterology
GC	Gender policy constraint
GYN	Gynecology
HC	Hard constraint
H-H	Hyper-heuristic
HPSO	Hybrid particle swarm optimization
HS	Harmony search
H-TS	Hybrid Tabu search
IP	Integer programming
LAHC	Late acceptance hill climbing
LB	Lower bound
LBA	Local best assignment
LOS	Length of stay
LP	Linear program
MAD	Maximum allowable deviation
MBBO-GBS	Modified BBO algorithm with guided bed selection mechanism
MDP	Markov decision process
MED	Medicine
MIP	Mixed integer programming
M-M	Med mid
M-S	Med short
NRP	Nurse rostering problem
Obj	Objective value
OR	Operating room
ORTH	Orthopedics
PAS	Patient admission scheduling
PASU	Patient admission scheduling problem under uncertainty
PBA	Patient-bed assignment
PRA	Patient-room assignment
PS	Proactive scheduling
RAP	Random allocation policy
RS	Reactive scheduling
RSV	Relaxed $SPAS_{SV}$ model

SA	Simulated annealing
SAA	Sample average approximation
SAA-SV	Hybrid sample average approximation and state-variable modeling
SB	scenario-based
SC	Soft constraint
SCA	Static constraint aggregation
SCS	Surgery case scheduling
S-L	Small long
SM	Simplified model
S-M	Small mid
SP	Stochastic programming
SPAS	Stochastic patient admission scheduling
S-S	Small short
SSFU	Surgical case scheduling in flexible operating rooms under uncertainty
SV	State-variable
TC	Patient transfer constraint
TPSO	Three-phase simulation-optimization
UAP	Uniform allocation policy
URO	Urology
Var.	Variable
VSS	Value of stochastic solution
WS	Warm start

List of Publications

Published/accepted papers

— **Haichao Liu**, Yang Wang, Jin-Kao Hao. Solving the patient admission scheduling problem using constraint aggregation. *European Journal of Operational Research* 316 (2024): 85-99.

— Yang Wang, **Haichao Liu**, Bo Peng, Haibo Wang, Abraham P. Punnen. A three-phase matheuristic algorithm for the multi-days task assignment problem. *Computers & Operations Research* 159 (2023): 106313.

— **Haichao Liu**, Yang Wang, Hongpu Wang. Advanced strategies for logic-based Benders decomposition. *Journal of Systems Engineering*, Accepted, September 2024. (This is a Chinese journal.)

Submitted papers

— **Haichao Liu**, Jin-Kao Hao, Yang Wang, Abraham P. Punnen. Stochastic patient admission scheduling with an exponential number of scenarios. *European Journal of Operational Research*, Under review, Augest, 2024.

— **Haichao Liu**, Yang Wang, Hongpu Wang, Jianguang Feng, Ada Che, Jin-Kao Hao, Abraham P. Punnen. Modeling and simulation-optimization approaches for surgery scheduling in flexible operating rooms under uncertainty. *Production and Operations Management*, Submitted, October 2024.

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

université angers

Titre : Solutions pilotées par modèle pour la planification des patients dans la gestion des admissions et des chirurgies

Mot clés : gestion de la santé, planification de l'admission des patients, planification des opérations chirurgicales, programmation stochastique, optimisation par simulation

Résumé : La planification efficace des patients est essentielle pour améliorer l'efficacité des ressources médicales et la satisfaction des patients dans le système de santé. Cette thèse présente des modèles mathématiques et des approches de solution pour le problème de planification des admissions de patients statiques/stochastiques et le problème de planification des cas chirurgicaux dans des salles d'opération flexibles sous incertitude. En raison de la complexité de ces problèmes, qui impliquent de multiples contraintes et incertitudes, des méthodes de résolution innovantes telles que l'optimisation en deux phases, l'approximation hybride de la moyenne des échantillons avec variable d'état (SAA-SV), la planification intégrée proactive et réactive, et l'optimisation de simulation en trois phases sont proposées. De plus, des approches de modélisation avancées comme l'agrégation de contraintes, la modélisation aléatoire basée sur des scénarios, et la modélisation à variable d'état sont développées pour créer des modèles de taille réduite afin d'améliorer l'efficacité de leur résolution. Des études computationnelles réalisées sur un ensemble d'instances de référence démontrent l'efficacité des méthodes proposées en comparaison avec les méthodes de l'état de l'art. Des expériences supplémentaires sont menées pour évaluer le rôle des composants clés des méthodes proposées.

Title: Model-driven solution approaches for patient scheduling in admission and surgery management

Keywords: healthcare management, patient admission scheduling, surgery scheduling, stochastic programming, simulation optimization

Abstract: Efficient patient scheduling plays a crucial role in improving the efficiency of medical resources and patient satisfaction in the healthcare system. This thesis presents mathematical modeling and solution approaches for the static/stochastic patient admission scheduling problem and the surgical case scheduling problem in flexible operating rooms under uncertainty. Given the complexity of these problems with multi-constraint and uncertainty, innovative solution approaches — two-stage optimization, hybrid sample average approximation and state-variable (SAA-SV), integrated proactive and reactive scheduling, and three-phase simulation-optimization — are proposed to solve the problems efficiently. Especially, advanced modeling approaches — constraint aggregation, scenario-based modeling, and state-variable modeling — are used to build reduced models to improve the solution efficiency. Computational studies performed on a set of benchmark instances show the effectiveness of the proposed methods in comparison with the state-of-the-art methods. Additional experiments are conducted to evaluate the roles of the key ingredients of the proposed methods.