Øyvind Asplin
Erlend Johan Corneliussen
Katarina Van de Pontseele

Flexible Resource Management for Surgical Scheduling

Applying Heuristic Solution Methods to a Stochastic Two-Stage Problem

Master's thesis in Industrial Economics and Technology Management
Supervisor: Anders Gullhav
Co-supervisor: Thomas Bovim

June 2022

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Economics and Management
Dept. of Industrial Economics and Technology Management

NTNU
Norwegian University of Science and Technology

Øyvind Asplin
Erlend Johan Corneliussen
Katarina Van de Pontseele

Flexible Resource Management for Surgical Scheduling

Applying Heuristic Solution Methods to a Stochastic Two-Stage Problem

Master's thesis in Industrial Economics and Technology Management
Supervisor: Anders Gullhav
Co-supervisor: Thomas Bovim
June 2022

Norwegian University of Science and Technology
Faculty of Economics and Management
Dept. of Industrial Economics and Technology Management

NTNU
Norwegian University of
Science and Technology

Preface

This study is our master's thesis in TIØ4905 Managerial Economics and Operations Research, Master's Thesis. It is written in the spring semester of our last year in the program Industrial Economics and Technology Management, under the Faculty of Economics and the Department of Industrial Economics and Technology Management at the Norwegian University of Science and Technology (NTNU). The thesis builds upon the specialization project we conducted in the previous semester (Asplin et al., 2021). We tackle the same problem, but develop alternative solution methods. Parts of the thesis are therefore based on Asplin et al. (2021).

The purpose of the thesis is to research the value of stochastic programming when planning for uncertainty in demand at a surgical clinic. It is furthermore to investigate the value of incorporating flexibility in surgery schedules. In order to investigate this properly, and due to the complexity of the stochastic optimization problem, we develop heuristic solution methods. Lastly, we aim to make a contribution to the literature and to provide a prototype for a planning tool for surgical clinics.

We would like to acknowledge the contributions in the work of this thesis of the people at both NTNU and the Clinic of Surgery at St. Olavs Hospital. Thanks to our supervisors from the Department of Industrial Economics and Technology Management, Anders Gullhav and Thomas Bovim, who have been a great support when tackling the fields of both operations research and health care. Thanks to our contacts in the management and surgical staff at the Clinic of Surgery at St. Olavs Hospital who helped us understand the dynamics and objectives of the surgical clinic. Thanks to Pia Isaksen and Martine Svagård for the work they did in their master's thesis based on the same surgical clinic, allowing us to hit the ground running in both our specialization project and this thesis (Isaksen and Svagård, 2021). Lastly, we would like to thank our fellow students of TIØ4905. It's been a pleasure working in an office space with such an open learning environment, where both time and knowledge have been shared across teams.

Øyvind Asplin, Erlend Johan Corneliussen and Katarina Van de Pontseele

Trondheim, June 7th 2022

Sammendrag

Befolkningen i Norge og andre i-land vokser og blir eldre. Dette gjør det utfordrende å møte pasientetterspørselen på sykehus. I den nåværende fire-års-planen til Helse- og omsorgsdepartementet er målet å møte denne etterspørselen på bærekraftig vis (Ministry of Health and Care Services, 2020b). Effektiv kirurgisk planlegging er et viktig tiltak for å nå dette målet, da over 70% av alle sykehispasienter har behov for kirurgisk behandling (Bovim et al., 2020). Til tross for at kirurgisk planlegging både er innviklet og tidkrevende blir det stort sett gjort manuelt. Operasjonsanalyse kan være med på å automatisere denne oppgaven og dermed bidra til å nå målet om et mer bærekraftig helsevesen.

I denne masteroppgaven tar vi for oss det taktiske problemet med å lage en modifisert kirurgisk masterplan (modifisert MSS) for den Kirurgisk klinikk på St. Olavs Hospital. En modifisert MSS er en gjentakende syklisk timeplan der kirurgiske spesialiteter blir tildelt operasjonsrom på ulike dager. Et gitt operasjonsrom på en gitt dag i en MSS kalles en tidsluke. Det som skiller en modifisert MSS fra en vanlig MSS er at noen av tidslukene i hver syklus ikke tildeles en spesialitet før tett opptil hver planleggingsperiode. Disse kan da benyttes av de spesialitetene med størst behov i hver syklus. Planleggingsverktøyet har som mål å minimere mengden umøtt etterpørsel ved å hensynta usikkerheten i etterspørsel for ulike kirurgiske inngrep. Den overordnede hensikten til masteroppgaven er å vurdere verdien i å ta hensyn til usikkerheten i pasientetterspørsel og verdien av å tillate fleksible tidsluker i kirurgisk planlegging. Det er mange faktorer som bør tas i betraktning i kirurgisk planlegging. Vi har valgt å hensynta begrensninger i antall operasjonssrom, ansatte og kapasiteten til sengepostene for postoperativ behandling.

Vi implementerer planleggingsverktøyet som en stokastisk to-stegs-modell. Informasjon om distribusjonen av pasientetterpørsel er kjent i førstesteget, men den faktiske etterspørselen for ulike kirurgiske inngrep er avsløres i andresteget. De kirurgiske inngrepene blir aggregert til operasjonsgrupper gjennom en klynge-algoritme utviklet i Isaksen and Svagård (2021). Hver operasjonsgruppe tilhører én kirurgisk spesialitet og opereres derfor kun i tidsluker tildelt denne spesialiteten i den mofifiserte MSSen. Usikkerheten i etterpørsel blir modellert ved å simulere etterspørselsscenarier for operasjonsgruppene, som så brukes som input til modellen. Klyngingen av kirurgiske inngrep og simuleringen av etterpørsel er basert på data fra 2019 fra Kirurgiske klinikk ved St. Olavs Hospital. Den stokastiske to-stegs-modellen er krevende å løse, og vi utvikler derfor tre heuristikker. Den enkleste fikserer den beste førstestegsløsningen funnet med et såkalt mixed integer program (MIP) i løpet av en gitt tid. Den optimerer deretter andrestegsproblemet. De to andre heuristikkene benytter seg begge to av metoden simulated annealing (SA) i førstesteget. Den ene benytter seg så av en MIP i andresteget, mens den andre benytter seg av en grådig konstruksjonsheuristikk (GCH) i andresteget.

Den modifiserte MSSen er fleksibel ved at den tillater noen fleksible tidsluker is den kirurgiske timeplanen. Disse blir, i motsetning til fikserte tidsluker, tildelt en spesialitet først i andresteget når pasientetterpørselen er kjent. Mengden fleksible tidsluker i den modifiserte MSSen er en parameter i modellen. Vår analyse av verdien av å planlegge for usikkerhet i kombinasjon med fleksibilitet viser at det å øke fleksibilitet minker mengden umøtt etterpørsel opp til et gitt punkt. Med vår inputdata, der nedstrøms sengekapasitet ikke er begrensende, stagnerer verdien av fleksibilitet ved rundt 10%. Den forventede umøtte etterpørselen over en fire-ukers planleggingsperiode er da omtrent 35.7% lavere enn når modellen er løst uten fleksibilitet. Vi utfører også analyser på instanser med redusert nedstrøms sengekapasitet, men disse viser seg å være betydelig mer komplekse å løse. Ved å kraftig redusere antallet scenarier derimot, klarer vi å undersøke verdien av fleksibilitet med mer mindre kapasitet på sengepostene. Resultatene viser at verdien av fleksibilitet minker med reduksjonen i sengekapasitet. Med den gitte objektivfunksjonen og inputdataen vår ser vi ingen verdi i å planlegge for usikkerhet i pasientetterpørsel, med mindre vi også tillater fleksibilitet. Konklusjonen er dermed at både verdien av å modellere etterspørselen stokastisk og verdien av fleksibilitet, sannsynligvis er avhengig av inputdataen og hvordan man definerer verdi.

Abstract

The population in Norway and other developed countries is growing and aging. This makes it increasingly difficult to meet patient demand in hospitals. In the current four-year plan of the Norwegian Ministry of Health and Care Services, the goal is to meet patient demand in a sustainable manner (Ministry of Health and Care Services, 2020b). With up to 70% of all hospital patients needing surgical intervention (Bovim et al., 2020), efficient surgical scheduling is an important effort in reaching this goal. Surgical planning is intricate and time-consuming, but is mostly done manually. Operations research can be used to automate and optimize these tasks, and thus contribute to the goal of sustainable health care services.

This thesis deals with the tactical problem of creating a modified Master Surgical Schedule (modified MSS) for the Clinic of Surgery at St. Olavs Hospital. A modified MSS is a cyclic schedule that assigns operating rooms to surgical specialties on different days, and typically repeats itself for six to twelve months before being revised. The combination of an operating room and a day in an MSS is referred to as a slot. What separates a modified MSS from a regular MSS, is that some slots during each cycle are not fixed to a specialty. These can be assigned to the specialties that need them the most in different cycles. The objective of the planning tool is to minimize the expected amount of unmet demand by considering uncertainty in demand for different surgical procedures. The overall purpose of this thesis is to find both the value of taking uncertainty in patient demand into account, and the value of incorporating flexibility when assigning operating rooms in surgical scheduling. There are many factors to account for in surgical scheduling. We consider limitations in operating room capacities, staff and the capacities of bed wards for post-operative care.

The planning tool is implemented as a stochastic two-stage model. Distributional information of demand is known in the first-stage, while the actual demand for different surgical procedures is revealed in the second-stage. Surgical procedures have been aggregated to surgery groups by a clustering algorithm developed in Isaksen and Svagård (2021). Each surgery group belongs to a surgical specialty, and is scheduled into slots assigned to their affiliated specialty in the modified MSS. Uncertainty in demand is modeled by simulating demand scenarios for the different surgery groups and using these as input to the two-stage model. The clustering of surgical procedures and simulation of demand is based on 2019 data from the Clinic of Surgery at St. Olavs Hospital. Due to the stochastic two-stage optimization model being computationally demanding, three heuristic solution methods are developed. The first is a simple matheuristic that fixes the best first-stage solution found with a mixed integer program (MIP) in a given time, and then evaluates that solution further with a MIP. The two others are both implemented using simulated annealing (SA) in the first-stage. One is a matheuristic, combining SA in the first-stage with a MIP in the second-stage. The other is fully heuristic, using a greedy construction heuristic (GCH) in the second-stage.

The modified MSS allows us to incorporate flexibility in the surgical schedule in the form of flexible slots. As opposed to fixed slots, flexible slots are assigned to specialties in the second-stage, when patient demand is known. The amount of flexible slots in the modified MSS is given as a percentage out of all the slots. Our analyses of the value of planning for uncertainty in combination with flexibility, reveal that increasing flexibility reduces the expected amount of unmet demand up to a point. For our input data, where the downstream bed ward capacities are non-binding, the value of flexibility stagnates at around 10%. At this level, the expected unmet demand over a four week planning period is approximately 35.7% lower than if the model is solved with no flexibility. We also perform the analyses on instances with reduced capacity of downstream bed wards, but these are more complex to solve. When vastly reducing the number of scenarios, however, we are able to investigate the value of flexibility with more constraining bed ward capacities. The results show that the value of flexibility decreases with the reduction in capacity. With our objective function and input data, we are not able to detect value in planning for uncertainty in demand, unless also incorporating flexibility. In conclusion, both the value of modeling demand stochastically and incorporating flexibility are likely dependent on the input data and how we define value.

Glossaries

Table 1: Terminology related to the health care industry used in this thesis.

Term / Abbreviation	Explanation
Bed ward	Where patients may spend the night in the hospital post surgery
Elective surgery	A surgical procedure planned in advance
Emergency surgery	An unforeseen surgical procedure not planned in advance
EN	The Department of Breast and Endocrine Surgery
GN	The Department of Lower Gastrointestinal Surgery
GO	The Department of Upper Gastrointestinal Surgery
IC ward	Intensive care ward
ICU	Intensive care unit
Inpatient	Patients that stay in the hospital overnight
Intensive care	Care of patients that need close supervision
KA	The Department of Vascular Surgery
KENS	The medium care ward belonging to The Department of Breast and Endocrine Surgery
KGAS1	The medium care ward belonging to The Department of Lower Gastrointestinal Surgery
KGAS2	The medium care ward belonging to The Department of Upper Gastrointestinal Surgery
KKAS	The medium care ward belonging to The Department of Vascular Surgery
KURS	The medium care ward belonging to The Department Urology
LOS	Length of stay
Master Surgical Schedule	Repeating cyclic schedule that assigns time slots to surgical departments
MC ward	Medium care ward
MCU	Medium care unit
Medical specialty	A branch of medical practice that focuses on a group of related medical conditions
Modified MSS	MSS with some flexible slots
MSS	Master Surgical Schedule
OR	Operating room
Outpatient	Patients that do not stay in the hospital overnight
Time slot	A given operating room on a given day in the master surgical schedule
TOV	Intensive monitoring unit
UR	The Department of Urology

Table 2: Abbreviations and terminology related to optimization used in this thesis.

Term / Abbreviation	**Explenation**
A-BEESE	Adaptive balanced explorative and exploitative search with estimation
ACO	Ant colony optimization
BEESE	Balanced explorative and exploitative search with estimation
CSMIP	Mixed integer program with the alternative cutting stock inspired formulation presented in Appendix B implemented in Gurobi Optimizer
EEVS	Expectation of the expected value solution
ESD	Empirical standard deviation
EV	Objective value of the solution to the expected value problem
EV problem	Expected value problem
EVS	Solution to the expected value problem
FMIP	Mixed integer program with the base formulation presented in Chapter 5 implemented in Gurobi Optimizer
FMIP-2SMIP	Heuristic algorithm which runs FMIP first and then 2SMIP in the second stage at the end for the best obtained FSS
FSS	First stage solution
GA	Genetic algorithm
GCH	Greedy construction heuristic
GCHD	Greedy construction heuristic daywise
GCHP	Greedy construction heuristic periodwise
GCHS	Greedy construction heuristic slotwise
LBBD	Logic-based Bender's method
MIP	Mixed integer program
MSSP	Master surgical scheduling problem
Optimality gap	(Primal bound - dual bound) / primal bound
OV	Objective value
OVRP	Objective value of the recourse problem
R-BEESE	Randomized balanced explorative and exploitative search with estimation
RP	Recourse problem
RPS	Recourse problem solution
SA	Simulated annealing
SA-GCH-2SMIP	Heuristic algorithm with SA in the first stage and GCH in the second stage + 2SMIP in the second stage at the end for the best obtained FSS
SA-2SMIP	Heuristic algorithm with SA in the first stage and 2SMIP in the second stage
SAA	Sample average approximation
SS	Stochastic solution
TS	Tabu search
VSS	Value of stochastic solution
2SMIP	Mixed integer program with the base formulation presented in Chapter 5 with a fixed first stage solution implemented in Gurobi Optimizer

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

With a growing and aging population in Norway and other developed countries, it is becoming increasingly difficult to meet patient demand in hospitals. Furthermore, people are expecting more from the health care services, and expenses are increasing steadily (Hulshof et al., 2012). Up to 70% of all hospital patients are in need of some kind of surgical intervention (Bovim et al., 2020). This makes it increasingly important to exploit available hospital resources through efficient surgical scheduling in order to treat more patients (Bovim et al., 2020). According to Batista et al. (2020), the biggest challenges ahead are in reducing patient waiting lists and waiting times.

A great deal of scientific literature has been devoted to the optimization of health care services, focusing on different aspects of the industry and using different approaches. The common goal is to organize the health care services in both an effective and efficient manner (Hulshof et al., 2012). Despite the extensive research and applied changes in the industry, we see that there is still plenty of room for improvement. Surgical planning is intricate and time-consuming, but most of it is still done manually. It is furthermore done by health professionals, who are often untrained in optimization and whose time is better spent treating patients. Therefore, we add to the existing literature on efficient tactical surgical scheduling, by combining promising practices and trends. This thesis is based on and applied to data from the Clinic of Surgery at St. Olavs Hospital. Its overall purpose is to find both the value of taking uncertainty in patient demand into account and the value of incorporating flexibility when assigning operating rooms (OR) in surgical scheduling. We develop a mixed integer program (MIP) and propose three heuristic solution methods to solve it. For larger instances of the problem, the heuristic solution methods find good solutions in reasonable time, whereas the MIP solved by a commercial solver sometimes struggles to even find feasible solutions. In developing these solution methods, we furthermore aim to create a valuable prototype of a planning tool for surgical clinics.

The planning tool is implemented as a stochastic two-stage model. Distributional information of demand is known in the first-stage, while the actual demand for different surgical procedures is revealed in the second-stage. This allows parts of the decision-making to be postponed until the second-stage, when all uncertain parameter values have been revealed. Surgical schedules are often developed to be cyclic (Hulshof et al., 2012), which means that a fixed schedule is repeated periodically. A cyclic structure, compared to a non-cyclic structure, increases stability and predictability for surgical and downstream resources. However, a fully cyclic schedule leaves no room for adaptations to demand. Compared to a more flexible schedule, this increases the size of patient waiting lists (Hulshof et al., 2012). The objective of our MIP is to level patient demand by minimizing patient waiting lists. Therefore, we propose a surgical schedule that is partly cyclic and partly non-cyclic. In doing this, we seek to reap the benefits of a cyclic schedule while also allowing for some flexibility, thereby reducing the size of patient waiting lists. The model's output is a modified Master Surgical Schedule (MSS). A modified MSS is a partly cyclic schedule that assigns ORs to surgical specialties on different days, which is then repeated for a certain amount of time (Hulshof et al., 2012). It incorporates flexibility in the schedule by allocating some of the ORs to surgical specialties in the second-stage, when patient demand is known. Our main contribution to the literature is thus in adding to the practices of two-stage modeling and heuristic solution methods for such models, as well as integrating flexibility in surgical scheduling.

In the following paragraph, we present the overall purpose of each section in this thesis. Chapter 2 covers information about the industry in which this thesis is situated and the data on which it is based. We review relevant literature and our contribution to the literature in Chapter 3, before presenting relevant techniques and methods used. The optimization problem is described in detail in Chapter 4, including its restrictions, the objective and the decision delineation. We present the mathematical model in Chapter 5, including the assumptions made, the notation and the model formulation. In Chapter 6, we describe the logic behind the heuristic solution methods and explain how they take advantage of the problem structure. The process of generating interesting test instances from raw data and ensuring that the heuristics are implemented efficiently is covered in Chapter 7. The solution methods are tested and compared in Chapter 8, and the value of both stochasticity and flexibility is examined. Lastly, we present concluding remarks Chapter 9.

Chapter 2

Background

This section concerns the relevant industry and concrete case on which this thesis is based. In Section 2.1 we present the Norwegian health care system, before moving on to relevant industry terminology in Section 2.2. Finally, in Section 2.3 we present the hospital and clinic studied in this thesis as well as the aggregated data that has been used. The section has been shortened compared to the corresponding section in Asplin et al. (2021), but is otherwise identical to before.

2.1 The Norwegian Health Care System

Public hospitals in Norway are owned by the state and supervised by the Ministry of Health and Care Services (Ministry of Health and Care Services, 2020a). The Norwegian government renews its plan for the national health and hospitals every fourth year (Ministry of Health and Care Services, 2020b). This serves as a guideline for the direction in which the health services should develop during the period. The current goal is to meet patient needs in a sustainable manner. With the ongoing demographic changes and people expecting more from the health care services, it is becoming increasingly difficult to meet demand with the available resources. Consequently, sustainable health care services require that we exploit the available technology and staff competency, and that tasks are solved as efficiently as possible.

2.2 Surgical Care Terminology

Hospitals normally consist of clinics, which are further split into departments. The departments are commonly associated with a medical specialty, which is a branch of medical practice that focuses on a group of related medical conditions. The surgical clinic is a vital part of hospitals and has a large impact on the operations of the hospital as a whole (Hulshof et al., 2012). It performs surgeries on patients to repair injuries, correct defects and cure diseases. As the scheduling of surgical procedures affects departments throughout the hospital, one should also consider upstream and downstream resources when planning (Samudra et al., 2016). In order to more easily elaborate on these topics later on, we proceed by presenting some hospital- and surgical care-specific terminology.

Patients are normally split into two classes based on their characteristics, namely elective patients and emergency patients. The former are patients whose surgeries can be planned in advance, while the latter are patients whose surgeries are unforeseen and therefore need to be made room for on short notice (Samudra et al., 2016). This means that we know the elective patient demand for some time in advance, but we normally do not know the emergency patient demand until the day of surgery. Additionally, elective patients can be split into inpatients, meaning patients who stay at the hospital at least one night after their treatment for observation and care, and outpatients, meaning patients who leave on the day of their treatment.

When staying overnight, inpatients either stay at one of the medium care units (MCU) or the intensive care units (ICU), each consisting of one or more bed wards. Here, the patients are provided with care and are given a bed and food for at least one night (Hulshof et al., 2012). The ICU is for patients in need of intensive care through close supervision or specific medical equipment. The medical specialties at a hospital normally have an associated and separate MC ward, while IC wards are often shared resources. The length of stay (LOS) is the number of nights an inpatient stays at a ward, and the bed utilization in a ward is the ratio of occupied beds versus the total number of available beds.

Most elective patients are able to perform the necessary preparations for surgery at home. Some do, however, need to come in earlier for imaging and preparations. Afterward, the time spent in the operating theater can according to Isaksen and Svagård (2021) be divided into three phases, namely pre-knife, knife time and post-knife. Pre-knife consists of preoperative care and normally involves hair removal and anesthesia. This is either done in the operating room (OR) dedicated for the operation, or in a preoperative room. Knife time is the phase where the actual operation is executed. Post-knife is the time reserved for initial recovery when the patient wakes up from anesthesia. This is normally done in a separate area, namely the post-anesthesia care unit. Lastly, the patient is either sent home for recovery or admitted to one of the aforementioned MCUs or ICUs.

A common way of allocating hospital resources to medical specialties is through block scheduling. This involves assigning surgery groups to operating time blocks (Fei et al., 2010), also called time slots. In order to do this, surgery groups must be identified. This is normally based on the specialty that a patient's diagnosis belongs to, the patient's medical urgency and their resource requirements, such as expected surgery duration and LOS (Hulshof et al., 2012). Block schedules that repeat themselves periodically are called cyclic block schedules or master surgical schedules (MSS) (van Oostrum et al., 2008). These are normally set for six months to a year, namely the planning horizon. During this time the MSS is repeated every one to four weeks, depending on the length of a cycle.

2.3 Surgical Care Services at St. Olavs Hospital

The surgical clinic discussed in this thesis is part of St. Olavs Hospital. It is the University Hospital for Mid-Norway and is integrated with the Norwegian University of Science and Technology, located in Trondheim (St. Olavs Hospital, 2016d). St. Olavs Hospital has several regional functions for the more than 700,000 inhabitants of Møre and Romsdal and Trøndelag, such as patient treatment, research and education. Its three main locations are at Øya in central Trondheim, Orkdal and Røros, with the hospital at Øya being the largest.

The surgical clinic at St. Olavs Hospital has facilities in both Orkdal and at Øya in Trondheim. We focus on the latter location in this thesis, as they receive the most complicated diseases and treatments of patients, in addition to dealing with more uncertainty and resource demand per patient (Isaksen and Svagård, 2021). From here on out, the surgical clinic at Øya is referred to as the Clinic of Surgery, or simply as St. Olav. We delimit our study to five of the eight departments at the clinic, namely the Department of Upper Gastrointestinal Surgery, the Department of Lower Gastrointestinal Surgery, the Department of Urology, the Department of Vascular Surgery and the Department of Breast and Endocrine Surgery (St. Olavs Hospital, 2016c). Each department is associated with a medical specialty, also referred to as a surgical specialty, and one or more downstream care units.

The following sections are partly based on the thesis of Isaksen and Svagård (2021). We have been given access to anonymized raw patient data from 2019 from the Clinic of Surgery, as well as a cleansed and analyzed version of the same data by Isaksen and Svagård (2021). Furthermore, we have been given access to information that Isaksen and Svagård (2021) received during meetings with hospital staff, and have ourselves acquired additional information through a meeting with staff at the Clinic of Surgery.

2.3.1 Specialties and Care Units

The Department of Lower Gastrointestinal Surgery (GN) receives patients with illnesses in their lower abdomen or intestinal tract. The Department of Upper Gastrointestinal Surgery (GO) receives patients with illnesses in the upper part of their digestive system. Both departments have an associated MC ward, namely KGAS1 for GN and KGAS2 for GO. Both MC wards have 16 available beds (St. Olavs Hospital, 2016b).

The Department of Urology (UR) carries out treatments on patients with illnesses in the kidney, urinary tract and prostate areas. The associated MC bed ward, KURS, has a capacity of 16 beds (St. Olavs Hospital, 2016e).

The Department of Vascular Surgery (KA) treats patients with medical conditions in their veins (St. Olavs Hospital, 2016f). The associated MC bed ward, KKAS, has a capacity of 9 beds (Isaksen and Svagård, 2021).

The Department of Breast and Endocrine Surgery (EN) takes in patients with vascular, breast and hormone-producing gland illnesses (St. Olavs Hospital, 2016a). The associated MC bed ward, KENS, has a capacity of 3 beds (Isaksen and Svagård, 2021).

Additionally, the Clinic of Surgery has access to two ICUs. One is simply called the ICU and is a shared resource with the other departments at the hospital. The other is the intensive monitoring unit (TOV), which is only used by the Clinic of Surgery and has a capacity of 7 beds (Isaksen and Svagård, 2021).

The aforementioned bed capacities represent the maximum number of beds that are available during the weekdays. The bed capacities are slightly reduced during the weekends, as presented in Table 2.1. The number of beds in the ICU is the average number of beds available to the Clinic of Surgery, as it is a shared resource (Isaksen and Svagård, 2021).

Table 2.1: Specialties' affiliation to the bed wards with their corresponding bed capacities during weekdays and weekends.

Bed Ward	**Specialty**		**Number of Beds**	
			Weekdays	**Weekends**
KGAS1	Lower Gastrointestinal	GN	16	14
KGAS2	Upper Gastrointestinal	GO	16	14
KURS	Urology	UR	16	12
KKAS	Vascular	KA	9	7
KENS	Breast and Endocrine	EN	3	2
TOV	All	-	7	5
ICU	All	-	4	1

2.3.2 Operating Rooms

The previously mentioned departments have in total seven ORs dedicated to elective surgeries. Their opening hours are 07:30 to 15:30 during the weekdays, with the possibility of extending the opening hours to 17:00. Each OR can only be utilized by a subset of the specialties, as indicated in Table 2.2. (Isaksen and Svagård, 2021)

In addition to the elective ORs, the Clinic of Surgery has access to three of the emergency ORs in the Emergency and Cardiothoracic Centre during the day (Isaksen and Svagård, 2021). Two of them are shared with the other clinics at the hospital, while the third is exclusive to the Clinic of Surgery. During the night, only one emergency OR is available across all clinics.

Table 2.2: The affiliation of specialties and operating rooms. Dark gray indicates that an OR is suitable for the specialty.

Specialty \ OR	GA-1	GA-2	GA-3	GA-4	GA-5	GA-6	GA-7
GN				■	■		■
GO				■	■		■
UR	■	■	■			■	
KA		■					
EN	■	■					

Although the emergency and elective ORs are separated, there are three occasions on which an emergency patient may be operated on in an elective OR. If there is idle time in an elective OR and there are emergency patients waiting for surgery, they may fill the available time window. Furthermore, an emergency patient may be operated on in an elective OR if their case is sufficiently urgent, thus delaying the planned elective procedures. Lastly, an emergency patient with a complicated medical condition may require a certain level of surgical expertise or experience. If the surgeons working in the elective line are more experienced and specialized within the field, the emergency patient is transferred to the elective line. This may also result in delays in the planned elective procedures. Emergency patients may only be scheduled to an elective OR if the OR is assigned to the specialty that the patient's diagnosis is affiliated with. The assignment of specialties to ORs will be discussed in more detail in the following section.

2.3.3 Surgery Scheduling

The current schedule at the Clinic of Surgery is a yearly MSS. It consists of a fixed two-week cycle, and specifies which specialty has access to a given OR on a given day in the cycle. Furthermore, it specifies whether or not the opening hours are extended for a given slot, which hereafter will be referred to as an extended slot. Experience has shown that full-day slots, where an OR is assigned to a specialty for an entire day, result in fewer cancellations than half-day slots. The MSS for the Clinic of Surgery in 2019 was therefore as presented in Table 2.3 (Isaksen and Svagård, 2021). We choose to refer to the MSS from 2019, since the data used in this thesis is also from 2019. The Admission Office is responsible for assigning patients to time slots. This is done manually, and is normally completed one to three weeks ahead of the surgery date. The clinic has a weekly meeting on Thursdays where they discuss the surgery schedule for the following week and make adjustments if necessary.

Table 2.3: MSS in 2019 at St. Olav. Each cell represents a slot, defined by an OR, the specialty it is assigned to and the day of the two-week cycle. Weekends (W) are excluded from the MSS.

	Odd Weeks						**Even Weeks**					
Room \ Day	**1**	**2**	**3**	**4**	**5**	**W**	**8**	**9**	**10**	**11**	**12**	**W**
GA-1	EN	EN*	UR	EN*	GN	-	EN	EN*	EN	EN*	-	-
GA-2	UR	EN	UR	UR	EN	-	UR	KA	UR	UR	EN	-
GA-3	UR	UR*	UR*	-	-	-	UR	UR*	UR*	-	-	-
GA-4	GN*	GN*	GN*	GN*	GN	-	GN*	GN*	GN*	GN*	GN	-
GA-5	GO*	GO*	GO*	GO*	GO	-	GO*	GO*	GO*	GO*	GO	-
GA-6	UR	UR	UR	UR*	UR	-	UR	UR	UR	UR*	UR	-
GA-7	GO*	GN	GO	GN	-	-	GO*	GN	GO	GN	GN	-

'*' indicates an extended slot

Chapter 3

Related Literature

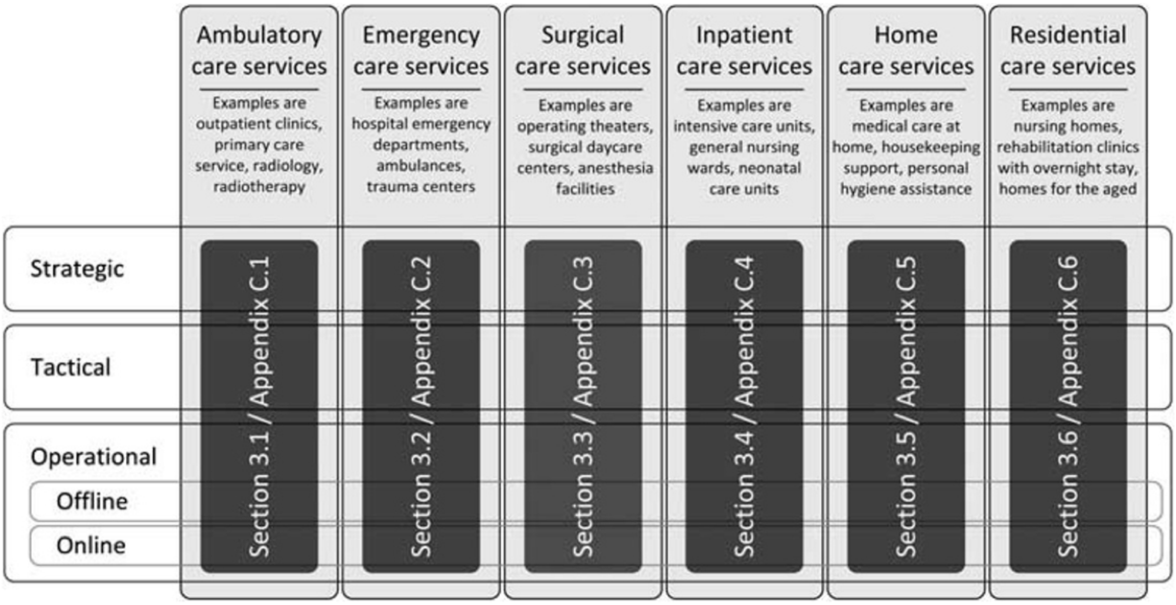
In the following sections, we review existing literature that is relevant to this thesis. First, we present what decisions must be made at what time in healthcare services in Section 3.1. Recent surgery scheduling literature is categorized in Section 3.2. Next, we discuss in Section 3.3 how this thesis contributes to the existing literature. Lastly, in Section 3.4, we introduce techniques and methods within operations research that are particularly relevant to this thesis. Section 3.1 has been shortened, but is otherwise identical its corresponding section in Asplin et al. (2021). Section 3.2 and Section 3.3 are both inspired by Asplin et al. (2021), but with substantial alterations.

3.1 Planning Decisions in Healthcare

Within the healthcare industry, decisions about how resources should be distributed in order to meet current and future demands are constantly being made. This is called resource capacity planning, and can in healthcare be classified based on when a decision is being made and within which healthcare service. Hulshof et al. (2012) is one of many to propose such a taxonomy, where the timing is spread out on the vertical axis and the different healthcare services make up the horizontal axis. Section 3.1 is primarily based on the work of Hulshof et al. (2012), focusing on the parts of the taxonomy most relevant to our study.

The vertical levels were first proposed by Anthony (1965), and consist of strategic planning, tactical planning and operational planning. The strategic level has a long planning horizon and concerns an organization's long-term goals. It involves structural decision making, like a facility's location and the dimensioning of resource capacities (Hulshof et al., 2012). The tactical level has a medium planning horizon and typically involves the construction of a master schedule, where staff is assigned to tasks in given time periods (Samudra et al., 2016). The operational level has a short planning horizon and mostly involves staffing from day to day and the scheduling of patients (Samudra et al., 2016). This level can be further split into offline and online planning. Offline operational planning is done before the schedule execution, when elective demand is known, but emergency demand is not (Hulshof et al., 2012). Online operational planning is done during schedule execution and mostly involves reacting to unplanned events, which are common in the healthcare industry (Hulshof et al., 2012).

Hulshof et al. (2012) suggests a horizontal axis consisting of six healthcare services. These services are strongly connected, as a patient may, and often will, move between them. Figure 3.1 displays the full taxonomy as proposed by Hulshof et al. (2012), with both the vertical and horizontal levels. We now proceed by presenting the two healthcare services that are relevant to surgical clinics and to this thesis. For each of them, we look at decisions that need to be made on the strategic, tactical and offline operational planning levels, focusing especially on tactical planning.



Hulshof et al. 2012

Figure 3.1: Full taxonomy for resource capacity planning and control in the healthcare services.

3.1.1 Surgical Care Services

The surgical care services perform operations on patients in order to repair injuries, correct defects and cure diseases. Most strategic surgical planning has already taken place at St. Olav, and is therefore not within the scope of this thesis.

Tactical planning
As previously mentioned, the resources set at the strategic level are allocated to specialties at the tactical level. The cyclic structure of an MSS, compared to a non-cyclic block schedule, increases stability and predictability for surgical and downstream resources. However, a cyclic schedule leaves no room for adaptations to demand. Compared to a more flexible schedule, this increases the size of patient waiting lists (Hulshof et al., 2012). Hulshof et al. (2012) and Fei et al. (2010) suggest modified block scheduling as a compromise between a non-cyclic and a cyclic plan. Here, only a part of the operating time capacity is assigned as in block scheduling, while the remainder of the capacity is planned at a later stage. This leaves some ORs on some days free for adapting to fluctuations in patient demand. When the block schedule has been set, an admission policy must be made. Taking the schedule into account, it determines how many surgeries should be performed from each surgery group on each day (Adan et al., 2009). It must do this while balancing staff satisfaction, patient service and resource utilization (Hulshof et al., 2012).

Operational planning
At the offline operational planning level, planned surgical cases are assigned to an OR, date and time of the day. Surgical case scheduling can be done integrally in one step, but is often decomposed into several steps (Hulshof et al., 2012). First, the planned length of the surgical case must be decided, which includes cleaning time and slack time. There's a trade-off between reserving too little slack time, which leads to staff overtime and patient waiting time, and reserving too much slack time, which results in idle time in the OR. Second, each surgical case must be assigned a date in an OR, and then a starting time (Hulshof et al., 2012). Surgical case scheduling is often done in isolation. It should, however, take into consideration other care services, such as the inpatient care services in order to avoid full bed wards, thus ensuring an increase in efficiency.

3.1.2 Inpatient Care Services

The inpatient care services are where hospitalized patients are provided with care and are given a bed and food for at least one night (Hulshof et al., 2012). Both the MCUs and the ICUs are part of the inpatient care services.

Strategic planning
As mentioned before, strategic planning is mostly beyond the scope of this thesis, also within the inpatient care services. However, there are some minor changes, or at least assumptions, that may be relevant at the strategic level. The inpatient care facility at a hospital is normally divided into several medical care units, each with its designated staff, beds and equipment. The aim is to ensure that patients are taken care of by appropriately skilled nurses with the right equipment, while using the available resources efficiently. Each specialty normally has its own, dedicated care unit, although this is not necessarily optimal. Merging care units, also called pooling, could lead to economies of scale and improved efficiency (Hulshof et al., 2012). One should, however, be careful when pooling care units for surgery groups in need of different service levels or nursing skills, as it requires the highest service level and nursing competence for all patients in the ward (Hulshof et al., 2012). It may also induce costs for installing extra equipment on all beds (Hulshof et al., 2012). Thus, the decision of whether to merge care units or not is case-dependent, and should be based on an evaluation of utility versus costs.

Tactical planning
At the tactical level of inpatient care services, bed reallocation is the first step to consider (Hulshof et al., 2012). Medium-term demand forecasts may show that decisions made at the strategic level are not optimal. With a somewhat flexible layout, beds can be moved around to meet the anticipated changes in demand. Although beneficial, one should not overlook the costs related to changing bed capacity and the effects on staff planning. Admission control determines what patients can be admitted to a given ward at a given time (Hulshof et al., 2012). It involves policies that aim to match demand and supply, such that cancellations and misplacements are minimized, while bed occupancy is maximized. Earmarking is a common admission control policy where a number of beds in a ward are reserved either for emergency or elective patients (Hulshof et al., 2012). Overflow policies decide what should be done when all reserved beds for a patient type are occupied. Policies that allow for some overflow take advantage of the same benefits as pooling. Lastly, surgical and inpatient care services should coordinate in order to avoid substantial differences in demand and therefore cancellations in elective surgeries.

Operational planning
At the offline operational planning level, elective patients are assigned to a bed in a ward where their medical needs will be met (Hulshof et al., 2012). This is normally done a few days prior to the scheduled surgery.

3.2 Operating Room Planning and Scheduling

Optimization has several applications in healthcare services, but in this thesis we focus on the tactical planning level of surgical care services in particular. There are many things to consider when studying hospital optimization that will define the scope and the outcome of the study. Fortunately, many researchers have tackled surgical scheduling before us. This helps us to identify promising practices and trends to consider in our study, as well as less explored approaches. We will in the following section classify recent surgery scheduling literature through a simple framework based on the work of Samudra et al. (2016). It is structured based on 6 different descriptive fields which are either problem or technically oriented, and studies 29 selected papers.

3.2.1 Patient Characteristics

As previously mentioned, hospital patients are normally split into elective patients and emergency patients, based on the urgency of their treatment needs. The amount of literature related to elective patients is extensive compared to that of emergency patients (Samudra et al., 2016). Out of the 29 papers we have studied, all of them include elective patients, but only nine of them also model emergency patients. One can further split elective patients into inpatients, meaning patients who stay overnight, and outpatients, meaning patients who leave the hospital on the day of their treatment. Many papers focus mainly on inpatients due to their inherent demand for care unit resources (Vanberkel et al., 2011; Fügener et al., 2014; Batista et al., 2020). Although we are seeing a substantial shift towards more outpatient care, the amount of literature focusing on these patients remains stagnant (Samudra et al., 2016).

When studying surgical scheduling, one should at least account for emergency patients (Samudra et al., 2016). In practice, the unplanned patients can be taken care of in several ways. One way is to integrate them as part of the elective schedule, normally through planning extra slack throughout the day. Both Adan et al. (2009) and Adan et al. (2011) incorporate a reservation policy for emergency patients, meaning they reserve buffer capacity for these patients. Another approach is to separate them from the elective schedule through dedicated emergency ORs, like Cappanera et al. (2014) and Fügener (2015). Although the latter benefits from a decrease in both overtime and the number of elective cancellations, it also reduces the available surgery time for elective patients.

Patients are often categorized based on the surgical specialty they belong to and their surgery type, making up surgery groups, as in van Oostrum et al. (2008) and Visintin et al. (2016). This allows for a generalization of patients in optimization models. Specialties are normally assigned to different ORs, allowing for a partly independent surgical scheduling. Different surgery types from a given specialty can, however, be performed in the same OR and should be scheduled efficiently.

3.2.2 Performance Measures

In optimization modeling, the performance measure chosen will favor one or more stakeholders over the others. Some researchers attempt to include the interests of several stakeholders through multi-objective optimization. Schneider et al. (2020) attempts to maximize OR utilization while minimizing the variance of the bed usage at wards. Bovim et al. (2020) maximizes the number of elective patients scheduled for surgery while minimizing the number of elective cancellations and the number of patients staying in wards not meant for them. Several papers solve their optimization problem separately for different objectives, such as Mannino et al. (2012). These approaches are all ways of incorporating various performance measures in order to consider several stakeholders' interests.

Overtime is, according to Samudra et al. (2016), the most commonly used performance measure. It can have several undesirable consequences, such as surgery cancellations, dissatisfaction of staff, downstream schedule interference and increased costs for the hospital. Both Kamran et al. (2018) and Heydari and Soudi (2016) incorporate the minimization of overtime as part of their multi-objective. Mannino et al. (2012) seeks to minimize overtime hours, while incorporating a light robustness modeling approach through setting a fixed maximum length on patient queues. Patient waiting time is another common performance measure. It is most often related to the waiting time on the day of the surgery due to delays, but also access time, corresponding to the size of the patient waiting list. Kamran et al. (2018) incorporates the latter as part of its multi-objective function. There are other frequently used performance measures, such as maximizing hospital revenue (Bruni et al., 2015; Fügener, 2015) or minimizing hospital costs (Lamiri et al., 2009; Wang et al., 2014; Kim and Mehrotra, 2015), leveling of the bed utilization in the downstream care units (Isaksen and Svagård, 2021; van Oostrum et al., 2008; Beliën and Demeulemeester, 2007), maximizing OR utilization (van Oostrum et al., 2008; Fei et al., 2010; Schneider et al., 2020), minimizing idle time (Fei et al., 2010), maximizing throughput (M'Hallah and Visintin, 2019; Spratt and Kozan, 2016) and maximising overall patient satisfaction through adjusting to their preferences (Makboul et al., 2021).

3.2.3 Decision Delineation

Decision delineation indicates what type of decisions are being made within surgery scheduling and to whom they apply. Common decisions in tactical planning involve assigning specialties, surgeons or patients to a date, time, OR or amount of capacity. Some papers take on several of these decisions. While Fei et al. (2010) first assigns a date to each patient and then an OR and a time, Beliën and Demeulemeester (2007) first assigns a specialty to each slot and then each patient to a slot and time.

A substantial part of the literature addresses the scheduling of surgery groups instead of individual patients (Samudra et al., 2016). Most of the papers we have studied, such as Neyshabouri and Berg (2017), Kumar et al. (2018) and Vanberkel et al. (2011), group patients in such a manner. Furthermore, the assignment of ORs and dates, i.e. slots, has received the most focus. In both Bovim et al. (2020) and Cappanera et al. (2014), specialties are assigned to slots. It is common to assign patients to a slot first, and then later on to a starting time within the slot. This is because the former is more easily planned weeks ahead, while the latter is normally determined closer to the surgery date. Fei et al. (2010) does exactly this in two stages. The assignment of capacity normally involves assigning slots to specialties, often resulting in an MSS. Many of the studied papers do this, with some examples being Cappanera et al. (2014), Mannino et al. (2012), Schneider et al. (2020) and Makboul et al. (2021).

3.2.4 Up- and Downstream Facilities

Surgical scheduling affects both up- and downstream services within the hospital. An integrated approach takes this into account when performing surgical scheduling, while an isolated approach does not. According to Samudra et al. (2016), both approaches are equally common in the literature. The majority of the papers we have studied use an integrated approach. An example of an upstream service to consider is outpatient clinics. Due to the scope of this thesis, however, we focus on literature accounting mainly for downstream resources. Bovim et al. (2020) and M'Hallah and Visintin (2019) both consider the availability at the downstream bed wards when developing an MSS. Several of the papers do, however, take an isolated approach for simplicity, such as Abdeljaouad et al. (2020) and Kamran et al. (2018).

The MSS greatly affects the bed management for inpatient care services. Fügener et al. (2014) develops an MSS with the objective of minimizing the costs in the care units. Furthermore, Beliën and Demeulemeester (2007), Vanberkel et al. (2011) and Isaksen and Svagård (2021) all optimize the MSS in order to level the expected bed occupancy in the downstream wards.

3.2.5 Uncertainty

Several elements of surgical care services contain uncertainty, making the task of surgery scheduling challenging. While deterministic approaches ignore uncertainty, stochastic approaches attempt to take uncertainty into account. According to Samudra et al. (2016), most stochastic models in the literature incorporate uncertainty in the form of emergency patient arrivals, like Bovim et al. (2020), or surgery durations, like Schneider et al. (2020). Some also incorporate uncertainty in the LOS post surgery at care units, such as Kumar et al. (2018), and many model the uncertainty of several factors. M'Hallah and Visintin (2019) looks at uncertainty in both surgery durations and LOS. Bruni et al. (2015) assumes stochasticity in both emergency patient arrivals and surgery durations. Incorporating some sort of uncertainty is not uncommon, as more than half of the papers studied in Samudra et al. (2016) model stochasticity.

One way of incorporating stochasticity is through multi-stage modeling, more commonly reduced to two-stage modeling. The latter splits the problem into two stages, where first-stage decisions are made before knowing the value of the stochastic parameters, while second-stage decisions are made after observing their outcome (King and Wallace, 2003). The objective is to find a first-stage solution that is expected to perform well when taking all possible parameter outcomes into account. Makboul et al.

(2021) initially assigns specialties to slots in the first-stage, based on all possible parameter outcomes. It then decides upon the optimal number of surgeries to perform in a slot in the second-stage, after knowing the parameter realizations. Both Kumar et al. (2018) and Kim and Horowitz (2002) take a different approach by planning the complete schedule in the first-stage. The recourse decisions in the second-stage therefore determine cancellations and alterations that need to be made after the actual realization of the parameters. Two-stage modeling is discussed in further detail in Section 3.4.1.

Another way of adapting to uncertainty in data is through incorporating flexibility in the model. Visintin et al. (2016) incorporates the flexible management of several resources, concluding that there are substantial benefits to managing either surgical teams or ORs flexibly. The first enables swaps between specialties assigned to slots, and the second enables a more flexible assignment of surgeries to slots, meaning both surgeries that are expected to result in a long LOS and a short LOS can be assigned to the same slot. Oliveira et al. (2021) also allows for flexibility by allowing changes to the MSS to be made on a weekly or monthly basis, following the dynamic patient demand. A flexible rolling horizon and flexible long-term solution are compared to a static long-term solution. Oliveira et al. (2021) finds that flexibility improves key performance indicators, such as total waiting time, tardiness and throughput. Although flexibility positively affects the objective value, it negatively affects the stability in the schedule. This creates a trade-off, where flexibility in the MSS makes it easier to meet demand, and stability in the MSS makes it easier to satisfy staff (Oliveira et al., 2021).

3.2.6 Operations Research Methodology

The research methodology used in a paper provides information about the type of analysis that is performed and the solution or evaluation technique that is applied (Samudra et al., 2016). Several of the papers studied incorporate discrete-event simulation in order to examine the performance of their models. Examples are Cappanera et al. (2014) and Mannino et al. (2012). Some use sample average approximation (SAA) to solve stochastic optimization problems, such as M'Hallah and Visintin (2019) and Kamran et al. (2018). Adan et al. (2011) uses goal programming, van Oostrum et al. (2008) uses column generation and most papers use mixed integer programming (MIP). Lastly, several of the studied papers use heuristics to find good solutions more quickly. Beliën and Demeulemeester (2007) develops a repetitive MIP heuristic, which involves sequentially solving a set of MIPs. After each MIP, another constraint is added to the model, thereby limiting the solution space. Fügener et al. (2014) uses an incremental improvement heuristic, where only one swap of two slots is allowed to an existing MSS. This is repeated for a defined number of maximum swaps. Lamiri et al. (2009) incorporates several heuristics, including the multi-start method, tabu search (TS) and simulated annealing (SA).

SA is one of the most commonly used heuristics for scheduling problems (Abdeljaouad et al., 2020). The algorithm continuously moves from the current solution to a neighboring solution, accepting worse solutions with a probability that decreases over time (Spratt and Kozan, 2016). This allows SA to escape local optimums while searching for the global optimum. Fügener (2015) and Abdeljaouad et al. (2020) both start with an initial MSS solution, either received from the hospital or created through a construction heuristic, which is then iteratively improved through an SA algorithm. Both Lamiri et al. (2009) and van Essen et al. (2014) advance their SA algorithm through continuously keeping track of the best solution found so far. In the studied papers, neighborhood structures from which one chooses the next solution are found in several ways. One common approach is through swaps, where a specialty operating on one day is swapped with a specialty operating on another day, i.e. a swap of two slots. The neighborhood then consists of all feasible swaps with the current solution. Vanberkel et al. (2011), van Essen et al. (2014), and Schneider et al. (2020) all implement such neighborhood structures.

3.3 Our Contribution to the Literature

In this thesis, we develop a planning tool for the tactical level in surgical care services. We do so through a two-stage stochastic model, accounting for the uncertainty in elective patient demand. We propose four solution methods, ranging from an exact MIP that can solve small instances of the problem, to an entirely heuristic SA algorithm that is applicable to larger instances. Its outcome is a modified MSS, where specialties are assigned to slots. Flexibility is incorporated by leaving a percentage of the slots open, thereby allowing some decisions to be made at a later stage when new information about elective patient demand is known. The focus is on elective patients, including both inpatients and outpatients, and downstream care units are integrated into the model. The objective is to minimize the total length of the elective patient waiting list. Our main contribution to the literature is thus in adding to the practices of two-stage modeling and heuristic solution methods for such models, in addition to incorporating flexibility in the MSS. The value of flexibility is investigated for various levels of flexibility and under different conditions, such as with varying bed ward capacities.

Table 3.1 and Table 3.2 show some of the most relevant characteristics of the studied papers, compared to this thesis. The ways in which they were found, either through other papers or through search phrases on Google Scholar, are presented in Appendix E.

Table 3.1: The table summarizes some of the most relevant characteristics of the studied papers, compared to this thesis. The last three columns indicate whether or not a paper incorporates an MSS, whether or not it integrates downstream units and whether or not it makes use of heuristic solution methods, respectively.

Paper	**Patients**	**Performance measure**	**Uncertainty**	**MSS**	**Integrated**	**Heuristic**
This thesis	Elective	Min patient list	Stochastic elective patient demand + two-stage	✓	✓	✓
Isaksen and Svagård (2021)	Elective + emergency	Min peak bed occupancy level	Stochastic LOS	✓	✓	
van Oostrum et al. (2008)	Elective	Multi-objective: Max OR utilization + level bed utilization		✓	✓	
Adan et al. (2009)	Elective	Multi-objective: Min overutilization + min underutilization	Stochastic LOS	✓	✓	
Fei et al. (2010)	Elective	Multi-objective: Max OR utilization + min overtime costs + min idle time		✓	✓	✓
Cappanera et al. (2014)	Elective	Min max ORs and beds utilization + min gap between max and min values of ORs and beds utilization + min sum of quadratic overrun of ORs and beds utilization from threshold		✓	✓	
Mannino et al. (2012)	Elective	Min overtime + balance patient list length		✓		
Beliën and Demeulemeester (2007)	Elective	Min bed shortage	Stochastic number of patients operated per day	✓	✓	✓
Vanberkel et al. (2011)	Elective	Balance ward occupancy		✓	✓	
Fügener et al. (2014)	Elective	Min cost in downstream units	Stochastic LOS	✓	✓	✓
Adan et al. (2011)	Elective + emergency	Min deviations of the resources' consumption to the target levels	Stochastic LOS		✓	
Bruni et al. (2015)	Elective + emergency	Max revenues	Stochastic emergency patient demand + surgery duration			✓
Lamiri et al. (2009)	Elective + emergency	Min overtime costs and patients' related costs	Stochastic emergency patient demand			✓
Wang et al. (2014)	Elective + emergency	Min expected operating costs	Stochastic emergency patient demand + surgery duration			✓
Bovim et al. (2020)	Elective + emergency	Max number of elective patients scheduled for surgery + min number of elective cancellations + min number of patients waiting in wards not designated for them	Stochastic two-stage	✓		

Table 3.2: The table summarizes some of the most relevant characteristics of the studied papers, compared to this thesis. The last three columns indicate whether or not a paper incorporates an MSS, whether or not it integrates downstream units and whether or not it makes use of heuristic solution methods, respectively.

Paper	**Patients**	**Performance measure**	**Uncertainty**	**MSS**	**Integrated**	**Heuristic**
Fügener (2015)	Elective	Max revenues	Stochastic elective patient demand	✓	✓	
M'Hallah and Visintin (2019)	Elective	Max throughput	Stochastic LOS + surgery duration + two-stage		✓	
Batista et al. (2020)	Elective + emergency	Multi-objective: Min resource utilization + min costs	Stochastic two-stage		✓	
Makboul et al. (2021)	Elective	Max time between surgery date and due date	Stochastic two-stage	✓	✓	
Neyshabouri and Berg (2017)	Elective	Min costs	Stochastic two-stage	✓	✓	
Kamran et al. (2018)	Elective + emergency	Multi-objective: Min waiting time + min tardiness + min cancellations + min overtime + min number of surgery days for each surgeon	Stochastic surgery duration + two-stage			
Heydari and Soudi (2016)	Elective + emergency	Multi-objective: Min makespan + min overtime	Stochastic two-stage			
Spratt and Kozan (2016)	Elective	Max number of surgeries performed	Stochastic surgery duration	✓		✓
Abdeljaouad et al. (2020)	Elective	Min opening duration + min surgeons' waiting times	Stochastic			✓
Visintin et al. (2016)	Elective	Max number of scheduled surgeries		✓	✓	
van Essen et al. (2014)	Elective + emergency	Many different objectives			✓	✓
Schneider et al. (2020)	Elective	Max OR utilization + min variance of bed usage at wards	Stochastic surgery duration	✓	✓	✓
Kumar et al. (2018)	Elective	Max throughput	Stochastic LOS	✓	✓	
Kim and Mehrotra (2015)		Min staffing costs + min resource costs	Stochastic two-stage		✓	
Oliveira et al. (2021)	Elective	Min deviations of the assigned OR time to target value		✓	✓	✓

3.4 Relevant Techniques and Methods in Operations Research

This section is meant to give an overview of techniques and methods that are particularly relevant to this thesis. We start by giving an introduction to stochastic programming in Section 3.4.1, mainly based on King and Wallace (2003). In Section 3.4.2 we discuss methods for generating scenarios in stochastic models, mainly based on Kaut and Wallace (2003). Section 3.4.3 concerns the SAA method for evaluating the performance of stochastic models. This last part is based on Ahmed and Shapiro (2002), as well as Shapiro and Philpott (2007). Lastly, in Section 3.4.4 we introduce several solution methods for stochastic optimization problems of varying problem sizes and complexity. Note that due to the lack of consistency in mathematical notation throughout this literature, the notation in this thesis may deviate somewhat from the cited works.

3.4.1 Stochastic Programming

Stochastic programming is a part of mathematical programming and operations research that studies how to incorporate uncertainty into decision making (King and Wallace, 2003). The motivation behind it is to obtain solutions that perform well in problems containing uncertainty. Ignoring such uncertainties can lead to serious issues when modeling decisions are put to life in the real world, and unanticipated events occur. What may look like a perfect solution for a given set of parameter values, can perform terribly for a very similar set of values. This is often referred to as a *knife-edge* property, and is the reason that deterministic models often perform poorly on average (King and Wallace, 2003). Stochastic programs, however, model what might happen and how to act in these different situations. By doing this properly, decisions that perform better on average can be made, since they account for a range of parameter values.

In the real world, decisions made at one point in time often have implications for decisions and outcomes in the future. Stages are the points in time where decisions are made in a model. Introducing stages enables us to model how decisions made at one point in time impact decisions made after new information has surfaced. This is an important concept in stochastic programming. It is worth mentioning that modeling stages in deterministic models does not make sense, as no new information is revealed in a deterministic setting.

The simplest form of a model with stages is an inherently two-stage model, often called an invest-and-use model. In these models, the first-stage decision is a major long-term decision or investment, while the remaining second-stage decisions represent the use of this investment. Second-stage decisions are often referred to as recourse decisions, as they can be considered corrective actions after the realization of a random variable. In this section we only cover models with relative complete recourse, which means that no first-stage decision can yield an infeasible problem in the second-stage (Shapiro and Nemirovski, 2005). Ahmed and Shapiro (2002) proposes a standard formulation for such a two-stage model with recourse.

$$\min_{x \in X} \{f(x) := c^T x + \mathbb{E}[Q(x, \xi(\omega))]\} \tag{3.1}$$

$$Q(x, \xi(\omega)) := \inf_{y \in Y} \{q^T y : Wy \geq h - Tx\} \tag{3.2}$$

Here, $f(x)$ is the objective function, x is the first-stage decisions with its feasible set X and c^T is an arbitrary cost matrix. (3.2) is called the the recourse function, and depends on both the first-stage decisions and an uncertain parameter vector $\xi(\omega) = (q(\omega), T(\omega), W(\omega), h(\omega))$. ω represents a realization of some randomness. The recourse function includes second-stage decision variables y with its feasible set Y and it connects the first- and second-stage as it has corrective abilities after the realization of the random parameter vector. In (3.2) the expectation of the recourse function is taken with respect to the probability distribution of $\xi(\omega)$. Note that all parameters do not need to be random and that there is an infinite number of realizations of ω if the probability distribution of $\xi(\omega)$ is continuous.

Shapiro and Nemirovski (2005) discuss several questions that naturally arise with respect to such a formulation. They consider topics like the probability distribution, the discretization of a continuous world, the complexity of the problem, and whether using expectation in the second-stage is even an appropriate measure. Nevertheless, these topics are not within the scope of this thesis.

Even though stochastic models seek to deal with an uncertain future, it is necessary to make assumptions about the future in order to construct such a model. For stochastic programming to be applicable, the distribution of the future must be discrete. This discretization of the future is called a scenario tree, and each discrete event of this tree is called a scenario. A scenario contains one realization of each random variable present in the model, as illustrated by Figure 3.2.

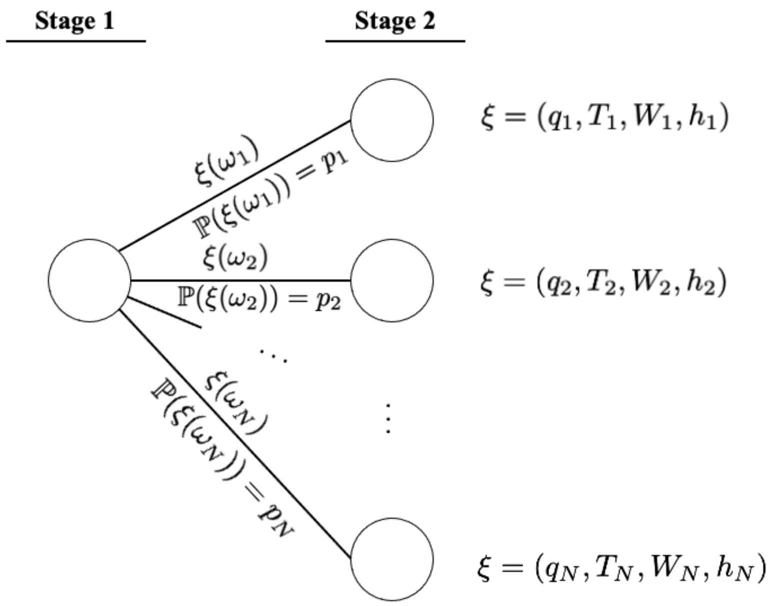


Figure 3.2: A visualization of a general scenario tree with the presented notation

If we assume a discrete probability distribution with a finite number of realizations, $\{\omega_1, \omega_2, ...\omega_N\}$ with probabilities $\{p_1, p_2, ...p_N\}$, we can describe the stochastic process with a scenario tree $\boldsymbol{\xi}_N = \{\xi(\omega_1), ...\xi(\omega_N)\}$. N is the number of scenarios in the second-stage of the scenario tree. In such a discrete world, it is possible to describe the deterministic equivalent of the problem as follows:

$$\min_{x \in X, y \in Y} \{c^T x + \sum_{i=1..N} p_i q_i y_i : W_i y_i \geq h_i - T_i x\}, \tag{3.3}$$

where y_i is the recourse decision in scenario i, q_i, W_i, h_i and t_i are the parameter realizations in scenario i and p_i is the probability of scenario i occurring. This deterministic equivalent is more practical for computational reasons as it is possible to model discretely. The general information and decision structure of a two-stage problem is presented in Figure 3.2.

One can to some degree evaluate how well the two-stage model performs versus a completely deterministic model. This evaluation is referred to as the value of the stochastic solution (VSS), and is the difference between the expectation of expected value solution (EEVS) and the optimal objective value of the recourse problem (OVRP). The expected value solution (EVS) is found by fixing the uncertain parameters to their expected values, and solving the resulting expected value (EV) problem. The EEVS is then the expected performance of the EVS across all scenarios used in the recourse problem (RP). A practical way of doing so is by fixing the solution in the RP to the EVS and solving it. The algorithm is presented below.

1. Solve the two-stage model with a scenario tree $\boldsymbol{\xi}_N$ with N scenarios in the second-stage, and record the OVRP

$$OVRP = \min_{x \in X, y \in Y} f(x, y; \boldsymbol{\xi}_N) \tag{3.4}$$

2. Find the expected values of the uncertain parameters over all N scenarios

$$\bar{\boldsymbol{\xi}}_N := \frac{1}{N} \sum_{n=1}^{N} \xi_n \tag{3.5}$$

3. Solve the EV problem with deterministic parameters and register the EVS

$$EVS := \left\{ x : \operatorname*{argmin}_{x \in X, y \in Y} f(x, y; \bar{\boldsymbol{\xi}}_N) \right\} \tag{3.6}$$

4. Find EEVS by calculating the average EVS over all scenarios

$$EEVS := \min_{y \in Y} f(y; x, \boldsymbol{\xi}_N) \tag{3.7}$$

5. Calculate VSS as

$$VSS = EEVS - OVRP \tag{3.8}$$

VSS is a simple way of estimating how good the solution from a stochastic program is compared to the solution of its deterministic counterpart.

Lastly, we describe the terms robustness and flexibility, as defined in King and Wallace (2003). When making decisions under uncertainty, we wish to withstand random events, but also accommodate them. In a two-stage model, robustness is in the first-stage decisions and flexibility is in the second-stage decisions. Flexibility lets us act if a system enters a disturbed state, while robustness lowers the probability of ending up in such a phase. Assuming that both robustness and flexibility come at a cost, they do not make sense to introduce in deterministic models. The simple explanation for this is that it will never pay off to invest in insurance for contingencies that are assumed to never occur. Deterministic models are always option free.

3.4.2 Scenario Generation

A challenge with stochastic models is to establish good scenario trees. The goal is that the scenario generation procedure should not affect the solution of the problem. Kaut and Wallace (2003) discusses different scenario generation methods and their quality. The seven proposed methods are very briefly described in Table 3.3.

Table 3.3: Short description of the seven methods for scenario generation proposed by Kaut and Wallace (2003)

Scenario Generation Method	**Description**
Conditional sampling	Sample marginals in a vector from only one univariate random variable and evolve the process downwards in the tree
Sampling from specific marginals and correlations	Sample marginals in a vector from different distributions and possibly add the correlation between them
Moment matching	Use statistical properties of the marginals in a vector to construct discrete distributions that satisfy those properties
Path-based	Generate complete paths first and use clustering algorithms to construct a scenario tree from them in a second step
"Optimal discretization"	Approximate a stochastic process (scenario tree) that minimizes an error in the objective function
Scenario reduction	Reduce the size of an existing scenario tree
Internal sampling	Generate a scenario tree during the solution process

Conditional sampling is the most common method for scenario generation. Here, several values are sampled from a stochastic process for every node of the scenario tree. This is done either by sampling directly from the distribution or by evolving the process downwards in the scenario tree by sampling conditional on realizations in parent nodes. In a two-stage process, it does not make sense to evolve the process, since there is only one time step. The norm is to sample from univariate random variables, and when sampling from a random vector, every marginal is sampled separately before combining samples afterward. Kaut and Wallace (2003) discusses the problems with sampling random vectors, since scenario trees grow exponentially with the size of the vector. If n samples are drawn from each of k marginals in a random vector, the result is n^k different scenarios. They also propose several methods to improve a sampling algorithm, however this is outside the scope of this thesis.

According to Kaut and Wallace (2003), there are at least two minimal requirements that a scenario generation method must satisfy:

1. Stability: If one generates several scenario trees with the same input, one should get the same optimal objective value when testing the model with these trees

2. No bias: The scenario tree should not introduce any bias compared to the true distribution

Stability tests for scenario trees
The second requirement stated above can be difficult to confirm, since the true solution of a stochastic problem is rarely known. However, the first requirement can, to some degree, be tested. Kaut and Wallace (2003) introduces two measures to evaluate the stability of a scenario generation method, namely in-sample stability and out-of-sample stability. First, let us denote the scenario tree of the true stochastic process $\tilde{\boldsymbol{\xi}}$ and our approximated discrete process $\breve{\boldsymbol{\xi}}$. If we generate M scenario trees,$\{\breve{\boldsymbol{\xi}}^1, ...\breve{\boldsymbol{\xi}}^M\}$, and find optimal solutions, $\{x_1^*, ..., x_M^*\}$ with objective values $\{f(x_1^*; \breve{\boldsymbol{\xi}}^1), ..., f(x_M^*; \breve{\boldsymbol{\xi}}^M)\}$, then in-sample stability is obtained if

$$f(x_m^*; \breve{\boldsymbol{\xi}}^m) \approx f(x_k^*; \breve{\boldsymbol{\xi}}^k) \qquad \forall \qquad m, k \in 1...M$$

and out-of-sample stability if

$$f(x_m^*; \tilde{\boldsymbol{\xi}}) \approx f(x_k^*; \tilde{\boldsymbol{\xi}}) \qquad \forall \qquad m, k \in 1...M$$

For the in-sample stability, the problem only needs to be solved for the generated scenario trees. However, testing for out-of-sample stability requires evaluating the first-stage solutions on the true distribution, $\tilde{\boldsymbol{\xi}}$. The problem with this is that full knowledge of $\tilde{\boldsymbol{\xi}}$ is rare, and even with full knowledge, it might not be straight forward to evaluate $f(x; \tilde{\boldsymbol{\xi}})$.

There is a practical difference between the two measures. Having out-of-sample stability means the real performance of a solution x_m^* is stable and does not depend on the scenario tree $\breve{\boldsymbol{\xi}}^m$ used for solving the optimization problem. However, if there is not in-sample stability, we can not determine how good the obtained solutions are in reality. Lacking out-of-sample stability is arguably more problematic than lacking in-sample stability, since the performance of the solution depends on what scenario tree is used. It should be noted that it is possible to lack in-sample stability in the objective value, but still have in-sample stability in the solution. In-sample stability in the solution will naturally cause out-of-sample stability, as the solutions are similar. In conclusion, if observing in-sample instability in the objective, one should look at the solution as well. Kaut and Wallace (2003) comments that for most practical applications, there will be either both in- and out-of-sample stability or none of them. As a result, an in-sample stability test is often sufficient, but an out-of-sample test should be performed if there is a practical way of doing so.

Kaut and Wallace (2003) suggests three ways of performing an out-of-sample test:

1. Monte Carlo simulation methods (if the true stochastic process $\tilde{\boldsymbol{\xi}}$ is known)
2. Backtesting with historical data (testing how well solutions x_m^* would have performed in the past)
3. Generating a large tree from a scenario generation method believed to be stable and testing how solutions x_m^* perform on this

In this thesis, the true stochastic process is assumed to be known, and Monte Carlo simulation will then be the obvious choice (Kaut and Wallace, 2003).

Monte Carlo simulation
Monte Carlo simulation is a type of simulation that relies on repeated random sampling for statistical analysis (Raychaudhuri, 2008). The basic idea of Monte Carlo simulation is described in the steps that follow. First, samples of a random variable are drawn from a distribution, or from a sample space which is called bootstrapping. Next, the samples are evaluated under some mathematical conditions and output is generated. When this process is repeated many times, statistical analysis can be performed on the output, either by treating the samples as an empirical distribution or by fitting them to a known probability distribution. The precision of the analysis increases with the number of samples.

Shapiro and Nemirovski (2005) argues that Monte Carlo sampling techniques are reasonably efficient for solving two-stage stochastic problems with recourse, provided certain conditions are met:

1. The solution space X is fixed and feasible regardless of the realization of the uncertain parameter vector ξ
2. For all $x \in X$ and $\xi \in \Xi$, the objective function $f(x, \xi)$ is real, meaning the problem is feasible and bounded
3. The stochastic program can be solved efficiently by a deterministic algorithm, provided the number of scenarios is not "too large"

When dealing with linear stochastic two-stage models with relatively complete recourse, which we will do in this thesis, all three conditions are met (Shapiro and Nemirovski, 2005).

3.4.3 Sample Average Approximation

Sample average approximation (SAA) is an approach based on Monte Carlo simulation to solve stochastic optimization problems. It is a technique where one can estimate the objective function value from random samples of scenarios.

The motivation behind SAA is to find good candidate solutions and good estimates of a stochastic program, without introducing the large number of scenarios needed to represent a stochastic process' distribution. Increasing the size of the scenario tree will increase the size of the problem and therefore also its solution time. By how much is, however, problem-specific. Consequently, the SAA method lets us keep the problem at a manageable size, while still producing good solutions. This is done by resolving the model an appropriate number of times with different random samples of the process' distribution.

Given a sample of N scenarios from the random variable's probability distribution, $\mathbb{E}[Q(x, \xi(\omega)]$ from (3.1) can be approximated by $\frac{1}{N}\sum_{n=1}^{N} Q(x, \breve{\xi}_n)$ where $\breve{\xi}_n = \breve{\xi}(\omega_n)$. Then we can approximate the true problem in (3.2) by solving

$$z_N = \min_{x \in X} \hat{f}(x) \tag{3.9}$$

$$\hat{f}(x) := c^T x + \frac{1}{N}\sum_{n=1}^{N} Q(x, \breve{\xi}_n) \tag{3.10}$$

Problem (3.9) is called the sample average approximation problem. The optimal value z_N and corresponding solution $\hat{x}_N$ give an estimate of the true stochastic problem in (3.1) when we include a sample of N scenarios in the calculation. By the Law of Large Numbers $\hat{f}(x) \to f(x)$ when $N \to \infty$, however this convergence is at a rate of $O_p(N^{-1/2})$ by the Central Limit Theorem. This states that for a sufficiently large sample (sampled with replacement), the distribution of the sample will be approximately normal.

The SAA method is used to calculate some well-known statistical measures of the solution. It proceeds by solving the SAA problem repeatedly for M independently generated scenario trees, each including N scenarios. Optimal objective values $z_N^1, z_N^2, ...z_N^M$ and corresponding solutions $\hat{x}_N^1, \hat{x}_N^2, ...\hat{x}_N^M$ are obtained, and then

$$\bar{z}_N = \frac{1}{M}\sum_{m=1}^{M} z_N^m \tag{3.11}$$

$$\hat{\sigma}^2_{\bar{z}_N} = \frac{1}{(M-1)M}\sum_{m=1}^{M}(z_N^m - \bar{z}_N)^2 \tag{3.12}$$

denote the average and empirical variance of the optimal objective value, respectively. We can then estimate the true objective value of our solutions, $f(\hat{x})$, by generating a new large scenario tree of size $N' \gg N$ and testing our M optimal solutions $\hat{x}_N^1, \hat{x}_N^2, ...\hat{x}_N^M$ on this tree. The estimated true performance and empirical variance of a solution then becomes

$$\hat{z}_{N'}^m = c^T \hat{x}_N^m + \frac{1}{N'} \sum_{n=1}^{N'} Q(\hat{x}_N^m, \xi_n) \tag{3.13}$$

and

$$\hat{\sigma}_{N'}^{m2} = \frac{1}{N'(N'-1)} \sum_{n=1}^{N'} \left(c^T \hat{x}_N^m + Q(\hat{x}_N^m, \xi_n) - \hat{z}_{N'}^m\right)^2 \tag{3.14}$$

The measures calculated above can now be used to estimate the optimality gap of a candidate solution $\hat{x}_N^m$ as $\hat{z}_{N'}^m - \bar{z}_N$ with an estimated variance of $\hat{\sigma}_{\bar{z}_N}^2 + \hat{\sigma}_{N'}^{m2}$. There are, however, different post-processing rules for deciding which of the M candidate solutions $\hat{x}$ to choose. Such rules could include choosing the solution yielding the smallest optimality gap. A step-by-step guide that summarizes the method above is presented below (Ahmed and Shapiro, 2002).

1. Choose the sizes of N,M and N' based on problem-specific characteristics
2. For each $m \in 1, 2, ...M$ do
   (a) Generate a random scenario tree $\breve{\boldsymbol{\xi}}^m$ of size N
   (b) Solve the SAA problem $\min_{x \in X} \hat{f}(x)$
   (c) Generate an independent scenario tree of size $N' \gg N$ and calculate $\hat{z}_{N'}^m$ and $\hat{\sigma}_{N'}^{m2}$ by (3.13) and (3.14)
3. Calculate $\bar{z}_N$ and $\hat{\sigma}_{\bar{z}_N}$ by equation (3.11) and (3.12)
4. Estimate the optimality gap $\hat{z}_{N'}^m - \bar{z}_N$ ((3.13) - (3.11)) and its corresponding variance $\hat{\sigma}_{\bar{z}_N}^2 + \hat{\sigma}_{N'}^{m2}$ ((3.12)+(3.14)) for every solution $\hat{x}_N^m$ and choose the preferred solution according to some rule

3.4.4 Solution Methods for Stochastic Optimization Problems

Optimization problems can be solved either through exact or heuristic solution methods. Exact solution methods find the optimal solution, but are often too time-consuming for complex optimization problems (Bianchi et al., 2008). Heuristic solution methods do not guarantee finding the optimal solution, but they do, however, often find sufficiently good solutions (Bianchi et al., 2008). Furthermore, heuristic solution methods are typically less time-consuming than exact ones, at least for larger instances of complex optimization problems. This section focuses mainly on stochastic optimization problems and ways in which to solve them, depending on their problem size and complexity.

One way of solving optimization problems is through decomposition methods. This is a way to tackle large-scale optimization problems that cannot be handled by MIP solvers (Laesanklang and Landa-Silva, 2017). The idea is to exploit the problem structure. Decomposition methods divide the problem into subproblems that are easier to solve than the original problem, also called the master problem (Laesanklang and Landa-Silva, 2017). Each subproblem is then solved separately, before their solutions are put back together to form a solution to the master problem (Laesanklang and Landa-Silva, 2017; Andersson, 2021). This can be done in two ways, either through constraint decomposition or variable decomposition (Laesanklang and Landa-Silva, 2017). In the former, constraints or cuts are added to the problem, thereby narrowing the feasible region and improving the bounds. In the latter, the problem is solved in two stages. In the first-stage, the values for a set of variables are fixed. In the second-stage, the optimal solution for the remaining variables is found, given the variable values set in the first-stage (Laesanklang and Landa-Silva, 2017). Bender's decomposition is one such decomposition method (Laesanklang and Landa-Silva, 2017), and it is also one of the most commonly used decomposition algorithms (Guo et al., 2021). Like other decomposition methods, it can be implemented either as an exact or a heuristic solution method (Laesanklang and Landa-Silva, 2017; Guo et al., 2021). When its subproblems contain integer variables, the classical Bender's decomposition cannot be applied (Guo et al., 2021). Modified variants, such as logic-based Bender's method (LBBD), have therefore been developed.

LBBD is an exact technique that has become increasingly popular over the last decade for solving combinatorial optimization problems (Roshanaei and Naderi, 2021). The technique has shown promising performances when applied to healthcare optimization problems, as long as the problem can be decomposed into an assignment problem in the first-stage and a packing, routing or scheduling problem in the second-stage (Roshanaei and Naderi, 2021). Roshanaei and Naderi (2021) applies LBBD to its deterministic integrated operating room planning and scheduling problem, which proves to outperform the existing Branch-Price-Cut algorithm. Guo et al. (2021) applies LBBD to a stochastic operating room scheduling problem. The surgery durations are modeled stochastically in order to derive more robust schedules, and the problem is modeled in both two and three stages. The two-stage LBBD reduces both the solution time and the optimality gap when compared to their MIP in a commercial solver.

As previously mentioned, exact solution methods often struggle to perform well for larger instances of stochastic optimization problems, and even small instances may require a lot of computational effort due to the problem complexity. Stochastic multi-stage models further increase the problem size drastically by introducing scenarios. In contrast to exact solution methods, metaheuristics are often able to find good solutions to realistically sized problem instances, using far less computational time (Bianchi et al., 2008). It is possible, and often convenient, to combine metaheuristics with exact methods, as the combination may help in reaching solutions of higher quality in a reasonable amount of time (Juan et al., 2021). This is called matheuristics, which have become popular over the last decade (Juan et al., 2021).

Two central concepts in metaheuristics are exploration and exploitation (Blum and Roli, 2003). Exploration describes the global search for promising solutions in the entire feasible region, whereas exploitation describes the local search for improved solutions in promising subregions (Andradóttir and Prudius, 2009). The two forces are both contrary and complementary to each other, and they determine the metaheuristic's behavior (Blum and Roli, 2003). One must find a balance between the two forces in order to efficiently explore the search space (Bianchi et al., 2008).

With the increasing popularity of metaheuristics, there are many to choose from when solving stochastic optimization problems. Examples of metaheuristics that have proven to be successful when applied to stochastic optimization problems are ant colony optimization (ACO), tabu search (TS), genetic algorithms (GA) and simulated annealing (SA) (Bianchi et al., 2008). ACO is according to Bianchi et al. (2008) one of the most successful nature-inspired metaheuristics, mimicking the foraging techniques of real ants (Blum and Roli, 2003). It is a probabilistic method that in essence is about finding good paths through graphs. TS relies on memory, as it uses information from the past to make better choices in the search process (Juan et al., 2021). Instead of stopping at a local optimum, it explores by choosing solutions that worsen the objective. To prevent going back and forth between the same improving and non-improving solutions, the heuristic keeps a tabu list. This is a list of the last solutions visited, making these temporarily illegal. In GA, a population of solutions evolves over several generations through the operators selection, crossover and mutation (Juan et al., 2021). Selection comes first, evaluating the current solutions through a fitness criteria. Based on this, only some of them are chosen to be part of the reproduction process. Second, new solutions are created through the crossover and mutation operators, thereby creating new offspring and diversifying the population. Lastly, the new generation fully or partly replaces the previously existing population. SA is often said to be the oldest of the metaheuristics (Blum and Roli, 2003). It builds on the principles of local search heuristics, but with modifications in order to explore the entire solution space (Blum and Roli, 2003). In each iteration, a neighboring solution to the current solution is found (Blum and Roli, 2003). If the neighbor has a better objective value, it is chosen as the new solution. If it has a worse objective value, it may still be chosen, with a probability depending on a control parameter called the temperature (Lundgren et al., 2010). In order for the method to converge, the temperature decreases over time, thereby reducing the probability of choosing worse solutions (Lundgren et al., 2010). The algorithm puts emphasis on exploration and global search at the start, gradually moving towards exploitation and local search towards the end.

Unlike SA, similar methods such as balanced explorative and exploitative search with estimation (BEESE) emphasize both exploration and exploitation throughout the entire search (Andradóttir and Prudius, 2009). At each iteration, randomized BEESE (R-BEESE) chooses two solutions, one from a global, and therefore explorative neighborhood, and one from a local, and therefore exploitative, neighborhood. The explorative solution is chosen with probability p and the exploitative solution is chosen with probability 1-p (Andradóttir and Prudius, 2009). Whereas R-BEESE randomly chooses the neighborhood in which it finds the next solution, adaptive BEESE (A-BEESE) adaptively alternates between picking from the explorative and the exploitative neighborhood, aiming to use a suitable neighborhood at each stage (Andradóttir and Prudius, 2009). Once again, we observe the centrality of balancing exploration versus exploitation in metaheuristics. It defines the different methods and largely determines their performance.

Out of the solution methods mentioned, SA is particularly popular due to its simplicity, runtime and solution quality (Spratt and Kozan, 2016). It tends to outperform other relevant heuristics like TS and GA (Spratt and Kozan, 2016), and is one of the most commonly used heuristics for scheduling problems (Abdeljaouad et al., 2020).

Chapter 4

Problem Description

The problem described in this section involves creating a modified MSS for a surgical clinic. This is also called a modified block schedule, as described in Section 3.1.1. Due to the problem being identical to the one in Asplin et al. (2021), the only changes made to this section are some clarifications.

Time slots are the components of an MSS. A slot is defined as a full day of access to one of the elective ORs belonging to the surgical clinic. Out of the total number of slots available during a cycle, a fraction of these must be fixed to a surgical specialty, while the remaining are considered flexible slots. When a specialty is assigned a fixed slot, they will have access to the corresponding OR on that day in every cycle. Flexible slots, however, may be assigned to a specialty on a date closer to the actual execution of that week's schedule. The benefit of flexible slots is that they can contribute to accommodating fluctuations in demand for different surgical procedures. Flexible slots are what differentiates a regular MSS from a modified MSS. The term MSS hereafter refers to a modified MSS.

The decisions of this problem are made at different points in time, as illustrated in Figure 4.1. Deciding which slots should be fixed and what specialties they should be assigned to is done once, typically just before the start of a new calendar year. This forms the cyclic MSS that is typically repeated for the following six to twelve months, referred to as the planning horizon. The assignment of specialties to flexible slots and subsequent packing of surgeries into the schedule is, however, performed closer to the execution of each planning period. It is best to make these decisions as close to the planning period as possible, due to the increased accuracy of the accumulated information on demand. Patients at the Surgical Clinic at St. Olav are informed of the timing of their surgical procedure one to three weeks prior to its execution. Therefore, we propose a two-week time gap between the assignment of specialties to flexible slots and the start of the planning period. The length of a planning period can be chosen arbitrarily, but it must correspond to the length of one or more cycles. For example, the planning periods in Figure 4.1 consist of two cycles each.

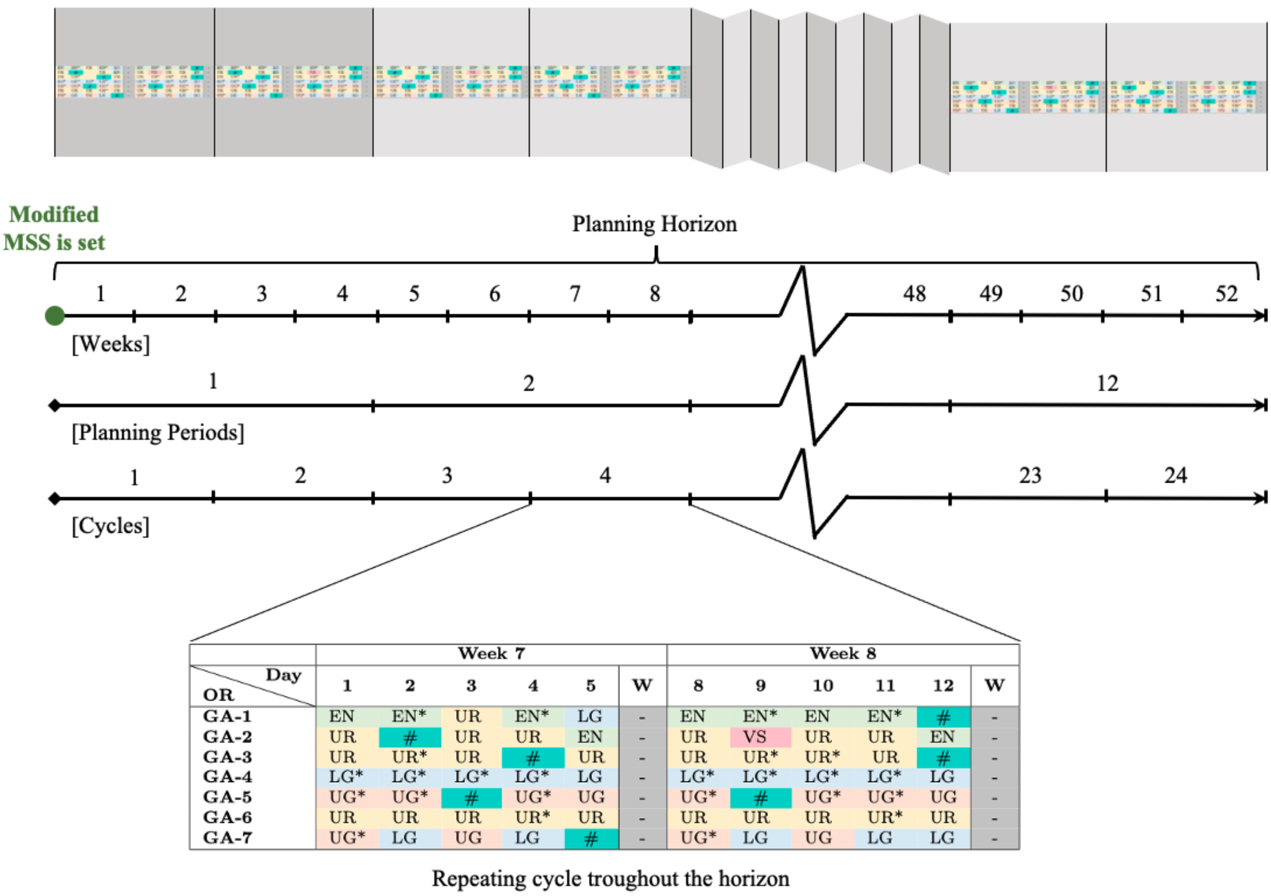
The different specialties have different needs in terms of facilities and equipment, and should only be assigned to ORs that fit their needs. Another limiting resource of the surgical clinic is its staff. The clinic comprises a set of surgical specialties, each with a predefined number of surgical teams available for each day in a cycle. The number of surgical teams available limits the number of slots a specialty can be assigned on a day. To each surgical specialty belongs a set of surgery groups. A surgery group consists of patients with equal surgery duration and LOS after surgery, and may only be operated by surgeons within the specialty that their group belongs to.

A third limiting resource is the capacity in the MC and IC wards downstream of the ORs. Each surgery group is associated with one MC ward and one or more IC wards. The individual wards have a predefined number of beds available for each day of a cycle, and the number of beds occupied by patients may not exceed this capacity. If a patient stays in an IC ward after surgery, they subsequently spend at least one night in an MC ward before being checked out. If a patient goes directly to an MC ward after surgery, they will not move to an IC ward afterward.

The sum of surgery durations for patients scheduled for surgery in a slot, including the time it takes to clean the OR after surgery, should not exceed the time available in the slot. By default, all ORs open and close at the same time every day that they are open, and thus a slot represents a default number of available minutes. However, a slot can be extended by postponing the closing of the OR. The maximum number of slots a specialty can extend during a cycle is predefined. Lastly, only fixed slots are eligible for extension.

Uncertainty is present in the demand for different surgery groups. The schedule should therefore accommodate fluctuations in demand. This requires assigning fixed slots in such a way that the schedule is robust to different levels of demand, taking into account that the remaining flexible slots can be used to respond to fluctuations in demand. In reality, there is also uncertainty in the surgery durations and LOS for the different surgery groups, but this is considered deterministic.

The objective in this problem is to minimize the expected amount of unmet demand at the end of a planning period. We assume that demand corresponds to the patients that are in line for surgery at the point in time where the schedule for a planning period is packed with patients from the different surgery groups. Accordingly, the expected unmet demand corresponds to the total surgery time of the patients in line for surgery that do not fit when packing the schedule. For example, if all patients except two are scheduled for surgery, the unmet demand is the sum of these two patients' surgery times.



OR \ Day	Week 7: 1	2	3	4	5	W	Week 8: 8	9	10	11	12	W
GA-1	EN	EN*	UR	EN*	LG	-	EN	EN*	EN	EN*	#	-
GA-2	UR	#	UR	UR	EN	-	UR	VS	UR	UR	EN	-
GA-3	UR	UR*	UR	#	UR	-	UR	UR*	UR*	UR	#	-
GA-4	LG*	LG*	LG*	LG*	LG	-	LG*	LG*	LG*	LG*	LG	-
GA-5	UG*	UG*	#	UG*	UG	-	UG*	#	UG*	UG*	UG	-
GA-6	UR	UR	UR	UR*	UR	-	UR	UR	UR	UR*	UR	-
GA-7	UG*	LG	UG	LG	#	-	UG*	LG	UG	LG	LG	-

Figure 4.1: An illustration of the relation between the planning horizon, the planning period and the MSS cycles. Letters inside a slot mean that the slot is assigned to a specialty as a fixed slot, and "*" signifies that the slot is extended. "#" means that a slot is flexible.

Chapter 5

Mathematical Model

In this section, we develop a two-stage stochastic model to tackle the problem described in Chapter 4. The model creates a cyclic MSS for a surgical clinic that takes into account uncertainty in demand. We also cater to fluctuations in demand by allowing for some flexibility in the MSS. Section 5.1 is the same as in Asplin et al. (2021). Section 5.2 and Section 5.3 are mostly identical to before, with some mathematical modifications. In addition to the MIP formulated in this section, we have formulated an alternative MIP inspired by the cutting stock problem. This alternative formulation (CSMIP) mostly performs inferior to the MIP formulated in this section, and will therefore not be introduced in the body of this thesis. However, the formulation of the CSMIP can be found in Appendix B.

5.1 Model Assumptions

In this thesis, we make some simplifications in order to reduce the complexity of the problem. Since the introduction of stochastic parameters and additional decision stages generally increase the complexity of a problem, several assumptions with regard to the problem setting are made. Our model assumptions are threefold. First, we introduce assumptions made with regards to the surgical clinic in Section 5.1.1, before proceeding to elaborate on those of the clinic's patients in Section 5.1.2. To conclude the model assumptions, we go into some details on how the modeling of stochastic parameters is handled in Section 5.1.3. In this last part, we also argue why a two-stage model is appropriate for our problem.

5.1.1 The Surgical Clinic

First and foremost, emergency patients are not taken into account. While emergency patients are a part of the surgical clinic's responsibility, the vast majority of the clinic's resources are used for elective patients.

Surgical teams within the same specialty are considered homogeneous. This means to say that differences in experience and performance of the staff in each surgical team are not taken into account, and have no impact on the duration of surgeries. The time it takes to clean an OR and prepare it for the next surgery is assumed to be constant. It is also assumed that after the last surgery of the day the ORs are cleaned and prepared for the first operation the following day.

At the bed wards downstream of the ORs, simplifications are made in order to handle the uncertainty in different patients' LOS. Most importantly, the numbers of occupied beds in wards are modeled as expected values. Obviously, a patient either occupies a bed or not in reality. However, since a patient's LOS is uncertain, we consider the use of expected occupancy to be more fitting than forcing the LOS of patients to be a predefined integer. Going into a planning period, it is assumed that some of the beds at the wards are occupied by patients operated prior to the planning period. For each ward and day in the planning period, the leftover bed occupancy from prior periods is modeled as a constant. This mechanism is described in detail in Section 5.3.2.

5.1.2 The Patients

Patients within the same surgery group are considered homogeneous. This implies that they have the same surgery duration, as well as probabilities of staying in their affiliated wards after surgery. Furthermore, it is assumed that for every surgery group the probability of a patient from this group staying in an affiliated ward on a given night after their surgery is known and unconditional. To exemplify how these probabilities translate to expected bed occupancy, we consider an example where a patient is operated on a Monday. If the patients in the surgery group that this patient belongs to have a probability of 0.5 of staying in an affiliated ward one day after surgery, then this patient will occupy 0.5 beds in that ward on the night between Monday and Tuesday.

As mentioned in Chapter 4, patients who stay in IC wards post-surgery will always move to an MC ward afterward. However, this dependency is challenging to model. Since we do not assume whether a patient stays in a ward or not, but merely count the probability of the patient staying in a ward on a night after their surgery, we do not know which patients actually end up staying in the IC wards. We circumvent this obstacle by simply assuming that patients may have positive probabilities of both staying at an IC ward and an MC ward, meaning that they contribute to occupancy at both wards simultaneously.

5.1.3 Modeling Stochastic Parameters

The idea of a two-stage stochastic model is to take into account the uncertainty of some parameters by making decisions based on several possible realizations of them. The mathematical model formulated in Section 5.3 includes a set of scenarios. Each of these scenarios contains an integer value demand for each surgery group. The generation of scenario trees is described in Section 7.1.3.

Modeling in two stages is obviously a simplification of reality in our case. First of all, the patients are not referred for surgery in batches once a month. In fact, they probably arrive in line for surgery quite continually in time. However, modeling continuous arrivals is very challenging, and simplifying the arrival of new information in stages is a common way to tackle this. Considering surgical clinics usually pack their schedules with surgeries for one or more weeks at a time, the aggregation to stages is not that unreasonable either.

Even though modeling with stages might be an appropriate approach, it might not be intuitive why two stages are more suitable than many. Since an MSS in practice is used for several consecutive cycles and planning periods, a multi-stage model might seem more suitable at first glance. However, the main implication of one planning period on the next, is the patients that will occupy beds in wards in the next planning period. Also, multi-stage models are notoriously complex to solve as they grow in size. This motivates us to reduce the problem into two stages, and by addressing the dependencies between planning periods that were just mentioned, this is arguably not unreasonable. Once again, we detail how bed occupancy across different planning periods is handled in Section 5.3.2.

The information structure of the two-stage model is illustrated in Figure 5.1. In the first-stage, we only know the distribution for different surgery groups' demand, while the actual demand is known in the second-stage. The assignment of fixed slots and extensions are done in the first-stage, while the assignment of flexible slots and packing of surgeries are second-stage decisions. Figure 5.2 illustrates how the continual arrivals of patients translate to aggregated demand used in the second-stage decisions. The main output of our model is the MSS with assigned fixed slots and unassigned flexible slots. Figure 5.2 describes how such an MSS could be used in reality by assigning flexible slots and packing the schedule with patients at regular intervals. In this example, we have assumed planning periods of four weeks.

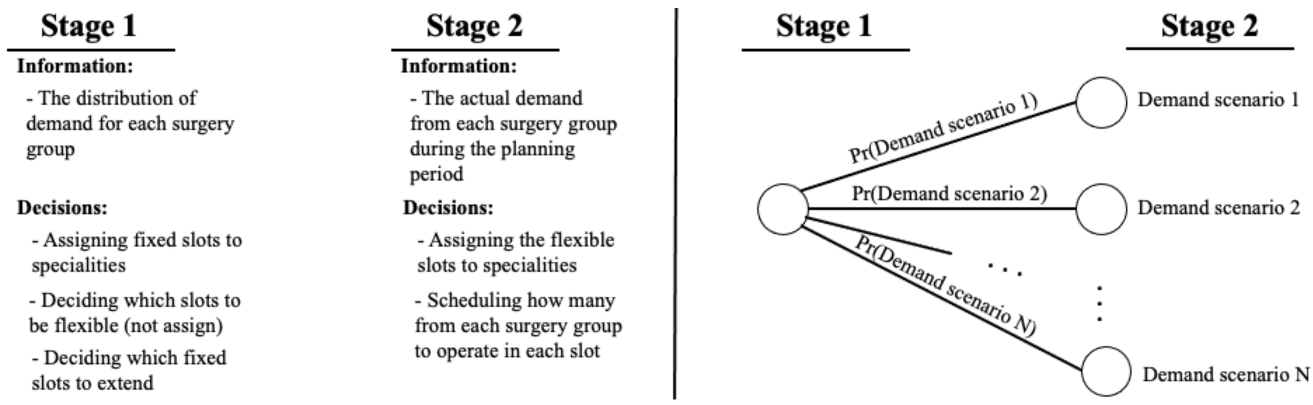


Figure 5.1: The figure gives an overview of the information and decision stages alongside a generic scenario tree for surgery group demands.

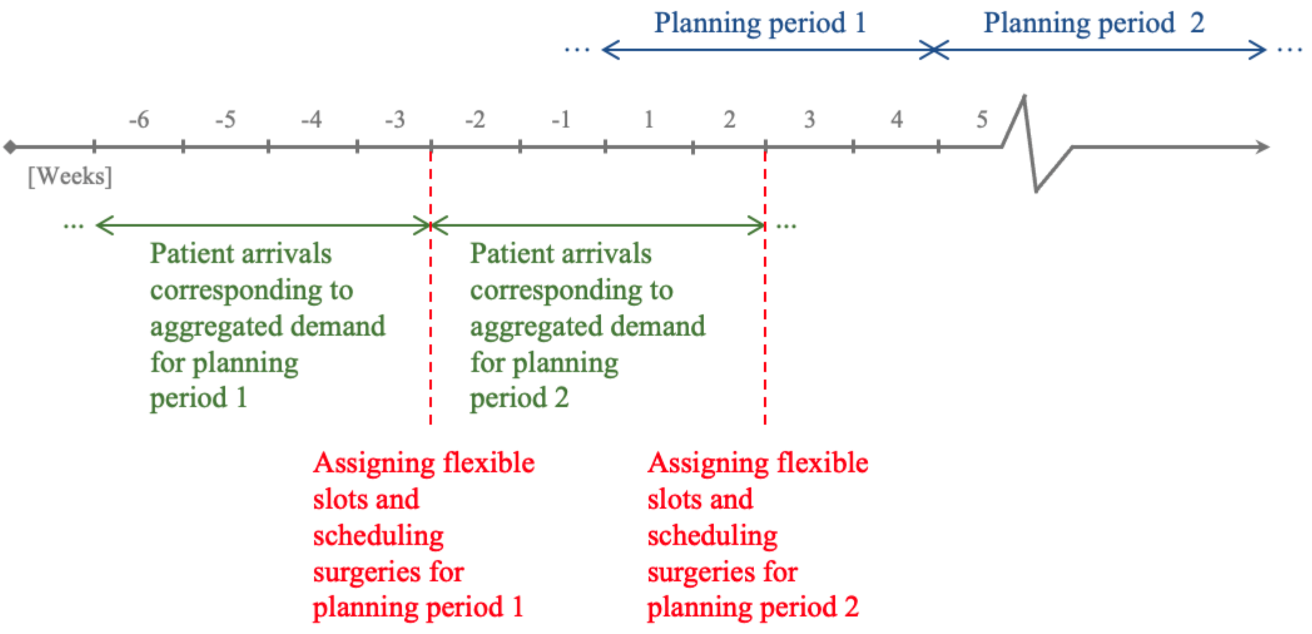


Figure 5.2: The figure exemplifies how arrival of new information is aggregated, and the timing of patient arrivals relative to each planning period.

5.2 Notation

Sets and indices

$\mathcal{W}$	Wards, indexed w
$\mathcal{S}$	Specialties, indexed s
$\mathcal{S}_r^{\mathcal{R}}$	Specialties suitable for OR r, indexed s
$\mathcal{G}$	Surgery groups, indexed g
$\mathcal{G}_w^{\mathcal{W}}$	Surgery groups that can receive postoperative care at ward w, indexed g
$\mathcal{G}_s^{\mathcal{S}}$	Surgery groups that can receive treatment from specialty s, indexed g
$\mathcal{R}$	ORs, indexed r
$\mathcal{R}_s^{\mathcal{S}}$	ORs suitable for specialty s, indexed r
$\mathcal{R}_g^{\mathcal{G}}$	ORs suitable for group g, indexed r
$\mathcal{D}$	Days in planning period, indexed d
$\mathcal{C}$	Scenarios, indexed c

Parameters

Π_c	Probability of scenario c occurring
C_g	Unit cost of not meeting the demand of surgery group g
T^C	Cleaning time post-surgery
L_g^{SD}	Surgery duration of a patient in surgery group g
F	Maximum percentage of flexible number of slots
N_d	Total number of available ORs on day d
U_s^X	Maximum number of times a specialty s may extend its opening hours during a cycle
I	Number of cycles in the planning horizon
K_{sd}	Number of surgical teams from specialty s available on day d
H_d	Default amount of time available in a slot if it is assigned at day d
E	Additional time available if a slot's opening hours are extended
Q_{gc}	Number of patients from surgery group g in line for surgery in scenario c
P_{gwd}	Probability that a patient from surgery group g occupies a bed in ward w, on the night d days after surgery
J_w	Maximum number of nights a patient may stay in ward w
B_{wd}	Number of available beds at ward w on the night following day d
Y_{wd}	Expected number of occupied beds in ward w on the night following day d in the current planning period, by patients operated in a prior planning period

Variables

First-stage decision variables:

$$\gamma_{srd} \quad \begin{cases} 1 & \text{if specialty } s \text{ is assigned a fixed slot in room } r \text{ on day } d \\ 0 & \text{Otherwise} \end{cases}$$

$$\lambda_{srd} \quad \begin{cases} 1 & \text{if specialty } s \text{ extends opening hours in room } r \text{ on day } d \\ 0 & \text{Otherwise} \end{cases}$$

Second-stage decision variables:

$$\delta_{srdc} \quad \begin{cases} 1 & \text{if specialty } s \text{ is assigned a flexible slot in room } r \text{ on day } d \text{ in scenario } c \\ 0 & \text{Otherwise} \end{cases}$$

x_{grdc} Number of patients from surgery group g operated in room r on day d in scenario c

Auxiliary variables:

a_{gc} Number of patients from surgery group g waiting in line for surgery, but not scheduled for surgery in scenario c

5.3 Model Formulation

The above-defined sets and indices, parameters and variables are used to make up a stochastic mixed integer program (MIP) below. After introducing the objective, we present the constraints in an order that follows the logic of a two-stage stochastic model.

5.3.1 Objective Function

$$\min \sum_{c \in \mathcal{C}} \Pi_c \sum_{g \in \mathcal{G}} C_g a_{gc} \tag{5.1}$$

The objective function, (5.1), minimizes the expected cost of all patients in line for surgery, but not scheduled for surgery. The scenario probabilities, Π_c, sum to 1, and thus the objective function can be considered an expected value.

5.3.2 Constraints

$$\sum_{s \in \mathcal{S}} \sum_{r \in \mathcal{R}_s^{\mathcal{S}}} \sum_{d \in \mathcal{D}} \gamma_{srd} \geq \left\lceil (1 - F) \sum_{d \in \mathcal{D}} N_d \right\rceil \tag{5.2}$$

Constraint (5.2) ensures that a minimum percentage of the total number of slots in the planning period is fixed, and thus implicitly sets an upper bound for the number of flexible slots. Since the LHS sums over binary variables, the sum will always be integer. Consequently, the RHS is rounded up to the nearest integer to make the constraint tighter. The interpretation of this constraint is that the number of fixed slots may range from the lower bound given by the RHS to 100%. This constraint is likely to be binding since the objective function provides no incentive to fixate slots, while flexible slots can contribute to meeting more demand.

$$\lambda_{srd} \leq \gamma_{srd} \qquad s \in \mathcal{S}, r \in \mathcal{R}_s^{\mathcal{S}}, d \in \mathcal{D} \tag{5.3}$$

$$\sum_{r \in \mathcal{R}_s^{\mathcal{S}}} \sum_{d \in \mathcal{D}} \lambda_{srd} \leq U_s^{\mathrm{X}} \qquad s \in \mathcal{S} \tag{5.4}$$

Constraints (5.3) and (5.4) regulate which and how many slots may be extended. The former enforce that a slot may only be extended if it is assigned to a specialty as a fixed slot, implying that flexible slots are not to be extended. The latter ensure that each specialty does not extend more slots during the planning period than they are allowed. Exceptions from the tendency of (5.2) to be binding can occur when the flexibility parameter, F, allows for fewer fixed slots than the total number of slots that may be extended. Since only fixed slots can be extended, the number of fixed slots is likely determined to be at least as high as the total number of slots that may be extended.

$$\gamma_{srd} = \gamma_{sr(\frac{|\mathcal{D}|}{I}+d)} \qquad s \in \mathcal{S}, r \in \mathcal{R}_s^{\mathcal{S}}, d = 1, ..., |\mathcal{D}| - \frac{|\mathcal{D}|}{I} \tag{5.5}$$

$$\lambda_{srd} = \lambda_{sr(\frac{|\mathcal{D}|}{I}+d)} \qquad s \in \mathcal{S}, r \in \mathcal{R}_s^{\mathcal{S}}, d = 1, ..., |\mathcal{D}| - \frac{|\mathcal{D}|}{I} \tag{5.6}$$

Constraints (5.5) and (5.6) ensure that fixed and extended slots repeat themselves in predefined cycles throughout the planning period. This translates to the fixed slots being equal in all cycles. The calculation of index d on the RHS of both constraints facilitates changes in the input parameters $\mathcal{D}$ and I. The constraints work for all planning period lengths, $|\mathcal{D}|$, and number of cycles, I, so long as the length of a cycle, $\frac{|\mathcal{D}|}{I}$, is an integer.

$$\sum_{s \in \mathcal{S}} \gamma_{srd} + \delta_{srdc} \leq 1 \qquad r \in \mathcal{R}_s^{\mathcal{S}}, d \in \mathcal{D}, c \in \mathcal{C} \tag{5.7}$$

$$\sum_{r \in \mathcal{R}_s^{\mathcal{S}}} \gamma_{srd} + \delta_{srdc} \leq \mathrm{K}_{sd} \qquad s \in \mathcal{S}, d \in \mathcal{D}, c \in \mathcal{C} \tag{5.8}$$

$$\sum_{s \in \mathcal{S}} \sum_{r \in \mathcal{R}_s^{\mathcal{S}}} \gamma_{srd} + \delta_{srdc} \leq \mathrm{N}_d \qquad d \in \mathcal{D}, c \in \mathcal{C} \tag{5.9}$$

Constraints (5.7)-(5.9) tie together the assignment of slots in the first- and second-stage. Constraints (5.7) ensure that no ORs are double-booked by allowing for at most one specialty to be assigned to a slot, either as a fixed or flexible slot. Constraints (5.8) further state that no specialty should be assigned more slots on a day than it has teams available. Finally, constraints (5.9) regulate the total number of ORs that may be assigned on a day, either fixed or flexible. The motivation of this constraint is to enable restricting the OR capacity, while still letting the model make the decision of what ORs to use. Constraints (5.7) - (5.9) are all defined for every scenario, because the second-stage variable, δ_{srdc}, may differ in every scenario. This means to say that flexible slots may be assigned to different specialties in different scenarios.

$$\sum_{g \in \mathcal{G}_s^{\mathcal{S}}} (\mathrm{L}_g + \mathrm{T}^{\mathrm{C}}) x_{grdc} \leq \mathrm{H}_d(\gamma_{srd} + \delta_{srdc}) + \mathrm{E}\lambda_{srd} \qquad s \in \mathcal{S}, r \in \mathcal{R}_s^{\mathcal{S}}, d \in \mathcal{D}, c \in \mathcal{C} \tag{5.10}$$

Constraints (5.10) link the assignment of slots to the scheduling of surgeries. It does so by making sure that the total planned operating and cleaning time in a slot does not exceed the slot's available time, possibly including additional time gained by extending the slot. Another important implication of (5.10) is that patients can only be planned for surgery in a slot assigned to the specialty that their surgery group belongs to. The RHS of the constraints will be zero for all specialties, except for the one that it is assigned to. Consequently, no patients from surgery groups not belonging to that one specialty can be planned for surgery in the slot.

$$\sum_{r \in \mathcal{R}} \sum_{d \in \mathcal{D}} x_{grdc} + a_{gc} = \mathrm{Q}_{gc} \qquad g \in \mathcal{G}, c \in \mathcal{C} \tag{5.11}$$

Constraints (5.11) keep track of the unmet demand of surgery group g in scenario c through the auxiliary variable a_{gc}. This variable is used in the objective function to penalize unmet demand. Since both x_{grdc} and a_{gc} are non-negative integer variables, the constraints above make sure that we cannot plan more patients for surgery than there are patients in line for surgery, in every surgery group in every scenario.

$$\sum_{s \in \mathcal{S}_r^{\mathcal{R}}} \delta_{srdc} \leq \sum_{g \in \mathcal{G}} x_{grdc} \qquad r \in \mathcal{R}, d \in \mathcal{D}, c \in \mathcal{C} \tag{5.12}$$

Constraints (5.12) ensure that a flexible slot is only assigned to a specialty if there is at least one planned operation in that slot. This can potentially cut away many second-stage solutions that are practically identical in scenarios where some flexible slots are redundant.

$$\sum_{g \in \mathcal{G}_w^{\mathcal{W}}} \sum_{r \in \mathcal{R}} \sum_{\delta = \max\{1, d+1-\mathrm{J}_w\}}^{d} \mathrm{P}_{wg(d-\delta+1)} x_{gr\delta c} \leq \mathrm{B}_{wd} - Y_{wd} \qquad w \in \mathcal{W}, d \in \mathcal{D}, c \in \mathcal{C} \tag{5.13}$$

Constraints (5.13) ensure that the expected number of occupied beds in a ward on the night following day d does not exceed the number of beds available. Since the scheduling of surgeries is specific to each scenario, so is the bed occupancy in wards. In order to explain how the constraints work, we consider an instance of the constraints for some ward on a day in a scenario. The LHS sums over all surgery groups that have a positive probability of staying in this ward after surgery and considers that these patients may have been operated on in any OR. The index δ is introduced in order to take into account all patients operated prior to the night following day d, but will only go as far back as the first day of the planning period or $J_w - 1$ days back in time. Since J_w defines the maximum number of nights a patient can stay in a ward after surgery, there is no need to look further back in time than this. Notice that the LHS does not necessarily equate to the total expected number of occupied beds in the ward, since there may be patients operated prior to the start of the planning period that still stay in the ward when the planning period begins. These patients are taken into account by deducting the parameter Y_{wd} from the total number of beds available, B_{wd}, on the RHS.

$$\gamma_{srd} = 0 \qquad s \in \mathcal{S}, r \in \{\mathcal{R} \backslash \mathcal{R}_s^{\mathcal{S}}\}, d \in \mathcal{D} \tag{5.14}$$

$$\delta_{srdc} = 0 \qquad s \in \mathcal{S}, r \in \{\mathcal{R} \backslash \mathcal{R}_s^{\mathcal{S}}\}, d \in \mathcal{D}, c \in \mathcal{C} \tag{5.15}$$

$$x_{grdc} = 0 \qquad g \in \mathcal{G}, r \in \{\mathcal{R} \backslash \mathcal{R}_g^{\mathcal{G}}\}, d \in \mathcal{D}, c \in \mathcal{C} \tag{5.16}$$

Constraints (5.14) and (5.15) ensure that a specialty is not assigned to an OR that is not suitable. This is done for every specialty by forcing the assignment variables to be 0 for all slots affiliated with an OR that is not suitable for the specialty. Furthermore, (5.16) makes sure that patients can not be planned for surgery in ORs that are not suitable for their corresponding surgical specialty.

$$\gamma_{srd}, \lambda_{srd} \in \{0, 1\} \qquad s \in \mathcal{S}, r \in \mathcal{R}, d \in \mathcal{D} \tag{5.17}$$

$$\delta_{srdc} \in \{0, 1\} \qquad s \in \mathcal{S}, r \in \mathcal{R}, d \in \mathcal{D}, c \in \mathcal{C} \tag{5.18}$$

Constraints (5.17) and (5.18) make sure variables for assigning slots are binary. This is necessary to reflect that a slot cannot be split between different specialties, and corresponds to a full day of access to an OR.

$$x_{grdc} \in \mathbb{Z}^+ \qquad g \in \mathcal{G}, r \in \mathcal{R}, d \in \mathcal{D}, c \in \mathcal{C} \tag{5.19}$$

$$a_{gc} \in \mathbb{Z}^+ \qquad g \in \mathcal{G}, c \in \mathcal{C} \tag{5.20}$$

Constraints (5.19) and (5.20) ensure that the given variables are integer and non-negative. This entails that the number of patients from a surgery group planned for surgery in a slot cannot be negative or fractional. Likewise, the auxiliary variable, a_{gc}, must be a non-negative integer to reflect the number of patients from each surgery group that has not been scheduled for surgery.

Chapter 6

Solution Methods

As we reveal in Chapter 8, the MIP from Chapter 5 can find near-optimal solutions for small instances of the MSSP when implemented in a commercial solver. However, instances with flexibility and any considerable amount of scenarios prove challenging to find good solutions for. In this chapter, we propose a simple extension of the MIP, as well as two different search heuristics based on a simulated annealing (SA) framework. Section 6.1 provides an overview of the four different solutions methods, while the remainder of Chapter 6 concerns the two SA heuristics. In Section 6.2 we detail how the SA heuristics search through first-stage solutions. Section 6.3 describes the greedy construction heuristic (GCH) used in the second-stage of one of the SA heuristics.

6.1 Overview of Solution Methods

From here on out, the MIP in Chapter 5 will be referred to as the Full MIP (FMIP) when it is implemented in a commercial solver and solves both the first- and second-stage simultaneously. When the implemented MIP is used to find the optimal second-stage solution for a fixed first-stage solution, we refer to it as the second-stage MIP (2SMIP). Preliminary testing reveals that the primal bounds of solutions found with FMIP can be lowered quite significantly by fixing the best first-stage solution found and evaluating it with the 2SMIP. This is a simple matheuristic, and we will refer to this extension as FMIP-2SMIP.

Chapter 6 primarily concerns the two SA heuristics. Both heuristics share the same first-stage method, but differ in how they evaluate first-stage solutions in the second-stage of the MSSP. The first variant uses the 2SMIP in the second-stage (SA-2SMIP). The other variant uses a GCH to find second-stage solutions, but evaluates the best first-stage solution found at the end of the SA with the 2SMIP (SA-GCH-2SMIP). A top-level flowchart is presented in Figure 6.1, giving an impression of how both of the SA heuristics search through first-stage solutions.

In Table 6.1 the different solution methods are listed with their abbreviated names. The table displays how the solution methods solve the first- and second-stage of the MSSP. The methods range from the entirely exact FMIP to the almost completely heuristic SA-GCH-2SMIP.

Table 6.1: Overview of solution methods proposed and whether they solve the first- and second-stage of the MSSP using exact or heuristic solution methods.

Solution method	**FMIP**	**FMIP-2SMIP**	**SA-2SMIP**	**SA-GCH-2SMIP**
First-stage	Exact	Exact*	Heuristic	Heuristic
Second-stage	Exact	Exact	Exact	Heuristic**

*The first-stage solution is fixed before it has been proven optimal, and the first-stage is therefore not solved exactly.
**During the evaluation of candidate first-stage solutions, the second-stage is solved heuristically using GCH. The best candidate first-stage solution found at termination of the SA algorithm is evaluated using 2SMIP (exact method).

The FMIP is versatile in the sense that it can be used to solve the first- and second-stage simultaneously, but this can be very time-consuming. While FMIP-2SMIP will not find any different first-stage solutions than the FMIP, it evaluates the first-stage solutions to optimality. The SA heuristics become necessary for instances where the FMIP and FMIP-2SMIP can no longer produce good first-stage solutions.

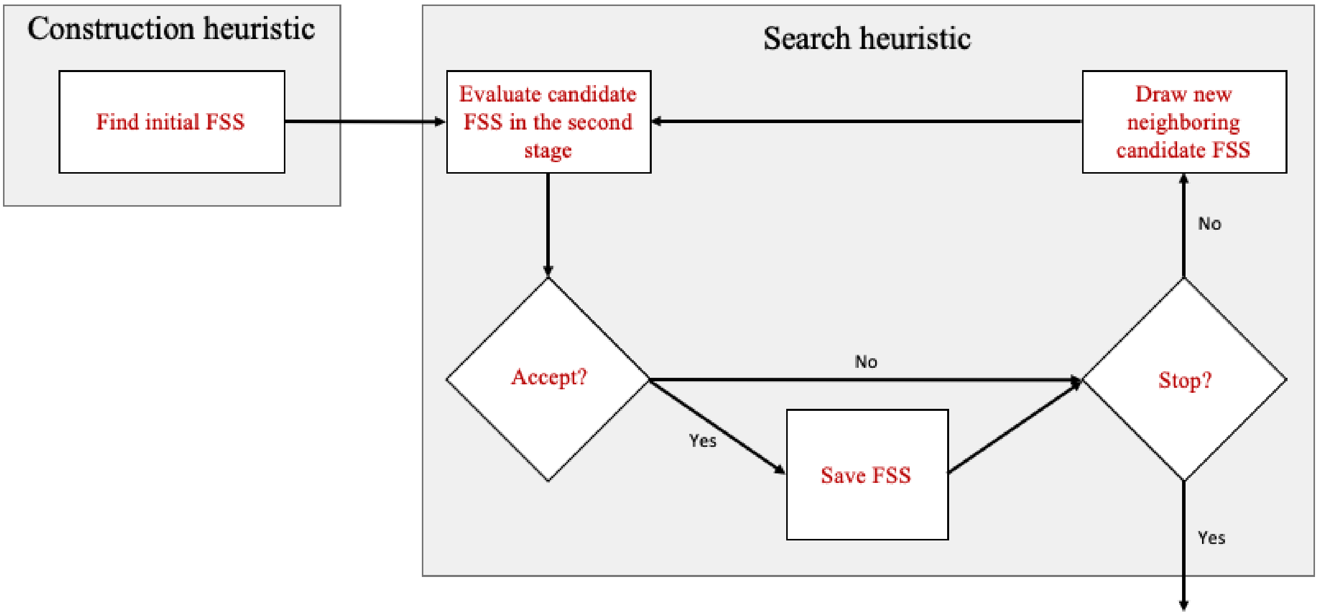


Figure 6.1: Top-level flowchart of the two SA heuristics' logic.

6.2 First-Stage Simulated Annealing Heuristic

Algorithm 1 provides a pseudocode for the first-stage SA heuristic. In Section 6.2, the word "solution" refers to a first-stage solution to our two-stage problem. The performance of a solution (first-stage solution) is the value of the objective function from Section 5.3 for the corresponding optimal second-stage solution. The initial solution for the search, $\varphi^0 = \{\gamma^0, \lambda^0\}$, is constructed by solving the expected value problem as detailed in Section 6.2.1. We define the current solution as φ, the best local solution found as φ^L and the best solution found globally as φ^G. In each iteration an operator $o \in \mathcal{O}$ is used to define the neighborhood to draw from. The neighbourhood of solution φ defined by operator o is denoted $N_o(\varphi)$. Each operator in $\mathcal{O}$ has an equal chance of being used, and in combination, they enable reaching all feasible first-stage solutions. The operators are detailed in Section 6.2.2. In each iteration a candidate solution is evaluated by a function $h(\varphi)$, and compared to the performance of the current best local solution $h(\varphi^L)$. This evaluation function is the 2SMIP for SA-2SMIP, while it is the GCH for SA-GCH-2SMIP: A move is made if the candidate solution performs better than the current local best solution. Moves can also be made to worse-performing solutions, depending on an acceptance criterion determined by the temperature t and the solution's performance. The algorithm consists of temperature levels with a predetermined number of iterations M for each level. It starts at an initial temperature T and decreases at the end of each level according to a temperature function $d(T)$. The procedure terminates when a minimum temperature T_{min} is reached.

Algorithm 1 First stage simulated annealing improvement heuristic

Construct initial solution φ^0; $\varphi \leftarrow \varphi^0$; $\varphi^L \leftarrow \varphi^0$; $\varphi^G \leftarrow \varphi^0$ (See Algorithm 2)
while $T \geq T_{min}$ **do**
    $m = 0$
    while $m < M$ **do**
        Sample operator o from $\mathcal{O}$
        Sample solution φ from $N_o(\varphi)$
        if $h(\varphi) < h(\varphi^L)$ **then**
            $\varphi^L \leftarrow \varphi$
            if $h(\varphi^L) < h(\varphi^G)$ **then**
                $\varphi^G \leftarrow \varphi^L$
            end if
        else
            $\Delta = h(\varphi) - h(\varphi^L)$
            $\xi =$ random number, uniformly drawn from $\mathcal{U}_{[0,1]}$
            if $\xi \leq e^{-\Delta/T}$ **then**
                $\varphi^L \leftarrow \varphi$
            end if
        end if
        $m = m + 1$
    end while
    $T = d(T)$
end while
return φ^G

6.2.1 Construction of Initial Solution

Intuitively, the amount of OR time each specialty is assigned should roughly reflect that specialty's relative share of the surgical clinic's total demand. Therefore, a reasonable initial solution in Algorithm 1 is the expected value solution (EVS). As described in Section 3.4.1, this is the solution to the EV problem associated with the stochastic two-stage problem being solved. The EV problem is deterministic, and therefore significantly easier to solve to optimality than its stochastic counterpart.

While the EVS is a well-suited initial solution for instances of the MSSP with no flexibility, it is not capable of intelligently distributing flexible slots in instances where flexibility is allowed. This is because the EVS stems from a deterministic problem where the demand is certain, and thus we are indifferent to whether a slot is fixed or flexible. The distribution of flexible slots in the MSS is, however, not irrelevant in the stochastic problem. We propose two actions to remedy the EVS' indifference to flexibility. First, we demand that the number of flexible slots is constant by treating Constraint (5.2) as an equality. This ensures that we do not get an EVS with fewer flexible slots than the allowed maximum. Second, we seek to let flexible slots benefit as many specialties as possible. To exemplify, we consider a flexible slot in the EVS that is affiliated with an OR suitable for two different specialties. If one of these specialties is assigned a fixed slot that day in an OR suitable for the same two specialties in addition to other specialties, this slot is more valuable as a flexible slot. Pseudocode for the construction of initial solutions is presented in Algorithm 2.

Algorithm 2 Construction heuristic for initial first stage solution

1: Solve EV-problem with MIP-solver by treating Equation 5.2 as an equality
2: $\mathcal{D} \leftarrow$ set of days; $\mathcal{R} \leftarrow$ set of rooms
3: $\mathcal{R}_d^F \leftarrow$ set of flexible rooms on day d; $\mathcal{R}_d^{NF} \leftarrow$ set of fixed rooms on day d
4: $\mathcal{R}_d^X \leftarrow$ set of rooms with extended opening hours on day d
5: **for** $d \in \mathcal{D}$ **do**
6: **while** $\exists \left\{ i \in \mathcal{R}_d^F, j \in \mathcal{R}_d^{NF} : |\mathcal{S}_j^{\mathcal{R}}| > |\mathcal{S}_i^{\mathcal{R}}|, \mathcal{S}_i^{\mathcal{R}} \subset \mathcal{S}_j^{\mathcal{R}} \right\}$ **do**
7: $\mathcal{R}_d^F$: Add j and remove i
8: $\mathcal{R}_d^{NF}$: Add i and remove j
9: **end while**
10: **end for**

6.2.2 Operators

In the following section, we present the three operators that are necessary in order to make changes to the MSS. Together they make it possible to reach all feasible first-stage solutions, assuming the numbers of flexible and extended slots are at their upper limit.

Reassigning Fixed Slots
This operator focuses on only one slot in the MSS and makes changes within that slot. It swaps the specialty that is assigned to a given room on a given day with another specialty. The slot must be non-extended both before and after the swap, and the day, room and new specialty are all chosen randomly. A swap must satisfy all constraints. First, a new specialty can only be chosen among the specialties for whom the relevant OR is suitable. Second, there must be enough surgical teams available on that day for the new specialty.

Moving Extended Slots
This operator focuses on two slots in the MSS and performs a swap between those. An extended slot assigned to a specialty is swapped with a non-extended slot assigned to that same specialty. That means that the extension is moved from one slot to another. Both slots are chosen randomly, as long as they are not on the same day, as this would create symmetric solutions.

Swapping Fixed Slots with Flexible Slots
This operator focuses on two slots in the MSS and performs a swap between those. The first slot is a fixed slot, either extended or not, and the second slot is a flexible slot. Both slots are chosen randomly, but no swap is made between slots on the same day, as this would create symmetric solutions. The move must furthermore satisfy the following model restrictions. The fixed slot can only be moved to a room where its associated specialty is allowed to operate and to a day on which there are enough surgical teams available for that specialty.

6.3 Second-Stage Greedy Construction Heuristic

The heuristic proposed in Algorithm 1 involves evaluating first-stage solutions φ with a function $h(\varphi)$. This evaluation requires assigning values to all second-stage variables, and is an optimization problem itself. It is preferable to obtain optimal solutions to these problems, because this gives the most accurate comparison of first-stage candidates. However, as shown in Chapter 8, solving the resulting packing problem to optimality for larger instances in reasonable time, is not realistic with the 2SMIP. We propose a Greedy Construction Heuristic (GCH) to evaluate first-stage solutions. This heuristic exploits the fact that the packing problem can be decomposed into one independent subproblem for each scenario. The only dependency between the subproblems is the first-stage solution, but this is fixed beforehand. Consequently, the values of second-stage variables in one scenario have no impact on the second-stage variables of other scenarios.

The GCH uses a logic similar to that of a cutting stock problem. In the works of this thesis, an alternative formulation to the MIP presented in Chapter 5 has been developed. This formulation is inspired by the cutting stock problem presented in Appendix B. A description of the GCH follows, but looking at the cutting stock formulation can be of help to understand the logic of the GCH. The GCH involves generating patterns that show how many patients from each surgery group that can be scheduled into a slot. As opposed to assigning an integer number of individual patients to each slot, as in the mathematical model in Chapter 5, one pattern is assigned to each slot. The logic and procedure for generating patterns are detailed in Section 6.3.1, and a description of the GCH is provided in Section 6.3.2.

6.3.1 Generating Patterns

The patients of the surgical specialties are split into one or more surgery groups, each with their own surgery duration. A pattern belonging to a specialty is a combination of surgeries from its surgery groups. The surgeries are unordered and can appear up to several times. Adding together the total surgery and cleaning time of the surgeries in a pattern yields its duration, which cannot exceed the time available in a slot. Extending a slot is therefore likely to expand the number of possible patterns a specialty can take on. The patterns belonging to a specialty are the collection of all possible patterns, either in a non-extended or an extended slot. Figure 6.2 illustrates an example of the pattern generation logic while the full sets of patterns for each instance can be viewed in Appendix F.

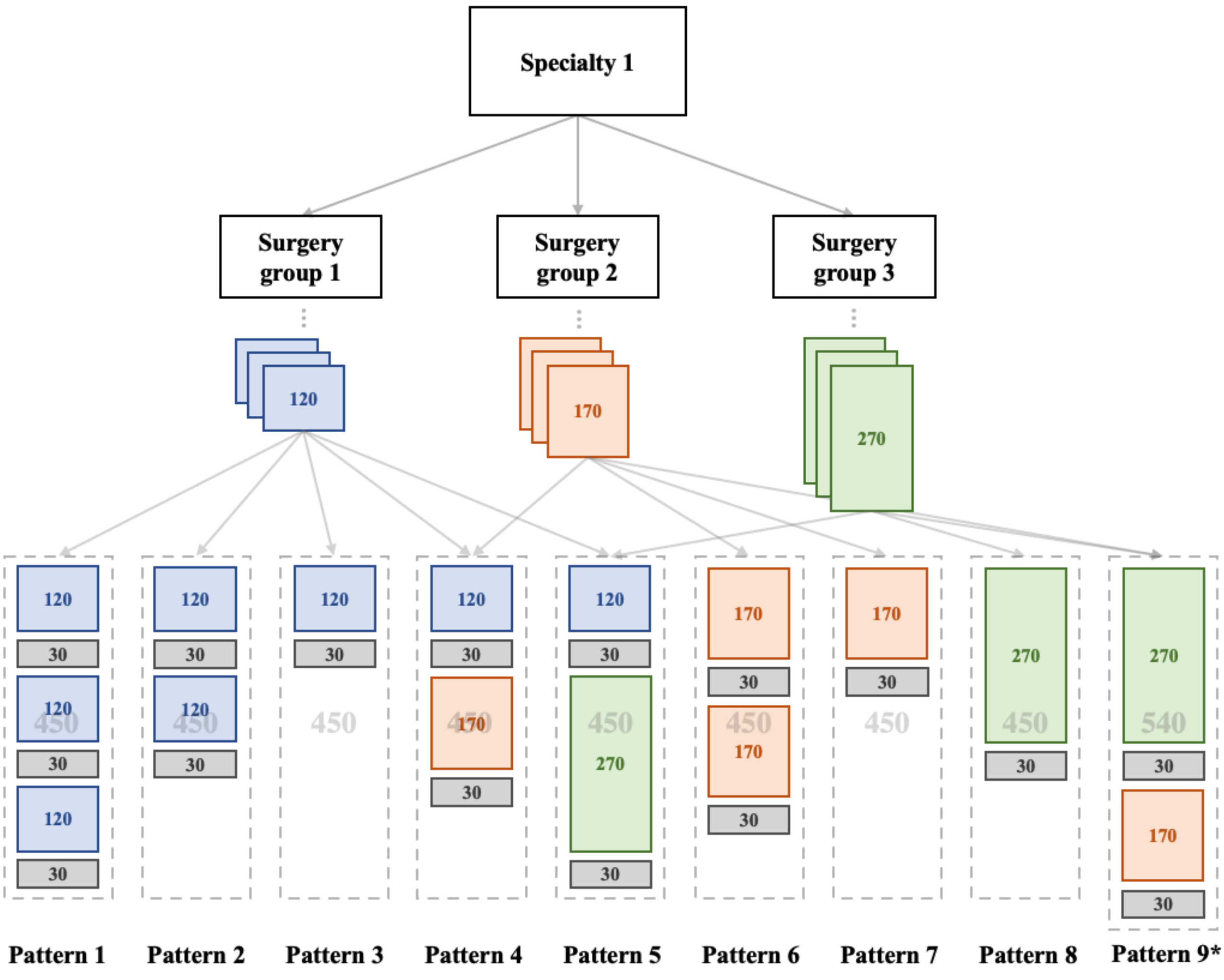


Figure 6.2: Example of pattern generation with one surgical specialty and three surgery groups with expected surgery durations of 120, 170 and 270 minutes. Slots have a total length in time of 450 minutes, with a possibility to extend opening hours with 90 extra minutes. Each procedure has an expected cleaning time of 30 minutes that must be accounted for.

6.3.2 Description of Routine

The GCH is intended to estimate the performance of first-stage solutions of the two-stage MSSP, and is detailed in Algorithm 3. To understand the procedure, consider an instance of the MSSP with $|\mathcal{C}|$ demand scenarios. At the point where Algorithm 3 is called, the first-stage variables of the problem are all fixed, and what remains is assigning flexible slots to specialties and packing the schedule with surgeries. This yields one subproblem for every scenario, meaning that we must pack $|\mathcal{C}|$ schedules for different demand scenarios. Any notation used in Algorithm 3 that is not explained below, has been introduced in Section 5.2.

Solving a scenario starts by letting the initial scenario objective, z_c, be the duration of all surgeries demanded in that scenario, including cleaning time. After this, sets of patterns for non-extended and extended slots are copied for every specialty. We refer to these sets as $\mathcal{P}_s$ and $\mathcal{P}_s^x$, respectively. Since any pattern that is feasible for a non-extended slot is also feasible for an extended slot, $\mathcal{P}_s \subset \mathcal{P}_s^x$. The sets being copied, $\mathcal{P}_{s,init}$ and $\mathcal{P}_{s,init}^x$, are generated beforehand. Since the GCH is called in every iteration of an SA heuristic, generating $\mathcal{P}_{s,init}$ and $\mathcal{P}_{s,init}^x$ each time the GCH is called would not be very efficient. Before commencing with the packing of fixed slots, infeasible patterns are removed from $\mathcal{P}_s$ and $\mathcal{P}_s^x$. This is necessary in order to avoid the unlikely case that one of the patterns include more patients from a surgery group than there is demand for that group during the entire planning period in that scenario.

The algorithm proceeds to pack fixed slots. We present three different variants of the algorithm in Section 6.3.3. All three variants assign patterns to fixed slots before considering flexible slots, but they differ in the order in which they pack within the fixed slots and the flexible slots, respectively. What follows is a general description of the GCH, independent of the order in which slots are packed. For each fixed slot the best feasible pattern, p, available to the specialty assigned to the slot, is chosen. Feasibility is checked by controlling that the capacity of care wards is not exceeded by choosing the pattern. After choosing a pattern for the slot, the duration of the pattern is deducted from the current scenario's objective. We denote the duration of a pattern D_p. In addition, the occupation of wards is updated and the sets of patterns for the relevant specialty are updated. Updating the sets involves removing patterns that contain more patients from a surgery group than there is leftover demand for that group at the current time of the heuristic.

After all the fixed slots are packed, each flexible slot is assigned to a specialty and then packed. For each flexible slot, the algorithm considers what specialty is able to pack the slot with the most valuable pattern, provided the OR is suitable for the specialty and that they have a surgical team available. When all scenarios are packed, an overall objective function is calculated. This is the estimated performance of the first-stage solution used to evaluate it in Algorithm 1.

Algorithm 3 Second-stage greedy construction heuristic

for $c \in \mathcal{C}$ **do**
    $z_c = \sum_{g \in \mathcal{G}} Q_{gc}(L_g^{SD} + T^C)$
    for $s \in \mathcal{S}$ **do**
        $\mathcal{P}_s \leftarrow \mathcal{P}_{s,init}$
        $\mathcal{P}_s^x \leftarrow \mathcal{P}_{s,init}^x$
        Remove infeasible patterns from $\mathcal{P}_s$ and $\mathcal{P}_s^x$
    end for
    for fixed slots in MSS **do**
        $s \leftarrow$ specialty assigned to the fixed slot
        if slot is not extended **then**
            $p \leftarrow$ best pattern in $\mathcal{P}_s$ not overfilling care wards
        else
            $p \leftarrow$ best pattern in $\mathcal{P}_s^x$ not overfilling care wards
        end if
        $z_c = z_c - D_p$
        Update bed ward occupation
        Remove infeasible patterns $\mathcal{P}_s$ and $\mathcal{P}_s^x$
    end for
    for flexible slots in MSS **do**
        $\mathcal{S}' \leftarrow$ specialties with available surgical teams and able to operate in this OR
        $p \leftarrow$ best pattern in $\mathcal{P}_s$ not overfilling care wards for $s \in \mathcal{S}'$
        $z_c = z_c - D_p$
        Update bed ward occupation
        Remove infeasible patterns from $\mathcal{P}_s$
    end for
end for
$z = \sum_{c \in \mathcal{C}} \Pi_c z_c$
return z

6.3.3 Variants of GCH

The order in which fixed and flexible slots are traversed in Algorithm 3 is not irrelevant. We define three variants of the GCH that are tested in Chapter 8. For each variant, we explain the way fixed slots are addressed. Each variant packs flexible slots after all fixed slots are packed, but traverses the two groups of slots by the same logic. Figure 6.3 illustrates the order in which each variant addresses slots. The subfigures are meant as examples, and not a blueprint for the schedules are packed every time one of the variants of GCH is called.

Greedy Construction Heuristic Slotwise (GCHS) traverses fixed slots by the index of the day in ascending order. Within each day, ORs are traversed by ascending index. Consequently, the first fixed slot being packed with a pattern is OR 1 on day 1 of the planning period. This is, of course, provided that this slot is not a flexible slot. See Figure 6.3a for an illustrative example.

Greedy Construction Heuristic Daywise (GCHD) traverses days by ascending index, but considers all slots of a day simultaneously. This means that the slot that can be filled with the pattern with the longest duration will be packed first and so on. When all fixed slots of a day are filled, GCHD continues with the proceeding day. See Figure 6.3b for an illustrative example.

Greedy Construction Heuristic Periodwise (GCHP) considers all unpacked fixed slots in every iteration, and fills the one that can hold the pattern with the longest duration first. In all likelihood, this involves packing the extended slots of the planning period first. See Figure 6.3c for an illustrative example.

Fixed Slots

	Odd Weeks						Even Weeks
OR \Day	1	2	3	4	5	W	...
GA-1	1	8	14	20	26	-	...
GA-2	2		15	21	27	-	...
GA-3	3	9	16		28	-	...
GA-4	4	10	17	22		-	...
GA-5	5	11		23	29	-	...
GA-6	6	12	18	24	30	-	...
GA-7	7	13	19	25		-	...

→

Flexible Slots

	Odd Weeks						Even Weeks
OR \Day	1	2	3	4	5	W	...
GA-1						-	...
GA-2		1				-	...
GA-3				3		-	...
GA-4					4	-	...
GA-5			2			-	...
GA-6						-	...
GA-7					5	-	...

(a) Greedy Construction Heuristic Slotwise (GCHS)

Fixed Slots

	Odd Weeks						Even Weeks
OR \Day	1	2	3	4	5	W	...
GA-1	4	9	16	21	27	-	...
GA-2	2		17	22	30	-	...
GA-3	3	8	18		28	-	...
GA-4	1	11	14	24		-	...
GA-5	5	10		20	26	-	...
GA-6	7	12	19	23	29	-	...
GA-7	6	13	15	25		-	...

→

Flexible slots

	Odd Weeks						Even Weeks
OR \Day	1	2	3	4	5	W	...
GA-1						-	...
GA-2		1				-	...
GA-3				3		-	...
GA-4					5	-	...
GA-5			2			-	...
GA-6						-	...
GA-7					4	-	...

(b) Greedy Construction Heuristic Daywise (GCHD)

Fixed Slots

	Odd Weeks						Even Weeks
OR \Day	1	2	3	4	5	W	...
GA-1	21	22	4	23	26	-	...
GA-2	1		5	18	24	-	...
GA-3	2	15	6		29	-	...
GA-4	7	8	25	10		-	...
GA-5	12	14		27	28	-	...
GA-6	3	16	17	20	30	-	...
GA-7	13	9	19	11		-	...

→

Flexible slots

	Odd Weeks						Even Weeks
OR \Day	1	2	3	4	5	W	...
GA-1						-	...
GA-2		4				-	...
GA-3				2		-	...
GA-4					3	-	...
GA-5			5			-	...
GA-6						-	...
GA-7					1	-	...

(c) Greedy Construction Heuristic Periodwise (GCHP)

Figure 6.3: GCH's traversing logic. The tables exemplify the order in which fixed slots are first addressed on the left, and then flexible slots afterwards on the right. The numbers represent an example of the order of addression for the three variants; in slots' ascending order in (a), in days' ascending order in (b) and in no constrained order at all in (c). This way, the heuristics are most (a), less (b) and least (c) constrained when making greedy choices in that respective order.

Intuitively, one would expect the performance of the variants above to progressively improve in the order that they are presented. However, one would also expect them to progressively become more computationally heavy. In Chapter 8, we compare the performance of these three packing procedures, both in terms of the objective function and runtime.

Chapter 7

Data and Implementation

In this section, we present how raw data has been used to generate test instances for the model formulated in Chapter 5, as well as demonstrate how the solution methods of Chapter 6 have been implemented. Section 7.1 concerns the raw data, the clustering algorithm to generate surgery groups and the scenario generation procedure. The section is inspired by its corresponding section in Asplin et al. (2021), but with substantial changes and additions. In Section 7.2 we explain how we determined what instances to test, while Section 7.3 deals with the implementation of solutions methods.

7.1 Preparing Input Data

While the model formulated in Chapter 5 is intended to be applicable to surgical clinics in general, the implementation in this thesis is based on the Clinic of Surgery at St. Olavs Hospital. Furthermore, this thesis builds on Isaksen and Svagård (2021), where data was gathered through interviews and the hospital's databases. The raw data and initial data cleaning is described in Section 7.1.1. The clustering script used in this thesis is also developed by Isaksen and Svagård, and we seek to provide a brief insight into how it works in Section 7.1.2. In Section 7.1.3 we continue by describing how scenarios are generated based on output from the clustering script.

7.1.1 Patients and Surgeries in 2019

The raw patient data stems from 2019 and is derived from two databases. The first provides information on all patients who were treated at the Clinic of Surgery in 2019, and the second provides information on all individual surgeries that were performed at the Clinic of Surgery during the year. For each registered surgery, we combine the two databases in order to extract the specialty by which the surgery was performed, the real surgery duration and the LOS of the patient in the MC wards post-surgery. Unfortunately, we have not been given access to data on the LOS at the IC wards. Consequently, these numbers are based on expert input from surgical staff at St. Olav. It is assumed that a LOS above 20 nights is unlikely. LOSs above 20 nights are therefore replaced with the median LOS of all surgical care patients, which is two days (Isaksen and Svagård, 2021).

7.1.2 Clustering Surgical Procedures

The surgery groups and affiliated parameters used in this thesis are derived by clustering NSCP-codes from surgeries performed at the Clinic of Surgery at St. Olav in 2019. To understand how the clustering works, it is necessary to understand how individual surgeries are aggregated into surgery groups. Every surgery performed by the clinic is registered with a set of NSCP-codes that define the procedure. All unique sets of NSCP-codes from the 2019 data together form the components of the surgery groups in this thesis. The relationship between individual surgeries, sets of NSCP-codes, surgery groups and specialties is illustrated in Figure 7.1. Each unique set of NSCP-codes is exclusive to one surgical specialty, so a clustering procedure is done for every specialty.

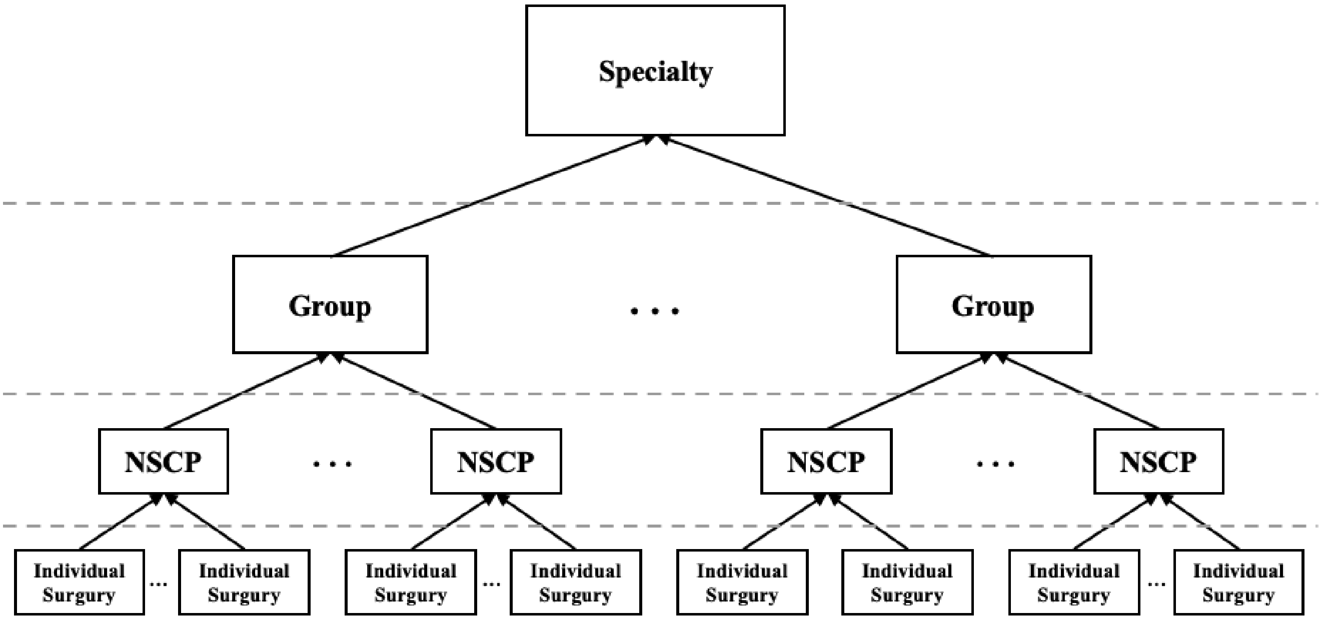


Figure 7.1: Relationship between individual surgeries, sets of NSCP-codes, surgery groups and specialties.

The idea of clustering sets of NSCP-codes is to identify groups of procedures that result in patients with similar resource usage. The most relevant resources in our problem are OR time and beds in postoperative wards. Consequently, the consumption of these resources for each set of NSCP-codes are the features that are used when clustering. The output from the clustering is a set of surgery groups for every surgical specialty, each with an average throughput for a period of two weeks during regular activity. This average throughput is derived by scaling the yearly throughput of the surgery group, which corresponds to scaling the total number of surgeries from 2019 that make up the surgery group. In addition to average throughput, the script outputs surgery time and probabilities of staying in different wards after surgery for every surgery group.

The script is based on the K-means clustering algorithm. This algorithm is initiated by randomly defining k cluster centers, called centroids, in the hyperspace spanned by the features that define the cluster objects. The first step in an iteration is then to assign each object, or set of NSCP-codes in our case, to its closest centroid in terms of euclidian distance. Step two involves recomputing the coordinates of the centroids to place them in the center of the objects that were assigned to them in the previous step. This iterative process is continued until the centroids stabilize. The features that span the hyperspace in the K-means clustering algorithm must be scalar and should be normalized. If the features are not normalized, the algorithm will put more emphasis on features where the objects vary more in absolute terms. As a result, the features used in this script have all been normalized by the method called Max-Min normalization.

7.1.3 Generating Scenarios

The generation of scenarios is performed in Python, and the following procedure outputs a scenario tree. See Table 7.1 for a description of the input parameters. For every scenario $c \in \{1, ..., |\mathcal{C}|\}$, a vector of length $|\mathcal{G}|$ is populated with random samples from a uniform distribution over [0,1]. The demand in scenario c for surgery group g is then determined by evaluating the inverse of a Poisson distribution function with parameter $\bar{Q}_g$ for the corresponding element in the random vector. An important detail about the implementation is that the demand for each surgery group in a scenario is drawn independently of others. This means that we assume no correlation between the demands of each surgery group. Also, since all scenarios are generated by drawing from the same uniform distribution, they have the same probability of occurring. Considering the probabilities in a scenario tree must sum to 1, the probability of each scenario in a tree of $|\mathcal{C}|$ scenarios must then be $\frac{1}{|\mathcal{C}|}$. In order to reproduce results, a seed is set at the beginning of the procedure. The standard pseudo-random number generator in Python is Mersenne Twister.

Table 7.1: Description of the input parameters to the scenario generation.

Parameter	**Description**
$\lvert\mathcal{C}\rvert$	Number of scenarios
$\lvert\mathcal{G}\rvert$	Number of surgery groups
$\bar{Q}_g$	Mean demand for surgery group k
S	Seed for random number generation

The assumption of the demand for different surgery groups following Poisson distributions is based on moment matching, as described in Section 3.4.2. However, the only moment of the demand distributions we have been able to infer from the 2019 data is the mean planning period demand. The quality of the data prevented us from estimating other moments, such as variance, which in turn prevented more sophisticated moment matching. Nevertheless, an underlying assumption of the Poisson distribution is that the time between arrivals is exponentially distributed. This is a common assumption when modeling queuing systems, and surgical clinics do in fact deal with patient queues. As such, we judged Poisson distributions to be the best choice considering the quality of the data. In Section 8.6, we discuss this topic further.

7.2 Generating Test Instances

In Chapter 8 we analyze a multitude of instances of the problem. These are similar in many aspects, but differ in others. In the following two subsections, we explain how we determine the number of surgery groups, bed ward capacities, and how we tune the demand of the different sets of instances. Input data for test instances is provided in Appendix F.

7.2.1 Determining the Number of Surgery Groups

In Section 2.3.1 we defined the five surgical specialties GN, GO, UR, VS and EN to be within the scope of this thesis. Isaksen and Svagård (2021) experimented with different numbers of surgery groups for these specialties, and concluded that increasing the number of surgery groups was beneficial up to a point. More surgery groups allowed for more ways of packing slots, which could lead to better utilization of OR time and downstream bed ward capacities. However, increasing the number of surgery groups also means that each group consists of fewer patients. From this, they determined instances with the number of surgery groups ranging from 5 to 25, not necessarily equally distributed between the different specialties.

In this thesis we work with three sets of instances with $|\mathcal{G}| = \{5, 9, 25\}$ surgery groups. The first of these has only one surgery group for each surgical specialty, making the number of ways a slot can be filled quite few. On the other side of the spectrum, the set of instances with 25 surgery groups yields a lot more ways to fill slots with surgeries from different surgery groups. The parameters affiliated with each set of instances are tabulated in Appendix F. The only parameters from the mathematical model in Section 5.2 that are not listed, are those for demand in each scenario and flexibility. What is listed, however, is the expected demand during a planning period for each surgery group. In Chapter 8 we generate demand scenarios based on these expected values, and test instances for a wide range of the flexibility parameter.

7.2.2 Tuning Demand and Managing Bed Ward Capacities

The three sets of instances all have an expected demand for each surgery group during a planning period. Even though the clustering algorithm described in Section 7.1.2 uses the same input data regardless of the number of surgery groups, the total demand in terms of minutes can vary by quite a lot. The total demand in terms of minutes is the sum of demand for each surgery group multiplied by the surgery group's surgery duration. For example, the instances with 25 surgery groups have 30% fewer minutes of total expected demand each month compared to instances with 9 surgery groups. This may be a result of how uncertainty in surgery duration is accounted for in this thesis. If letting the surgery duration of each surgery group be the average surgery duration of the surgeries in the cluster, we would underestimate the surgery duration for 50% of the surgeries. To plan for some slack, we have set the surgery duration of surgery groups to be the 65th percentile of the actual surgery durations in the cluster. This could explain why the total number of minutes of expected demand decreases as the number of surgery groups increases. While the instances with 9 surgery groups allow interesting analyses in Chapter 8, the demands for the two other sets of instances have to be tuned.

The expected demand per surgery group has been tuned so that the optimal solution to the EV problem for each instance just barely manages to operate on all patients. Since the objective function in our model measures the amount of unmet demand, it is hard to compare two solutions that both manage to operate on all patients. Having an expected demand that just barely allows for operating all patients in the EV problem, makes it so that instances with scenarios generally result in an objective function greater than zero.

Making changes to the parameters generated by the clustering algorithm has at least two other implications. First, we potentially move further away from the reality at the Clinic of Surgery at St. Olavs Hospital. Second, results like the objective function from instances with different numbers of surgery groups become less interesting to compare. Nevertheless, we find the instances with tuned demand to be more relevant for the purposes of this thesis, namely investigating the value of flexibility in surgical scheduling and developing solution methods for two-stage stochastic problems.

In addition to tuning demand, our implementation contains three assumptions regarding downstream bed wards. The first is that we have chosen to aggregate different bed wards into two distinct wards, namely an MC ward and an IC ward. In Chapter 2 we presented five MC wards that each belong to a surgical specialty at St. Olav, as well as two IC wards that are shared resources. Preliminary testing using these original wards proved that some of these wards form bottlenecks long before others. In talks with employees at the Clinic of Surgery at St. Olavs Hospital, it was revealed that in reality patients belonging to one surgical specialty can be admitted to an MC ward belonging to another specialty if needed. Therefore we have combined the five MC wards to one, and the two IC wards to one.

The second assumption concerns the parameter Y_{wd} introduced in Section 5.2. Remember, Y_{wd} is the expected number of occupied beds in ward w on the night following day d in the current planning period, by patients operated on in a prior planning period. Obviously, the actual value of Y_{wd} will depend on how many and what type of patients were operated on in the prior planning periods. Nevertheless, we model bed occupancy using expected values, and as such, it is only fitting that expected values are used for Y_{wd} too. To determine values for Y_{wd} we have therefore assumed that the number and types of patients operated on in the previous planning period corresponds to a scenario where the demand for each surgery group is equal to its expected demand. We assume that surgeries of patients from each surgery group have been distributed evenly across all weekdays in the prior planning period. From this, we calculate the values of Y_{wd}.

The third assumption is regarding the capacities of MC wards relative to IC wards. Remember from Section 5.2, B_{wd} is the number of available beds at ward w on the night following day d. Preliminary testing in Chapter 8 reveals that there is considerable slack in bed ward capacity constraints for the parameter values implemented for B_{wd}. Therefore, we perform several analyses in Chapter 8 where we reduce the capacity of bed wards. The reduction is controlled through a scaling factor β which is multiplied with B_{wd} for $w \in \mathcal{W}$ and $d \in \mathcal{D}$. We investigate instances for $\beta \in \{0.4, 0.5, 0.6, 1.0\}$, where $\beta = 1.0$ means original bed ward capacity and $\beta = 0.4$ corresponds to a 60% reduction in bed ward capacities. The underlying assumption is therefore that the capacity of MC wards relative to IC wards stays the same when they are both reduced from the original capacities provided by St. Olav.

7.3 Implementing Solution Methods

The MIP from Chapter 5 has been implemented in Python and solved using Gurobi Optimizer (version 9.5.1). The computer used has an Intel® CoreTM i7-10700 CPU @ 2.90GHz, 16.0 GB of RAM and runs on a 64-bit Windows 10 Education operating system. Regardless of what solution method is used, input is read from an Excel file. There exists one such file for each set of surgery groups, and scenarios are generated in Python based on parameters from Excel. Replicability in the scenarios is ensured by setting a seed of choice at the beginning of the generation procedure. Tweaking of other parameters, like flexibility and capacities in wards, is done in the Python code. In this way, several model runs for ranges of parameters can be performed in loops, without intervention.

The implementation of the FMIP is straightforward in Gurobi. The time limit is set before the FMIP is run, and if an optimal solution is found earlier, then the solver will terminate. The FMIP-2SMIP is similar, and will simply fix the first-stage solution found by the FMIP at termination, and find the corresponding optimal second-stage solution. The implementation of the SA heuristics requires some more explaining, and we dedicate the remainder of Section 7.3 to describing this.

7.3.1 Simulated Annealing with 2SMIP in the Second-Stage

As described in Section 6.2.1, the SA heuristics are initiated by solving the EV problem with the FMIP. The values of the first-stage variables are then stored in a Python dictionary. From this point, the implementations of the SA heuristics differ, depending on whether the evaluation of the candidate solution is performed by the 2SMIP or by the GCH. The implementation of the SA-GCH-2SMIP is described in Section 7.3.2. In the next few paragraphs, we focus on the SA-2SMIP.

Before starting the search, the first-stage solution obtained from the EVS has to be evaluated when applied to the actual scenarios of the problem instance. Using the recorded values from the EVS, first-stage variables in the Gurobi model object are fixed. In practice, this involves setting both the lower and upper bound of the variables equal to the recorded values. The 2SMIP is then solved to optimality, providing an objective value that is used when comparing to neighboring first-stage solutions in the procedure that follows.

Using a MIP solver to consecutively solve several instances of the same problem can be very time-consuming if the solver has to start from scratch in every iteration. Before the solver can start searching the solution space, a Gurobi model has to be created. Creating the model and finding an optimal solution can take minutes, and not seconds, for some of the instances tackled in this thesis. Spending minutes evaluating every candidate solution would make the SA-2SMIP heuristic very slow. Fortunately, Gurobi has a couple of features that can help mitigate this. Firstly, a Gurobi model can be stored temporarily in a model object. This object has attributes like variables and constraints that can be altered during runtime, making it possible to change the bounds of two variables for example. In this way, a new model object does not need to be created in every iteration.

Even after the model is built though, finding an optimal second-stage solution from scratch can be time-consuming. However, evaluating a neighboring first-stage solution essentially means changing a couple of bounds and solving a very similar problem. The solution to this problem resembles that of the previous one in most cases, and we exploit this by using a second feature of Gurobi. In a file format called .mst, the values of variables can be stored and quickly loaded into a model object. This makes reoptimization from a slightly different first-stage solution a lot faster, since the search is initiated close to the optimal solution. If our heuristic search was an engine, using .mst files would equate to keeping the engine warm. To give an understanding of the efficiency gained by reusing Gurobi model objects and storing solutions in .mst files, we consider the solution time for 2SMIP for an instance with $|\mathcal{G}|= 9$ surgery groups and $N = 250$ scenarios. If the model has to be created and solved without reoptimization from a similar solution, the build takes 2163 seconds, while finding the optimal second-stage solution takes 40 seconds. If we have stored a model object and an .mst file from the previous iteration, however, the model does not need to be built and reoptimazation takes 10 seconds. Figure 7.2 illustrates how a Gurobi model object is used to evaluate neighboring first-stage solutions during the SA-2SMIP.

7.3.2 Simulated Annealing with GCH in the Second-Stage

The GCH is implemented with a focus on computational efficiency. When running the SA-GCH-2SMIP, the GCH uses a quite different format to store solutions compared to the FMIP and 2SMIP. The mathematical formulation in Section 5.3 uses the variables γ_{srd}, λ_{srd} and δ_{srdc} to couple specialties with fixed, extended and flexible slots. The GCH on the other hand comprises this information in a few matrices.

To describe a first-stage solution, two matrices with a row for each OR and a column for each day in the planning period are used. The first matrix contains an index in each element, representing the specialty that is assigned to that slot as a fixed slot. Flexible slots share the same index, since they are not assigned yet in a first-stage solution. The second matrix has binary elements, signifying whether slots are extended or not. A second-stage solution is stored in $|\mathcal{C}|$ matrices, one for each scenario, of the same dimensions as above. Each element in these matrices contains the index of the pattern of surgeries assigned to the slot in that scenario. A flowchart illustrating the implementation of the SA-GCH-2SMIP is presented in Figure 7.3.

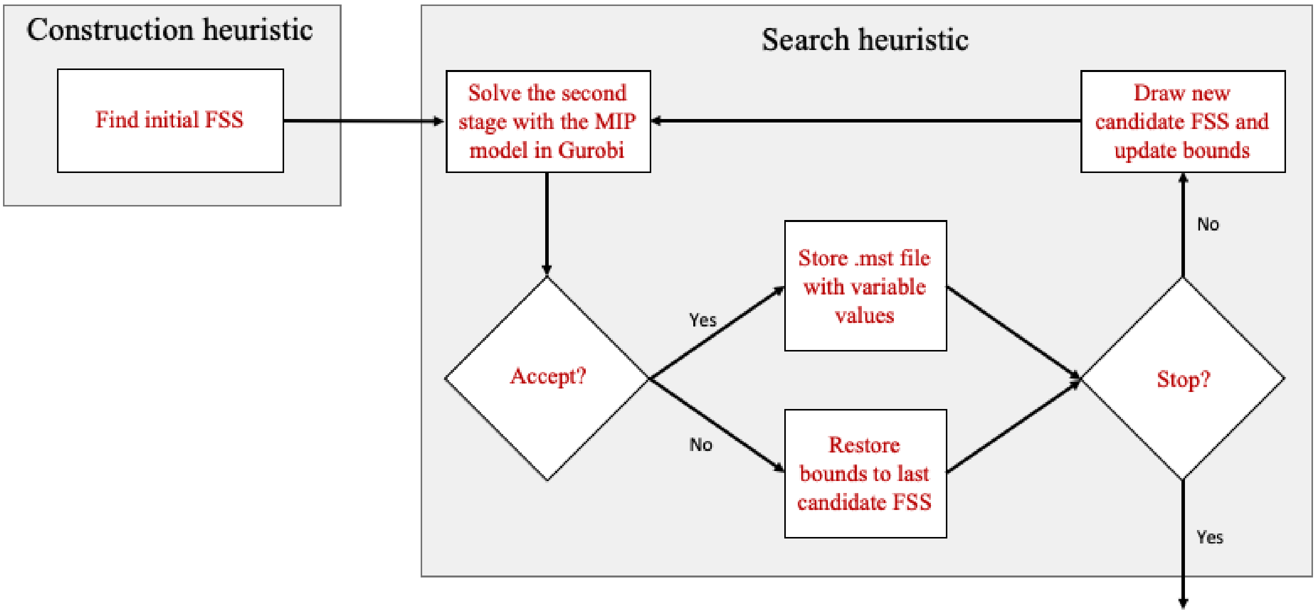


Figure 7.2: Flowchart of implemented SA-2SMIP showing how a Gurobi model object is continually used to evaluate candidate first-stage solutions (FSS), and how variable values are stored in .mst files after moves.

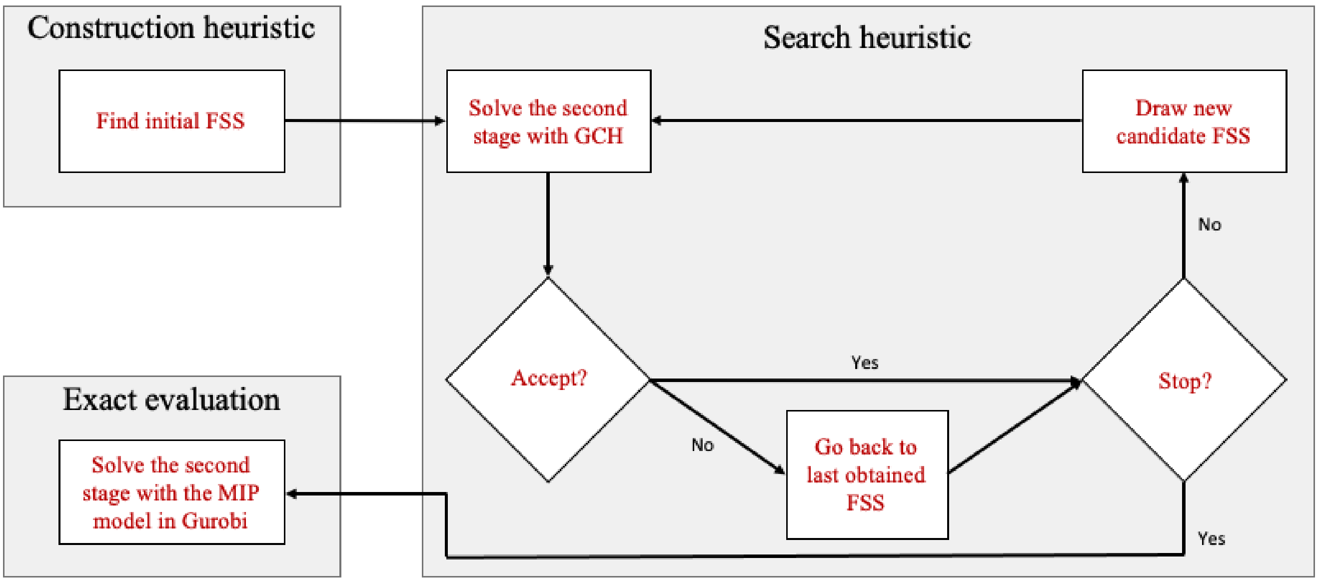


Figure 7.3: Flowchart of implemented SA-GCH-2SMIP showing how GCH is used to evaluate candidate first-stage solutions (FSS). First-stage solutions are stored in matrices that are altered during runtime. The best first-stage solution found at the end of the search heuristic is evaluated by using the MIP.

Chapter 8

Computational Study

This section contains analyses and discussion around results obtained through a computational study of the Master Surgical Scheduling Problem (MSSP) presented in Chapter 4. Technical aspects related to the optimization problem are covered in the first four sections. This includes investigating problem size and drivers of complexity, as well as finding good scenario trees and tuning parameters for the SA first-stage heuristic. The technical analysis culminates in Section 8.4 by comparing the performance of the implemented solution methods. In Section 8.5, we apply the solution methods in order to assess the value of flexibility in the MSSP. Lastly, in Section 8.6, we discuss the limitations of this study and address some interesting topics for future research.

8.1 Problem Size and Complexity

In this subsection, we provide statistics showing the size and complexity of a range of problem instances. We lean on these results when choosing problem instances to investigate in the continuation of this computational study, as well as motivate the need for the heuristic solution methods introduced in Chapter 6. All tests are from runs with the FMIP in Chapter 5 solved by Gurobi Optimizer. We define problem complexity as the time it takes to obtain optimal solutions. For instances where obtaining optimal solutions is not possible in one hour, we express complexity as the percentage optimality gap. In this thesis, we follow the convention that an optimality gap for minimization problems is $(primal - dual)/primal$. The size of instances is expressed as the number of rows, columns, binary variables and nonzero elements after the presolver in Gurobi has been run.

Table 8.1 presents results for instances with a range of number of surgery groups and scenarios. As is evident by the solution time in the rightmost column, the instances with 5 surgery groups are quite easily solved to optimality in ten minutes or less. One could, however, argue that these instances are the least interesting to investigate out of the nine tabulated. With only one surgery group per surgical specialty, these instances generalize reality quite extensively. In this regard, the instances with 9 and 25 surgery groups probably capture the dynamics of planning at a surgery clinic to a larger extent.

In mathematical modeling, there is generally a trade-off between modeling reality accurately and formulating a model that is easy to solve. The optimality gaps [%] in Table 8.1 reveal that the instances with 9 surgery groups require a long time to obtain reasonable optimality gaps, and even more so for the ones with 25 surgery groups. Furthermore, these instances are likely to require high numbers of scenarios to achieve stability, as discussed in Section 8.2. With this in mind, it is natural to consider heuristic solution methods.

Table 8.1: Problem size and complexity at different instances of groups $\mathcal{G}$ and scenarios $\mathcal{C}$ at flexibility $F = 10\%$ and normal bed ward capacity $\beta = 1.0$.

$\lvert\mathcal{G}\rvert$	F	$\lvert\mathcal{C}\rvert$	β	Prim [min]	Dual [min]	Gap	Presolver Row	Col	Bin	Nonzero	Time [s]
5	*10%*	*10*	*1.0*	667	667	0,0%	6310	5439	2810	53197	*124*
5	*10%*	*50*	*1.0*	1087	1087	0,0%	31205	26398	13253	265354	*161*
5	*10%*	*100*	*1.0*	1002	1002	0,0%	62330	52607	26338	530621	*603*
9	*10%*	*10*	*1.0*	310	236	23,7%	6404	7868	5024	84826	**3600*
9	*10%*	*50*	*1.0*	792	565	28,7%	31480	38464	23918	423978	**3600*
9	*10%*	*100*	*1.0*	2126	606	71,5%	62883	76627	47637	852415	**3600*
25	*10%*	*10*	*1.0*	1580	738	53,3%	6571	17876	7222	197845	**3600*
25	*10%*	*50*	*1.0*	3494	225	93,6%	32340	88685	35470	990299	**3600*
25	*10%*	*100*	*1.0*	2190	325	85,2%	64572	176817	70159	1973904	**3600*

All scenario trees have been generated by the same seed $m = 1$.
Row, Col, Bin and Nonzero refer to the constraints, variables, binary variables and non-zero elements in the model after the presolver have been run.
* means that the maximum runtime is reached.

The size and complexity of a problem are typically positively correlated, but not without exceptions. We see that the FMIP yields a smaller optimality gap for the largest instance than for the second-largest instance in Table 8.1, but the results generally display larger optimality gaps for larger instances. However, the problem complexity can also fluctuate depending on parameter values. Figure 8.1 shows the optimality gaps after one hour for different levels of flexibility in (a) and different levels of bed ward capacity in (b). Based on (a), the problem complexity seems to increase with more flexibility. This is likely because the number of combinations of second-stage values increases. It could be that the growing solution space that comes with increasing flexibility takes more time to search through.

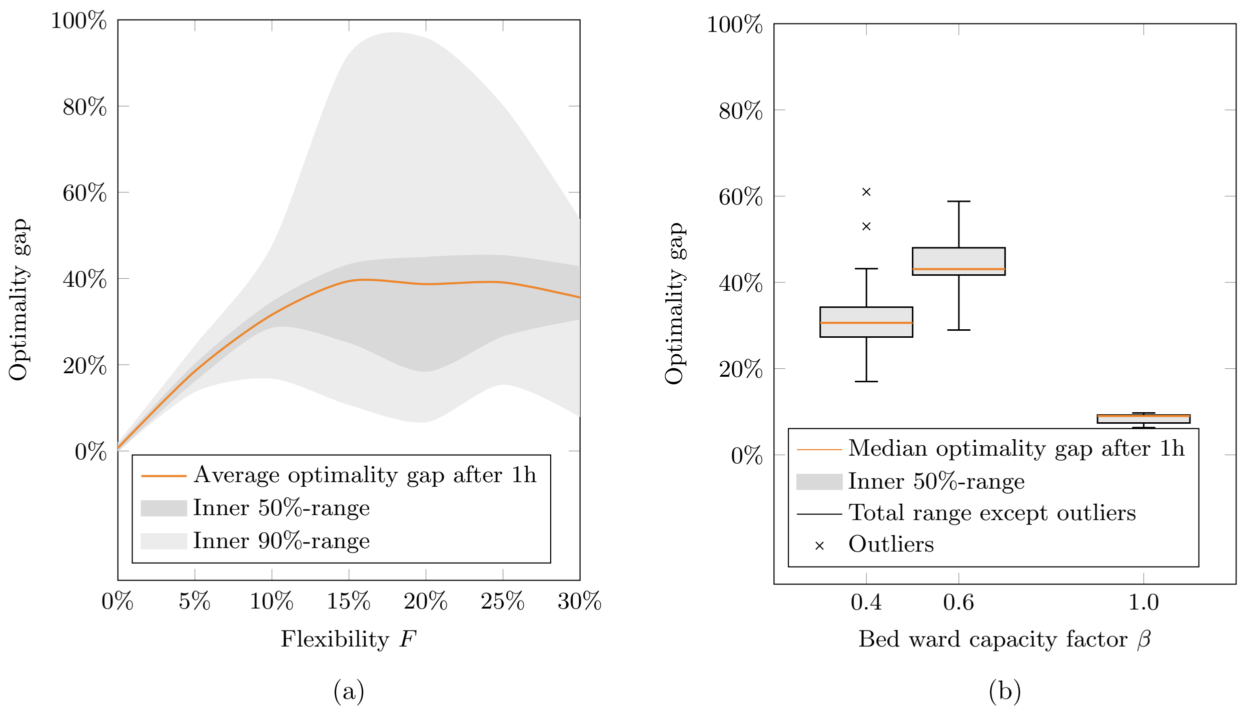


Figure 8.1: Optimality gaps after one hour for different levels of flexibility F and bed ward capacity β. The problems in both (a) and (b) are run with a scenario tree of size $N = 50$ generated from the seeds $m \in \{1, 2, \ldots 30\}$. Furthermore, the problems in (a) are run with bed ward capacity factor $\beta = 1.0$, and the problems in (b) with flexibility $F = 10\%$.

Looking at the trend line in (b), it seems like the problem becomes more complex as we reduce the capacity to $\beta = 0.6$. However, the problem complexity seems to decrease when β moves from 0.6 to 0.4. We hypothesize that this is due to the amount of slack in the model constraints at the three different bed ward capacities. At β=1.0, there may be enough capacity at the bed wards so that surgeries can be planned in ORs without really worrying about the wards downstream. The way in which the ORs are utilized is then defining for the objective outcome, making the ORs the constraining resources. That leaves a lot of slack in the constraints associated with the bed ward capacity. At β=0.4 on the other hand, the bed ward capacity is the limiting resource. In these instances, it might not matter so much how the ORs are utilized, since the bed wards are filled up long before the ORs anyway. In turn, this could be the reason why these instances appear to be less complex than at higher values of β.

At β=0.6, we believe that there is little leeway in both how the ORs and the bed wards are utilized most effectively. As a result, there is little slack in the constraints associated with both resources. The variable interdependencies have a bigger impact on complexity when there is little slack in the constraints connecting them. This is because one small change is likely to demand many more changes, in order to keep satisfying all constraints. The combination of variable interdependencies and little slack in the constraints connecting them then increase the problem complexity.

8.2 Stability

In this section, we determine the appropriate size of scenario trees for instances with $|\mathcal{G}| = \{5, 9, 25\}$ surgery groups. The goal is to find the size needed in order for a first-stage solution's performance to not depend on how the tree is generated. For the scenario generation procedure implemented in this thesis, this means that the performance of a first-stage solution when tested on a scenario tree should not depend on what seed and size were used when generating that scenario tree.

With our notation, the out-of-sample stability criteria proposed by Kaut and Wallace (2003) reads:

$$\min_{\boldsymbol{\phi} \in \Phi} f\Big(\boldsymbol{\phi}; \boldsymbol{\varphi}_N^{m*}, \tilde{\boldsymbol{\xi}}\Big) \approx \min_{\boldsymbol{\phi} \in \Phi} f\Big(\boldsymbol{\phi}; \boldsymbol{\varphi}_N^{k*}, \tilde{\boldsymbol{\xi}}\Big) \qquad m, k \in 1, 2, \dots M \tag{8.1}$$

$$\boldsymbol{\varphi}_N^{m*} := \Big\{ \{\boldsymbol{\gamma}, \boldsymbol{\lambda}\} : \underset{\boldsymbol{\phi} \in \Phi}{\operatorname{argmin}} \ \hat{f}(\boldsymbol{\phi}; \breve{\boldsymbol{\xi}}_N^m) \Big\} \tag{8.2}$$

Where $\boldsymbol{\phi}$ is the full set of decision variables $\{\boldsymbol{\gamma}, \boldsymbol{\lambda}\ , \boldsymbol{\delta}, \boldsymbol{x}\}$ from the feasible set $\boldsymbol{\Phi}$ defined in Chapter 5. $\boldsymbol{\varphi}_N^{m*}$ is the set of optimal first-stage decisions $\{\boldsymbol{\gamma}^*, \boldsymbol{\lambda}^*\}$ for a scenario tree of size N generated with seed m. We denote this scenario tree $\breve{\boldsymbol{\xi}}_N^m$, while $\tilde{\boldsymbol{\xi}}$ denotes the true stochastic process. Lastly, $\hat{f}$ denotes the objective function of the minimization problem with $\hat{f}(\boldsymbol{\phi}) \to f(\boldsymbol{\phi})$ when $\breve{\boldsymbol{\xi}} \to \tilde{\boldsymbol{\xi}}$.

As our scenario trees are sampled using Monte Carlo simulation, we know the true stochastic process and wish to find at which size N, $\breve{\boldsymbol{\xi}}_N \approx \tilde{\boldsymbol{\xi}}$. Note that $\breve{\boldsymbol{\xi}}_N \to \tilde{\boldsymbol{\xi}}$ when $N \to \infty$. In other words, how large our scenario trees must be in order to properly represent the underlying probability distribution for the surgery groups' demand. At this size, the scenario tree $\breve{\boldsymbol{\xi}}_N$ serves as a proxy for the true stochastic process.

A drawback with the process described above is that finding N means finding optimal first-stage solutions to potentially very large instances of the MSSP. As revealed in Section 8.1, this is not feasible for our FMIP. Therefore, we take an alternative approach to finding N. A mathematical description follows below in a procedure inspired by SAA.

Procedure

1. Generate M' first-stage solutions by solving different SAA problems with small scenario trees of size N'

$$\boldsymbol{\varphi}_{N'}^{m'*} := \Big\{ \{\boldsymbol{\gamma}, \boldsymbol{\lambda}\} : \underset{\boldsymbol{\phi} \in \Phi}{\operatorname{argmin}} \ \hat{f}(\boldsymbol{\phi}; \breve{\boldsymbol{\xi}}_{N'}^{m'}) \Big\} \qquad m' = \{1, 2, \ldots, M'\} \tag{8.3}$$

2. For each first-stage solution m':

   (a) Evaluate the performance of the first-stage solution on scenario trees of growing size up to $\overline{N}$, with M trees of each size

$$z_N^{m',m} = \min_{\boldsymbol{\phi} \in \Phi} \hat{f}\Big(\boldsymbol{\phi}; \boldsymbol{\varphi}_{N'}^{m'*}, \breve{\boldsymbol{\xi}}_N^m\Big) \qquad m \in \{1, 2, \ldots, M\}, N \in \{1, 2, \ldots, \overline{N}\} \tag{8.4}$$

   (b) For each tree size N, calculate the empirical variance between the objective function values of the M different trees

$$\hat{\sigma}_N^{m'2} = \frac{1}{M-1} \sum_{m=1}^{M} \left(\hat{z}_N^{m',m} - \frac{1}{M} \sum_{i=1}^{M} \hat{z}_N^{m',i} \right)^2 \qquad N \in \{1, 2, \ldots, \overline{N}\} \tag{8.5}$$

3. For each tree size N, find the mean empirical variance $\bar{\sigma}_N^2$

$$\bar{\sigma}_N^2 = \frac{1}{M'-1} \sum_{m'=1}^{M'} \hat{\sigma}_N^{m'2} \qquad N \in \{1, 2, \ldots, \overline{N}\} \tag{8.6}$$

4. Choose appropriate tree size N^* as the smallest N where there is no significant reduction in mean empirical variance by growing the tree further

Results and findings

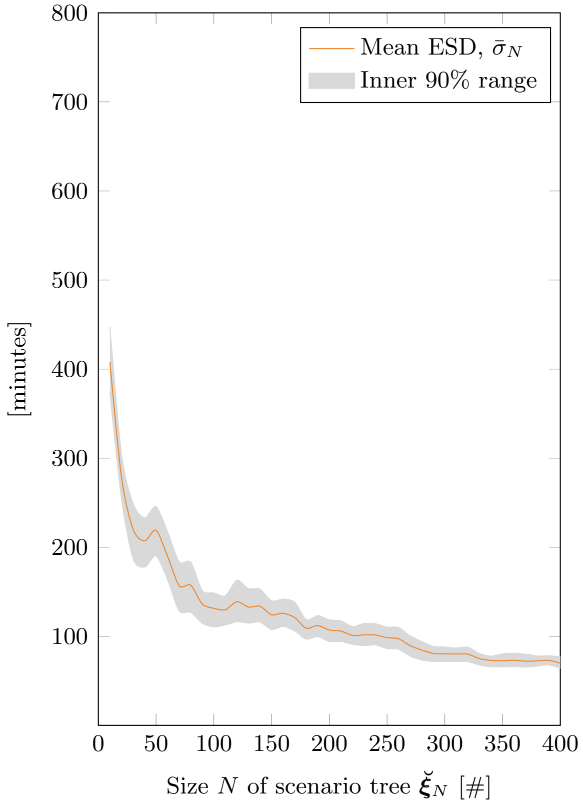
One stability test is performed for each of $|\mathcal{G}| \in \{5, 9, 25\}$ with flexibility $F = 0\%$ and initial bed ward capacities with $\beta = 1.0$. For each number of surgery groups, the tree size N is increased with increments of 10 until there is no visible benefit in increasing it. Furthermore, since the SAA method seeks to approximate the objective value by averaging values derived from M scenario trees of each size, variance in the objective value can be somewhat reduced by increasing M. We choose $M = 30$, weakly grounded in the assumption that 30 is the minimal sample size where it is in our view reasonable to assume that the Central Limit Theorem holds. For the same reason, we choose $M' = 30$. Lastly, we choose trees of size $N' = 3$ when generating the independent first-stage solutions, as these problems are easy to solve.

We follow the procedure presented above, except that instead of solving $z_N^{m',m}$ to optimality in step 2(a), we estimate it heuristically with GCHS as presented in Section 6.3.3. This is necessary, since the tests involve solving $(M \times M')$ second-stage problems for each tree size N, which is too time-consuming to solve to optimality with the 2SMIP. Although estimating the objective values can seem inaccurate, we see in Section 8.3.1 that the GCHS is closely correlated with the optimal solution.

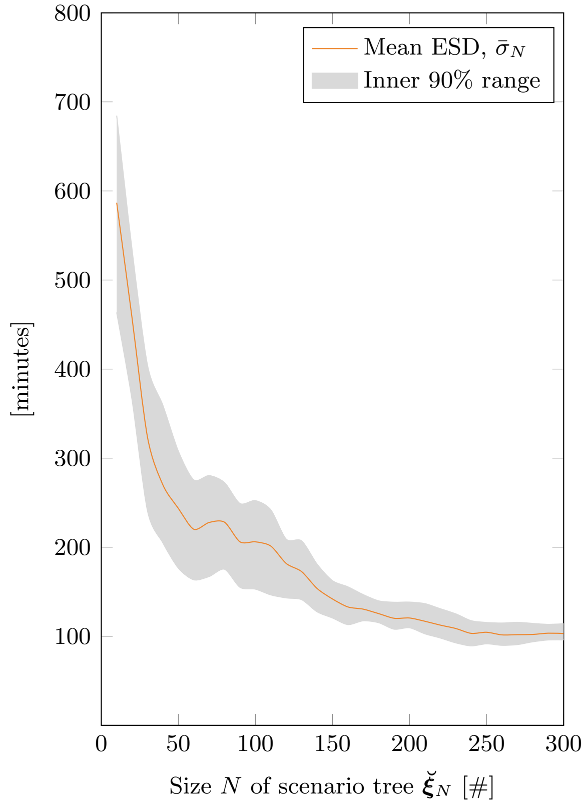
Figure 8.2 shows how the mean empirical standard deviation $\bar{\sigma}_N$ and its spread is developing as N increases. We see from the figure that $N^* \approx \{350, 250, 600\}$ for $|\mathcal{G}| = \{5, 9, 25\}$ seems like a reasonable choice. From these points, there seems to be little to no reduction in standard deviation by growing the scenario trees. Our hypothesis is that at these levels, we obtain the smallest scenario trees that properly represent the true stochastic process. Consequently, these sizes will be used when solving and analyzing the problem for the rest of this computational study.

To give credibility to the findings about tree size, we apply one of the tests for out-of-sample stability suggested by Kaut and Wallace (2003). In Figure 8.12 of Section 8.5.1, we see results from using the SA-2SMIP to solve the instance of $|\mathcal{G}| = 9$, flexibility $F = 0\%$, $\beta = 1.0$ and $N = 250$. We refer to the optimal objective value found for this instance as z. If we evaluate the same first-stage solution on a significantly larger scenario tree of size $N = 2000$ and find the optimal second-stage solution, we can test for out-of-sample stability. The optimal objective value of the first-stage solution fixed on the $N = 2000$ scenarios is denoted z'. The relative difference between the objective values z and z' is less than 0.23%. This suggests that our findings are reasonable, or at least that $N^* \approx 250$ for 9 surgery groups is a well-performing scenario tree.

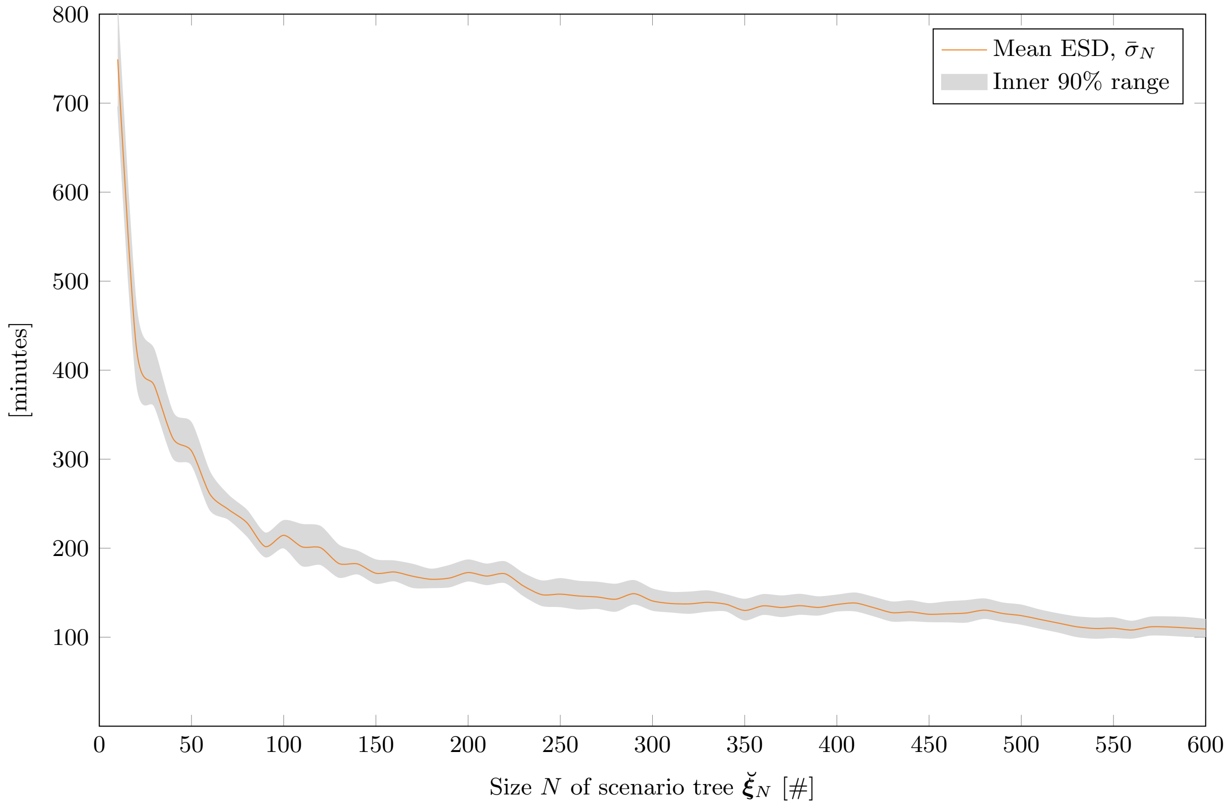
The fact that instances with $|\mathcal{G}| = 5$ surgery groups seem to require more scenarios than those with $|\mathcal{G}| = 9$, is somewhat surprising. Considering the number of surgery groups represents the number of stochastic parameters the instances have, we might expect that the number of scenarios needed to achieve stable scenario trees would increase with the number of surgery groups. We speculate that the reason why instances with $|\mathcal{G}| = 5$ surgery groups need so many scenarios is a result of the surgery duration of the surgery groups. Remember, these instances only contain one surgery group per surgical specialty, yielding few ways of packing ORs with surgeries. Some of the surgery groups have surgery durations that give very low utilization of ORs, just over 50%. The reason why these instances need many scenarios to stabilize could therefore be a result of high variance in the objective for each scenario. A scenario with abnormally many patients from one of the surgery groups that yield low utilization of ORs will probably give a much worse objective value. Such a variance between objectives in different scenarios could explain why so many scenarios are needed before the scenario trees become stable.



(a) $|\mathcal{G}| = 5$



(b) $|\mathcal{G}| = 9$



(c) $|\mathcal{G}| = 25$

Figure 8.2: Empirical standard deviation $\bar{\sigma}_N$ in the objective value for $M' = 30$ randomly generated first-stage solutions, $\boldsymbol{\varphi}_N^{m'}$ estimated with GCH for $|\mathcal{G}| \in \{5, 9, 25\}$ at $M = 30$ different scenario trees $\breve{\boldsymbol{\xi}}_N^m$ of each size N. These $\boldsymbol{\varphi}_N^{m'}$ solutions are constructed by running the FMIP with flexibility $F = 0\%$, initial bed ward capacity $\beta = 1.0$ at $M' = 30$ different scenario trees $\breve{\boldsymbol{\xi}}_3^{m'}$ of size $N' = 3$.

8.3 Preliminary Testing and Parameter Tuning of the Heuristics

This section contains two parts. The first, Section 8.3.1, concerns the GCH and how well it serves as a substitute for the 2SMIP when evaluating first-stage solutions. In Section 8.3.2, we briefly describe how the parameter values for the SA heuristic are determined.

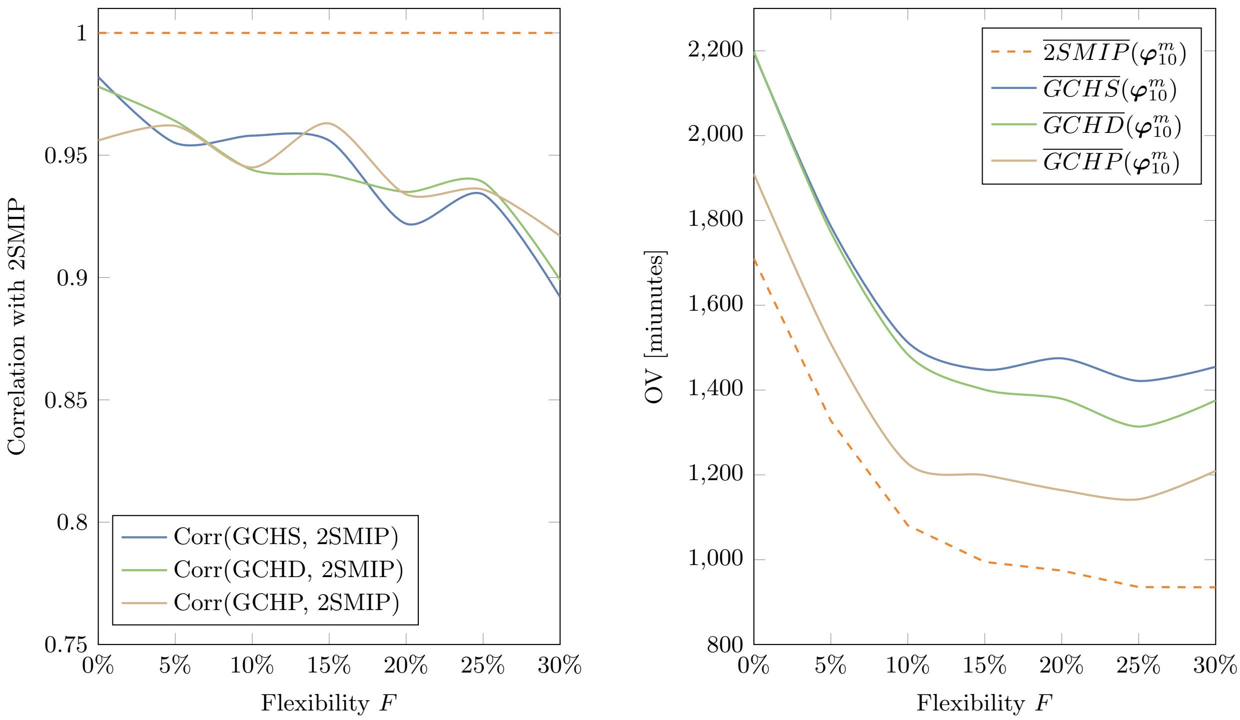
8.3.1 Greedy Construction Heuristics's Performance

In Section 6.3.3, we introduced three different variants of the GCH. Each have their own way of traversing the slots in a first-stage solution when creating a second-stage solution. As we shall see in Figure 8.3 and Figure 8.4, the variants perform differently. These two figures display results from tests on instances with 9 surgery groups and a range of flexibility levels. Figure 8.3 represents original bed ward capacity, while the capacity is halved in Figure 8.4, where $\beta = 0.5$. This is the only part of the computational study where this value of β is used. It is meant to be representative of the two other values of β that are used when reducing bed ward capacities, namely $\beta \in \{0.4, 0.6\}$.

We assess the variants of GCH across three performance measures. These are correlation with the optimal second-stage solution, runtime and the objective value itself. Correlation with the optimal second-stage solution is an important measure because the GCH is intended to be a time-efficient proxy for the 2SMIP when solving the second-stage. Since candidate first-stage solutions in the SA heuristics are accepted or rejected based on their objective value when packed in the second-stage, we would like the GCH to be as closely correlated with the optimal second-stage solution as possible. Since this heuristic is implemented in order to be computationally efficient, the time it takes to evaluate a first-stage solution is also a relevant performance measure. We would like to see the GCH come as close as possible to the optimal objective value, but this is less important than correlation and runtime.

The graphs in Figure 8.3 and Figure 8.4 are averaged values for second-stage solutions of 30 first-stage solutions at each level of flexibility visible on the horizontal axes. The first-stage solutions are generated by using the FMIP to find the optimal first-stage solution to different problem instances with $N' = 3$ scenarios. These first-stage solutions are then fixed on new scenario trees of size $N = 10$ and packed by each variant of the GCH, as well as being packed to optimality by the 2SMIP. Using modestly sized scenario trees with $N = 10$ when fixing the first-stage solutions is done in order for the 2SMIP to be able to solve the second-stage problems in reasonable time. However, since the scenarios can be packed independently in the second-stage, we believe the results from these tests are representative for larger size scenario trees as well.

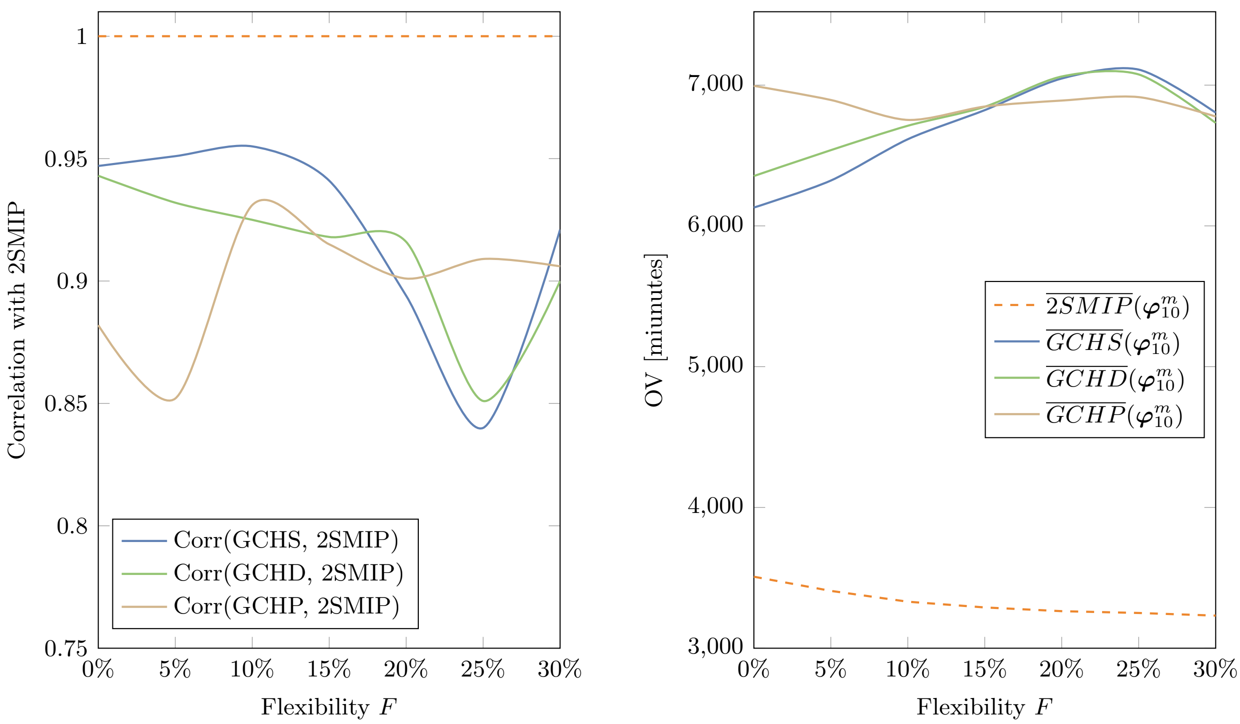
First we consider the performance of the GCH variants for instances with the original bed ward capacity as presented in Figure 8.3. The correlation with the optimal objective value is above 0.9 for all variants when the flexibility level is 25% or lower. The correlation seems to diminish as the flexibility increases. The reduction in correlation could be due to the fact that fixed slots are always packed before flexible slots when using GCH. This could lead to second-stage solutions where fixed slots are packed with patterns that demand a lot of beds downstream, while the bed wards could have been utilized more efficiently by prioritizing the packing of a flexible slot. If there exists a flexible slot that can be assigned to a specialty that can utilize the OR better with patients demanding less capacity downstream, then this is superior to packing all fixed slots first. Looking at the variants' average objective value relative to the optimal solution in (b) of Figure 8.3, gives the impression that GCHP is significantly closer to the optimal solutions than the two other variants.



(a) Correlation

(b) Average objective value

Figure 8.3: The plots compare objective values estimated with the GCH variants versus the optimal objective values for $M = 30$ random generated first stage solutions φ_{10}^m with $|\mathcal{G}| = 9$ number of groups. These solutions are constructed by running the 2SMIP with initial bed ward capacity $\beta = 1.0$ at $M' = 30$ different scenario trees $\breve{\boldsymbol{\xi}}_3^{m'}$ of size $N' = 3$ for each flexibility level.



(a) Correlation

(b) Average objective value

Figure 8.4: The plots compare objective values estimated with the GCH variants versus primal objective values found by the 2SMIP for $M = 30$ random generated first stage solutions φ_{10}^m with $|\mathcal{G}| = 9$ number of groups. These solutions are constructed by running the 2SMIP with initial bed ward capacity $\beta = 0.5$ at $M' = 30$ different scenario trees $\breve{\boldsymbol{\xi}}_3^{m'}$ of size $N' = 3$ for each flexibility level for 300s with an average optimality gap $< 3\%$.

The performances of the GCH variants look less promising in Figure 8.4. The correlation in (a) is high for GCHS and GCHD for flexibility levels below 15%. However, none of the variants demonstrate stable, high correlation across all levels of flexibility. While the average optimal solutions become better in (b), the GCH variants actually perform worse as the flexibility increases beyond 15%. Without investigating this further, we speculate that the effect has the same cause as for $\beta = 1.0$, discussed above.

The time it takes for the three different variants to finalize a first-stage solution on a scenario tree of size $N = 10$ is listed in Table 8.2. GCHP takes more than 30 and 6 times more time than GCHS and GCHD, respectively. The runtime is negligible for an instance of this size, but will grow proportionally to the number of scenarios. This is because the scenarios are packed one by one, and thus $N = 100$ scenarios take 10 times longer to pack than $N = 10$. Even though we know from Section 8.1 that the complexity of the problem is affected by the capacity of bed wards, the value of β has no significant impact on the run times of the GCH variants. This can be explained by the fact that the procedure of these three heuristic solution methods remains more or less unchanged, regardless of how much capacity there is downstream. For comparison, the 2SMIP packs the instances where $\beta = 1.0$ in just under a second, but needs more than 300 seconds to reach below 3% optimally gaps for some instances where $\beta = 0.5$.

Table 8.2: Average runtime from using the three presented variants of GCH to find a second-stage solution for a general first-stage solution $\boldsymbol{\varphi}$ on a scenario tree $\breve{\boldsymbol{\xi}}_{10}$ of size $N = 10$ across all flexibility levels F.

Bed ward capacity factor	**GCHS**	**GCHD**	**GCHP**
$\beta = 1.0$	0.0142s	0.0715s	0.4271s
$\beta = 0.5$	0.0145s	0.0602s	0.4275s

In conclusion, all three variants of the GCH are best suited for solving instances where the downstream resources of bed wards are not running on high utilization. For $\beta = 1.0$, however, the variants show a high correlation with the optimal solution. The most advanced variant of GCH, namely GCHP, on average comes closer to the optimal solution. However, the correlations with the optimal second-stage solution and the runtime are the most important performance measures. As long as the correlation with the optimal second-stage solution is high, the evaluation of candidates should be fair. The best solution found will be evaluated with the 2SMIP at the end of SA-GCH-2SMIP, so how close the GCH can come to the optimal second-stage objective value is not as important as the correlation. Keeping the runtime as low as possible is also beneficial for saving time. Therefore, we proceed this computational study using the GCHS. For simplicity, we will continue to write GCH, but keep in mind that this refers to GCHS.

8.3.2 Simulated Annealing Parameter Tuning

The first-stage SA heuristic is implemented with four parameters. In this section, we present results from tests performed for a large number of parameter combinations. The parameters' notation, description and tested values are listed in Table 8.3. The goal of performing this analysis is to find good parameter values to use when comparing the solution methods in Section 8.4, as well as for the analyses in Section 8.5.

Table 8.3: SA parameters with description and values to be investigated during the parameter tuning.

Parameter	**Description**	**Values**
T_0	Initial temperature	$10, 10^2, 10^3, 10^4$
α	Temperature decrease factor	$0.3, 0.6, 0.9$
I_{max}	Iterations between decreases	$25, 50$
T_{min}	Minimum temperature	$T * 10^{-1}, T * 10^{-2}$

Table 8.4 shows results for some of the most promising parameter combinations tested. Results for all combinations are in Table D.5 in Appendix D. The problem instance used for these tests has $|\mathcal{G}| = 9$ surgery groups, $N = 250$ scenarios, flexibility $F = 10\%$ and $\beta = 1.0$. For every combination of parameters, the SA-GCH-2SMIP is run 50 times, and statistics regarding the objective value of the best solution and last solution are presented. It should be mentioned that the best solution found in each run of Table 8.4 is not evaluated with the 2SMIP. This choice was made in order to save time, since we test 48 combinations of parameters 50 times each.

Table 8.4: Statistics for the most promising parameter combinations from Table 8.3 after 50 runs each with the SA-GCH-2SMIP (not using 2SMIP to evaluate the best solution). The table is an abbreviated version of Table D.5 in Appendix D. Notice that the table is color coded from green to red representing low to high values in each column, and that low values are preferred in all columns. The chosen parameter combinations are highlighted in bold.

Parameters				**Best Objective**				**End Objective**				***Time***	
T_0	α	I_{max}	T_{min}	$\hat{\mu}_b^o$	$\hat{\sigma}_b^o$	$L_b^{5\%}$	$U_b^{95\%}$	$\hat{\mu}_e^o$	$\hat{\sigma}_e^o$	$L_e^{5\%}$	$U_e^{95\%}$	μ_b^t	μ_r^t
10^2	*0.9*	*25*	10^{-1}	1608	186	1289	2009	1712	178	1396	2009	24	170
10^2	*0.9*	*25*	10^{-2}	1632	152	1345	1982	1695	148	1439	1982	26	325
10^2	*0.9*	*50*	10^{-1}	1540	138	1304	1928	1639	128	1430	1928	46	332
10^2	*0.9*	*50*	10^{-2}	1601	151	1302	2051	1691	163	1390	2051	39	644
10^3	*0.9*	*25*	10^{-1}	1537	90	1381	2366	1870	221	1416	2366	160	242
$\mathbf{10^3}$	***0.9***	***25***	$\mathbf{10^{-2}}$	**1483**	**91**	**1341**	**2347**	**1628**	**164**	**1364**	**2347**	**207**	**411**
10^3	*0.9*	*50*	10^{-1}	1483	83	1321	2292	1808	215	1380	2292	308	479
$\mathbf{10^3}$	***0.9***	***50***	$\mathbf{10^{-2}}$	**1467**	**76**	**1305**	**1975**	**1630**	**134**	**1341**	**1975**	**402**	**814**
10^4	*0.9*	*25*	10^{-1}	1794	124	1543	7983	4040	1530	1616	7983	30	258
10^4	*0.9*	*25*	10^{-2}	1523	94	1323	2380	1830	226	1371	2380	407	504
10^4	*0.9*	*50*	10^{-1}	1765	127	1491	6072	3580	1112	1962	6072	151	513
10^4	*0.9*	*50*	10^{-2}	1489	94	1307	2984	1835	281	1451	2984	824	993

$\hat{\mu}_b^o$, $\hat{\mu}_e^o$: sample average of best and end objectives, $\hat{\sigma}_b^o$, $\hat{\sigma}_e^o$: empirical standard deviation of best and end objectives, $\hat{\mu}_b^t$, $\hat{\mu}_r^t$: sample average of time to find best objective and of total runtime, $L_b^{5\%}$, $L_e^{5\%}$: lower 5%-ile of best and end objectives, $U_b^{95\%}$, $U_e^{95\%}$: Upper 95%-ile of best and end objectives

When choosing what combination of parameters to proceed with, we look at three statistics in particular. Perhaps most important is the expected objective value of the best solution found, $\hat{\mu}_b^o$. The SA heuristics are meant to find solutions that minimize the objective, and thus the parameter combination that yields the lowest $\hat{\mu}_b^o$ is a natural choice. If we in addition have a parameter combination that has little empirical standard deviation in the end objective, $\hat{\sigma}_b^o$, then we can feel more confident that it will consistently provide good results. The expected runtime of the SA heuristic is the last statistic we examine when choosing the parameters. The values in the rightmost column of Table 8.4 reveal $\hat{\mu}_r^t$, which is how long the SA heuristic took on average for the 50 runs with each parameter combination. Note that the lowest row in bold has the lowest $\hat{\mu}_b^o$, but also one of the longest expected runtimes, $\hat{\mu}_r^t$.

In the continuation of this computational study, we use one of two different parameter combinations. Which one depends on how the candidate first-stage solutions of the SA heuristic are evaluated. If they are evaluated heuristically, as in SA-GCH-2SMIP, the parameter values $\{T_0, \alpha, I_{max}, T_{min}\} = \{10^3, 0.9, 50, 0.01\}$ are used. This combination yielded the lowest $\hat{\sigma}_b^o$, and with a heuristic evaluation of first-stage solutions, the runtime is not an issue. For the SA-2SMIP, however, running the instance in Table 8.4 for the same combination of parameters takes more than 15 hours. Using the 2SMIP to evaluate first-stage solutions is much more time-consuming, and thus we settle for a slightly different combination. All else the same, letting $I_{max} = 25$ instead of 50, cuts the runtime in half. Comparing $\hat{\sigma}_b^o$ for the two combinations shows that we are not sacrificing much in terms of the standard deviation of the best solution we find either.

8.4 Comparing Solution Methods

In this section, we compare the performance of the different solution methods implemented. This involves a discussion around the trade-off between solution quality and solution time. We focus on the applicability of each solution method, which means the results presented concern the larger instances of our problem. The FMIP is used as a benchmark when evaluating the performance of the three heuristics. Section 8.4.1, Section 8.4.2 and Section 8.4.3 concern the performance the FMIP-2SMIP, SA-2SMIP and SA-GCH-2SMIP, respectively. In Section 8.4.4 we sum up by comparing the solution methods all-against-all.

As discussed in Section 8.1, the main drivers of complexity in our problem are problem size, flexibility level and capacity of bed wards downstream of the ORs. The solution methods are all tested on an instance with $|\mathcal{G}|= 9$ surgery groups, $N = 250$ scenarios, flexibility $F = 10\%$ and original bed ward capacity $\beta = 1.0$. This instance is chosen because we believe it demonstrates the potential of solution methods, and how they differ in performance. For simplicity, we refer to the instance as the test instance in the following subsections.

8.4.1 FMIP-2SMIP

The most basic of the heuristic solution methods is the FMIP-2SMIP. In Figure 8.5 we present the evolution of primal and dual bounds from a 12 hour run on the test instance. The dashed orange and blue lines show the best primal and dual bounds found after 12 hours by the FMIP, and serve as benchmarks for the heuristic solution methods to come.

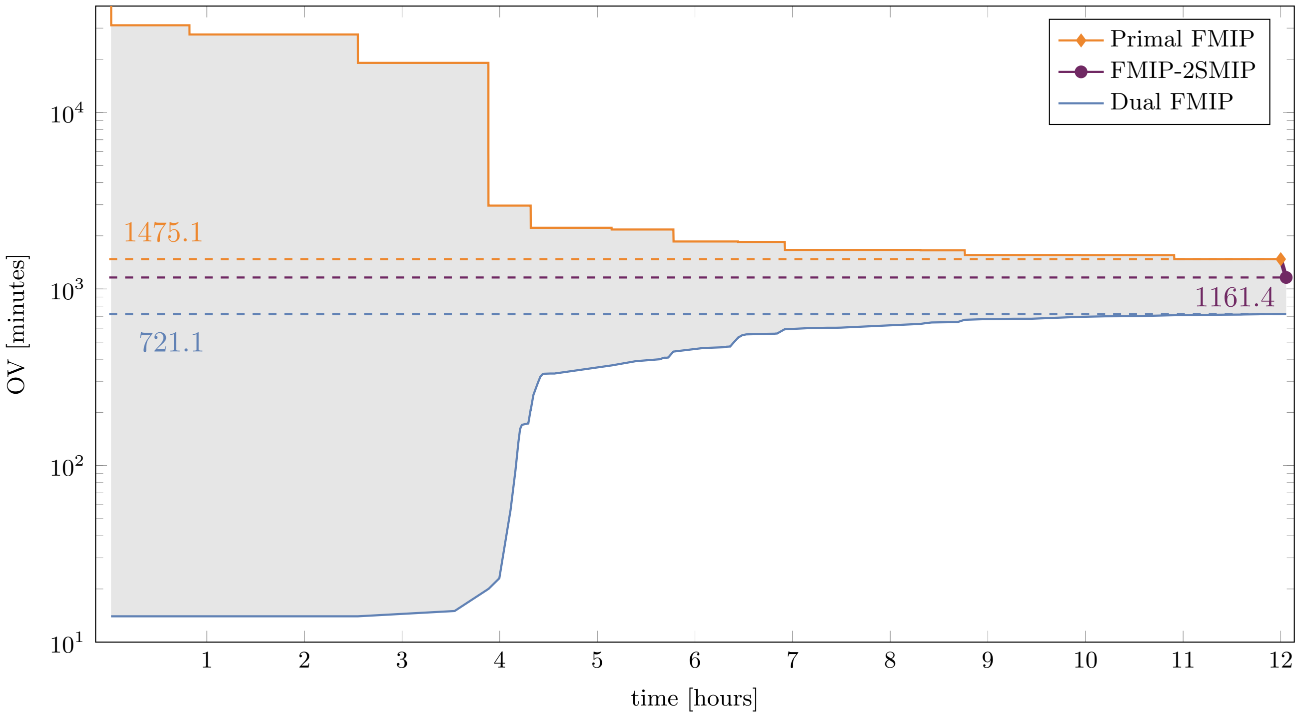


Figure 8.5: FMIP-2SMIP versus FMIP development over time. The problem instance has $|\mathcal{G}| = 9$ surgery groups, $N = 250$ scenarios generated from seed $m = 1$, initial bed ward capacity $\beta = 1.0$ and flexibility $F = 10\%$. The FMIP has been run for 12 hours followed by the 2SMIP for approximately 200 seconds. Notice that the vertical axis is logarithmic.

The FMIP-2SMIP is, as mentioned, an extension of the FMIP. We see 2SMIP improving the primal bound found by the FMIP in the small segment in time after the FMIP terminates, represented by the purple graph. Remember, the FMIP-2SMIP finds no other first-stage solutions than the FMIP, but it evaluates the best first-stage solution found to optimality. The improvement is quite significant.

Spending 200 seconds with the 2SMIP at the end of a 12-hour run, lowers the optimality gap from 51% to 38%. With this in mind, there is no reason to not extend the FMIP with the 2SMIP.

Looking at the primal bound of the FMIP as a function of time, the improvement diminishes quite heavily after approximately 5 hours. Cutting the FMIP at this point in time and using the 2SMIP would likely yield a decent primal bound in less time. Keep in mind, however, that the vertical axis in Figure 8.5 is logarithmic. As such, the primal bound after 5 hours of the FMIP is above 2000.

8.4.2 SA-2SMIP

The SA-2SMIP evaluates first-stage solutions by using the 2SMIP to solve the second-stage. The solid red graph in Figure 8.6 shows the objective value of the best solution found as a function of time for a run of the SA-2SMIP on the test instance. This value of the best solution found was referred to as $h(\varphi^G)$ in Algorithm 1. In Figure 8.7 we provide a more detailed view of the SA-2SMIP's run, where we also see the objective value of the best local solution as a function of time $h(\varphi^L)$. In the orange and blue graphs, we recognize the primal and dual bounds of the FMIP.

As we can tell from the primal and dual bounds in Figure 8.6, the FMIP struggles to provide even moderately well-performing solutions in the first four hours of running. After this, the optimality gap falls markedly, before closing at a declining rate during the last half of the runtime. Both Figure 8.6 and Figure 8.7 show the best run of SA-2SMIP, which was run ten times. Out of these ten runs, tabulated in Table D.1 Appendix D, the average best objective found was 1033.4 minutes, while the highest best objective was 1078.4. We see that SA-2SMIP starts at an initial solution that outperforms the best primal found by the FMIP after 12 hours. The development of the SA-2SMIP's run is difficult to see in the logarithmic scale of Figure 8.6. Nevertheless, the graph serves to show that the heuristic finds a good solution quickly, and that it is over the course of 9.5 hours able to improve the solution further. The duration of a run with the SA-2SMIP depends on how many moves are made and what first-stage solutions are being evaluated. The ten runs performed for this analysis ranged from 6 to 10 hours of runtime. Since the number of first-stage solutions evaluated during a run is fixed, these runtimes mean that it on average takes between 20 to 32 seconds to evaluate a first-stage solution.

Looking at the blue graph in Figure 8.7, we see how the heuristic explores more at the beginning where the temperature is higher, and varies less and less as the temperature decreases. An interesting observation is that most of the ten runs with SA-2SMIP did not converge to the best solution found when the algorithm neared the end. In fact, most of the runs found the best solution about halfway through the runtime, and converged to a local best solution with a slightly higher objective value. The run plotted in Figure 8.7 was in a good neighborhood of first-stage solutions in the final stages of the algorithm, and we see how the intensified local search at low temperatures pays off. All runs provide good solutions, but finding the very best solutions requires being in a good neighborhood near the end.

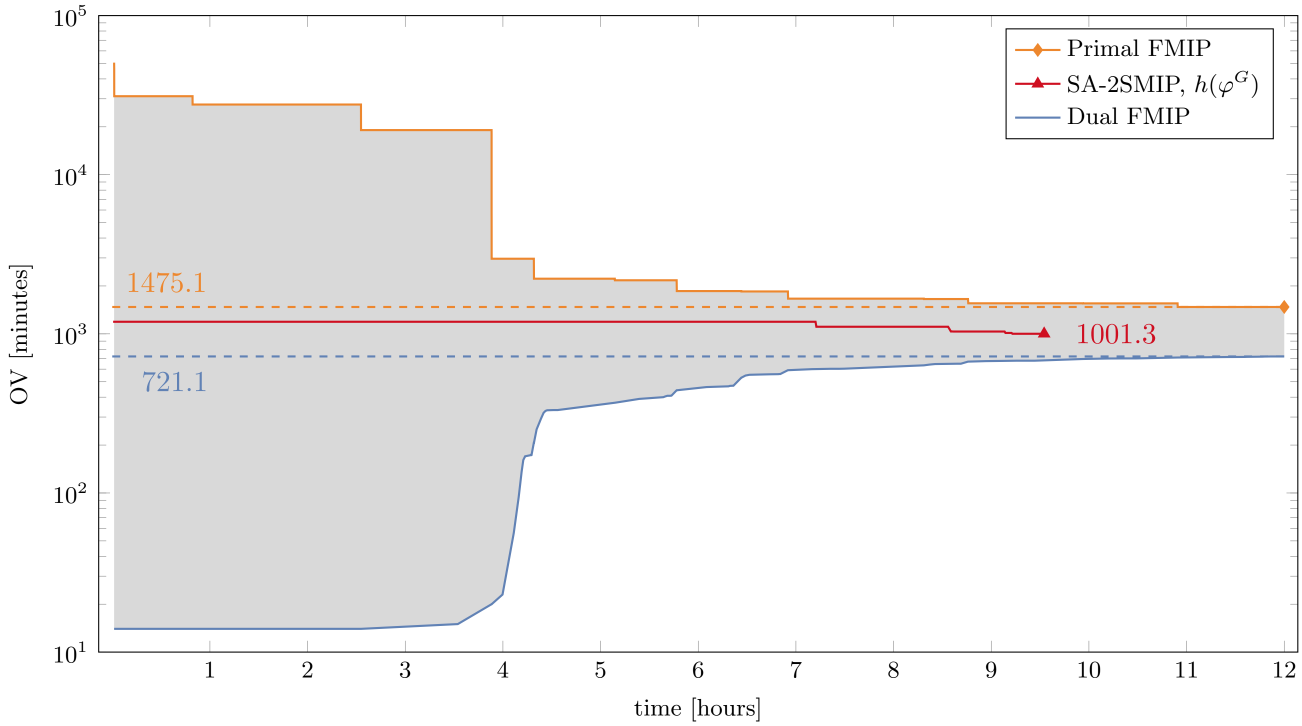


Figure 8.6: SA-2SMIP versus FMIP development over time. The problem instance has $|\mathcal{G}| = 9$ surgery groups, $N = 250$ scenarios generated from seed $m = 1$, initial bed ward capacity $\beta = 1.0$ and flexibility $F = 10\%$. The FMIP has been run for 12 hours. The SA-2SMIP has been run 10 times, and the best run is plotted. The other runs are tabulated in Appendix F in Table D.1. Notice that the vertical axis is logarithmic.

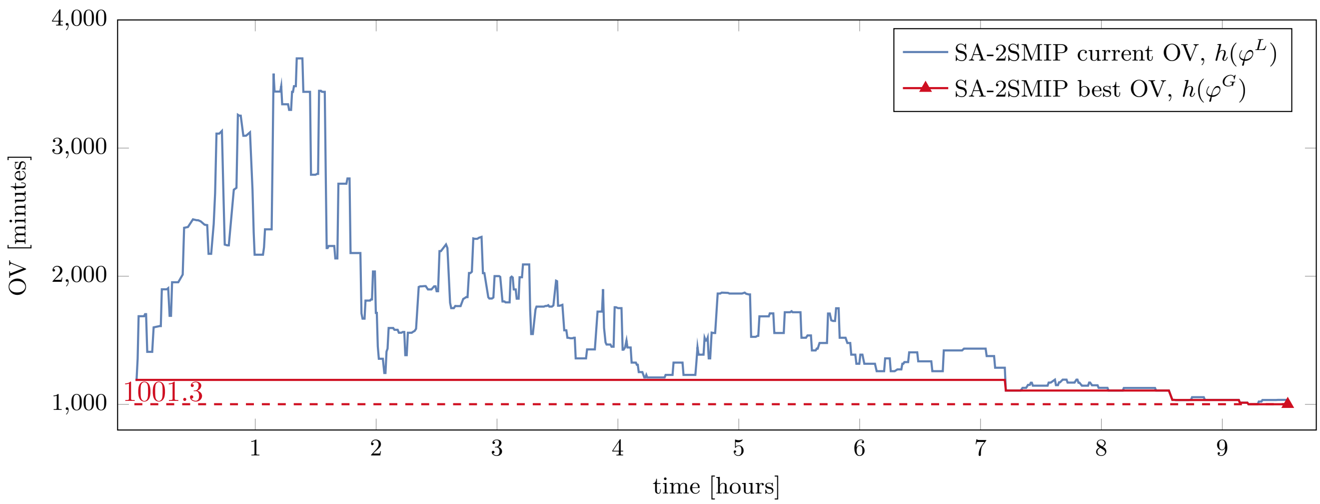


Figure 8.7: One of the lowest objective SA-2SMIP runs out of the 10 runs tabulated in Appendix D, Table D.1 for $|\mathcal{G}| = 9$ groups, a scenario tree $\breve{\boldsymbol{\xi}}^1_{250}$ of size $N = 250$ generated from seed $m = 1$, initial bed ward capacity $\beta = 1.0$ and flexibility $F = 10\%$.

8.4.3 SA-GCH-2SMIP

SA-GCH-2SMIP evaluates candidate first-stage solutions heuristically. However, the 2SMIP is used to evaluate the best solution found during runtime. As such, both SA-GCH-2SMIP and SA-2SMIP need to be able to solve the second-stage more or less to optimality for some first-stage solution. Naturally, the time it takes to solve the second-stage with the 2SMIP is a lot more important for SA-2SMIP, since this is done in every iteration. In any case, the dependency on the 2SMIP makes SA-GCH-2SMIP most interesting for problem instances where it can serve as a faster alternative to the SA-2SMIP. In Figure 8.8, we see SA-GCH-2SMIP's performance on the test instance.

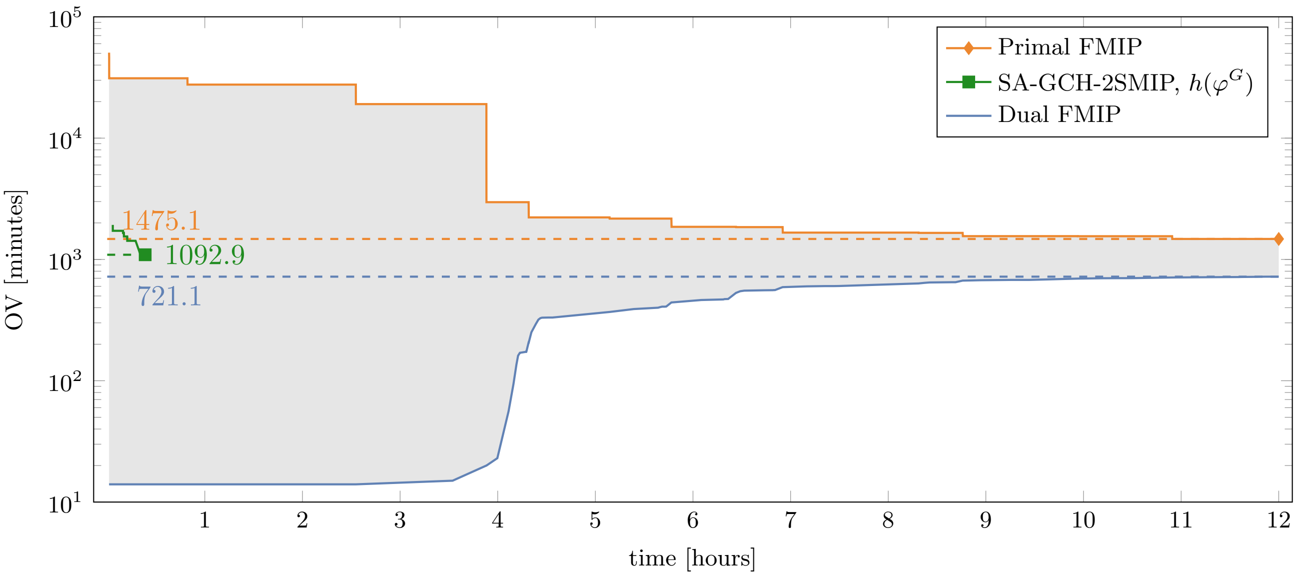


Figure 8.8: SA-GCH-2SMIP versus FMIP development over time. The problem instance has $|\mathcal{G}| = 9$ surgery groups, $N = 250$ scenarios generated from seed $m = 1$, initial bed ward capacity $\beta = 1.0$ and flexibility $F = 10\%$. The FMIP has been run for 12 hours. The SA-GCH-2SMIP has been run 10 times, and the best run is plotted. Results for all the runs are tabulated in Appendix D in Table D.2. Notice that the vertical axis is logarithmic.

As in Section 8.4.2, SA-GCH-2SMIP has been run ten times. In these ten runs, the heuristic on average yielded a best objective value of 1152.0, while the highest best objective value was 1176.8. On the test instance, the SA-GCH-2SMIP takes less than 20 minutes on a run. This makes it possible to run the heuristic several times and pick the best solution, and still spend less time than some of the other solution methods. We see in Figure 8.9 how the SA-GCH-2SMIP explores and exploits during the search. Remember that this heuristic evaluates twice as many candidate first-stage solutions at each temperature level compared to the SA-2SMIP. The linear downward sloping sections at the end of the graphs are where the 2SMIP evaluates the best first-stage solution.

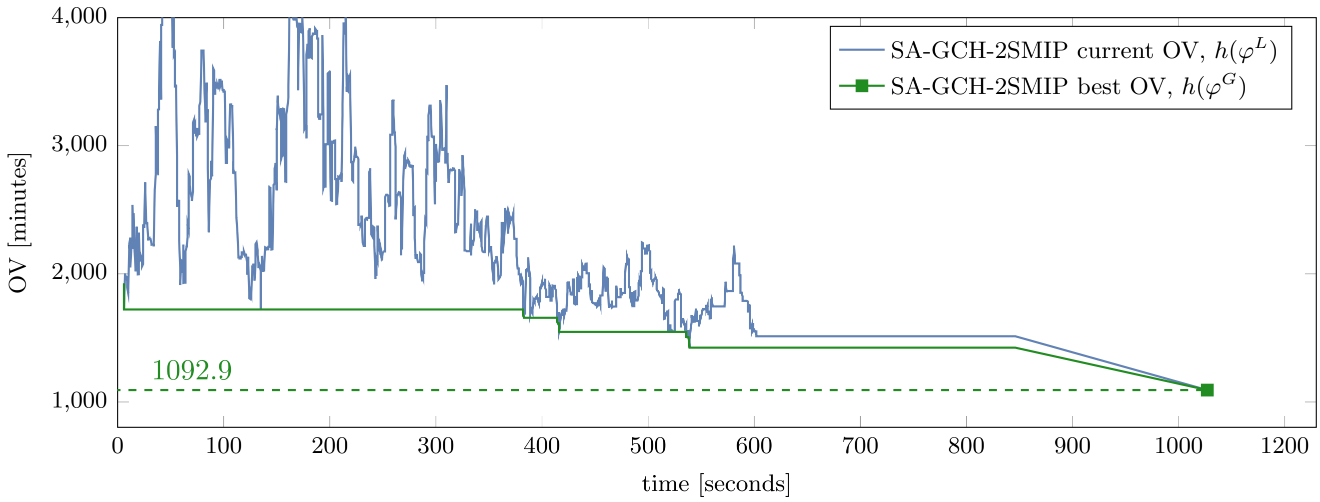


Figure 8.9: The lowest objective SA-GCH-2SMIP run out of the ten tabulated runs in Appendix D, Table D.2. The problem instance has $|\mathcal{G}| = 9$ surgery groups, $N = 250$ scenarios generated from seed $m = 1$, initial bed ward capacity $\beta = 1.0$ and flexibility $F = 10\%$.

8.4.4 Comparing Solution Methods Summary

To summarize the performance of the four solution methods, we present all results from the test instance in Figure 8.10. Looking at the dashed lines from top to bottom, we see that the primal of the FMIP is the highest objective value of the solution methods. The FMIP-2SMIP provides a noticeably better objective value, but this is the same first-stage solution as the FMIP finds.

Both of the SA heuristics find better solutions than the FMIP/FMIP-2SMIP. We see that the SA-GCH-2SMIP can serve as a faster alternative to the SA-2SMIP, albeit with slightly worse performing solutions. Which one is preferable depends on how important solution quality is relative to runtime. If the goal is to use the resulting first-stage solution at a real-life surgical clinic, then the runtime of 6-10 hours for the SA-2SMIP is probably not a problem. After all, an MSS is typically used for 6-12 months before being revised. However, if the goal is to perform analyses on several instances, then the faster SA-GCH-2SMIP can give good solutions in far less time.

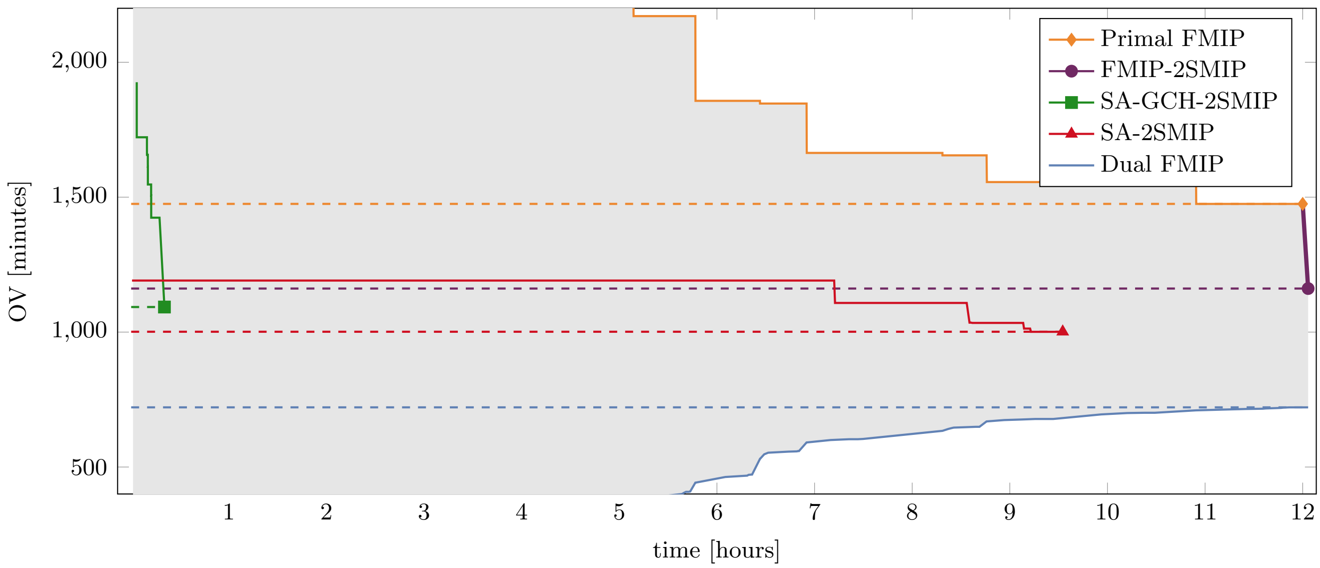


Figure 8.10: Comparison of solution methods' development over time. The problem instance has $|\mathcal{G}| = 9$ surgery groups, $N = 250$ scenarios generated from seed $m = 1$, initial bed ward capacity $\beta = 1.0$ and flexibility $F = 10\%$. The results shown are the same runs as presented in Figure 8.5-8.9.

8.4.5 Comparing Solution Methods on a Larger Instance of the Problem

To round off the comparison of solution methods, we test SA-GCH-2SMIP on an instance where SA-2SMIP, for all intents and purposes, is too slow. In Section 8.2, we learned that for instances with $|\mathcal{G}|= 25$ surgery groups, scenario trees should be of size $N = 600$ in order to properly reflect the underlying demand distribution. Figure 8.11 shows a 12 hour run of the FMIP compared to SA-GCH-2SMIP. As we can see, the FMIP terminates with an optimality gap of almost 99%. Extending the FMIP with the 2SMIP does help decrease the primal bound significantly, but the optimality gap remains high at 96%.

SA-GCH-2SMIP spends just shy of two and a half hours searching through different first-stage solutions, while the last hour and a half of its runtime is spent evaluating the best first-stage solution with the 2SMIP. With the dual bound from the FMIP being so low, the solution found by the SA-GCH-2SMIP still gives an optimality gap of 75%. Nevertheless, this solution is obviously far superior to that of the FMIP-2SMIP.

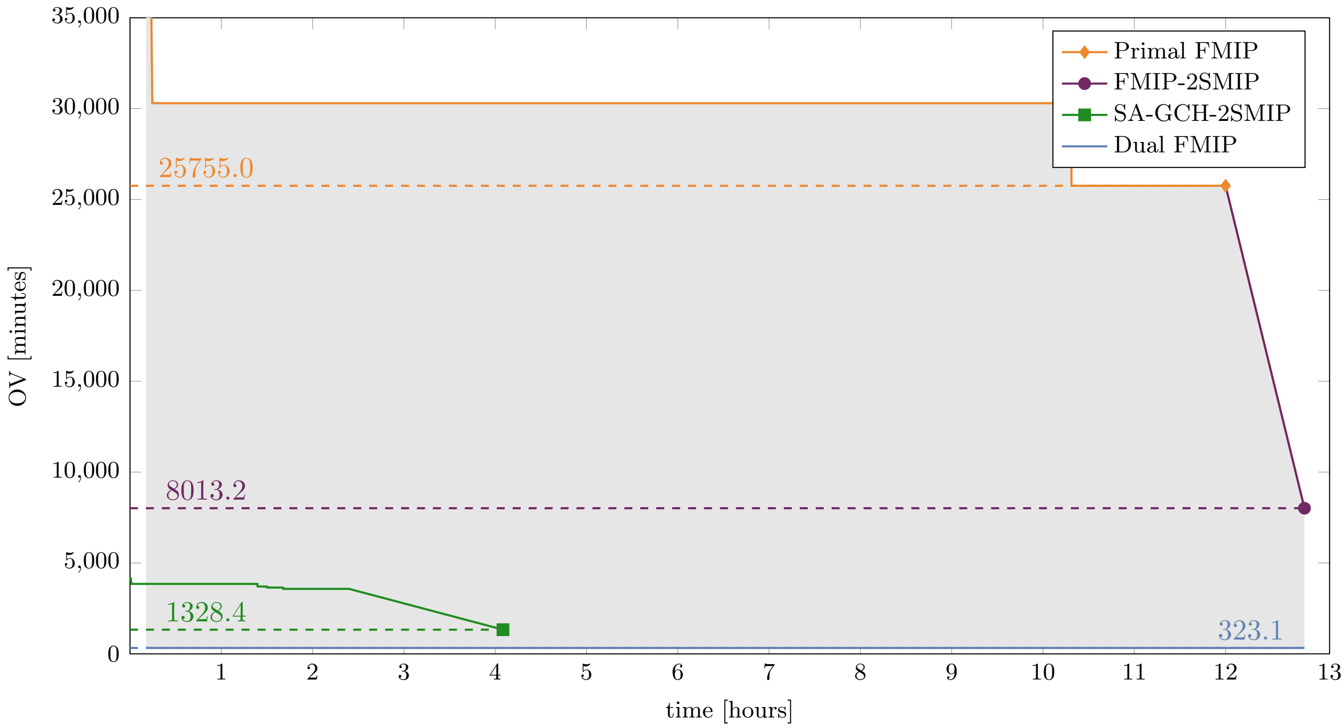


Figure 8.11: Solution metods' development over time for a large instance of the problem. The problem is solved for $|\mathcal{G}| = 25$ groups, a scenario tree $\breve{\boldsymbol{\xi}}^{1}_{600}$ of size $N = 600$ generated from seed $m = 1$, initial bed ward capacity $\beta = 1.0$ and flexibility $F = 10\%$. The FMIP had a time limit of 12 hours, and the 2SMIP in both FMIP-2SMIP and SA-GCH-2SMIP had an optimality gap cutoff on 1%.

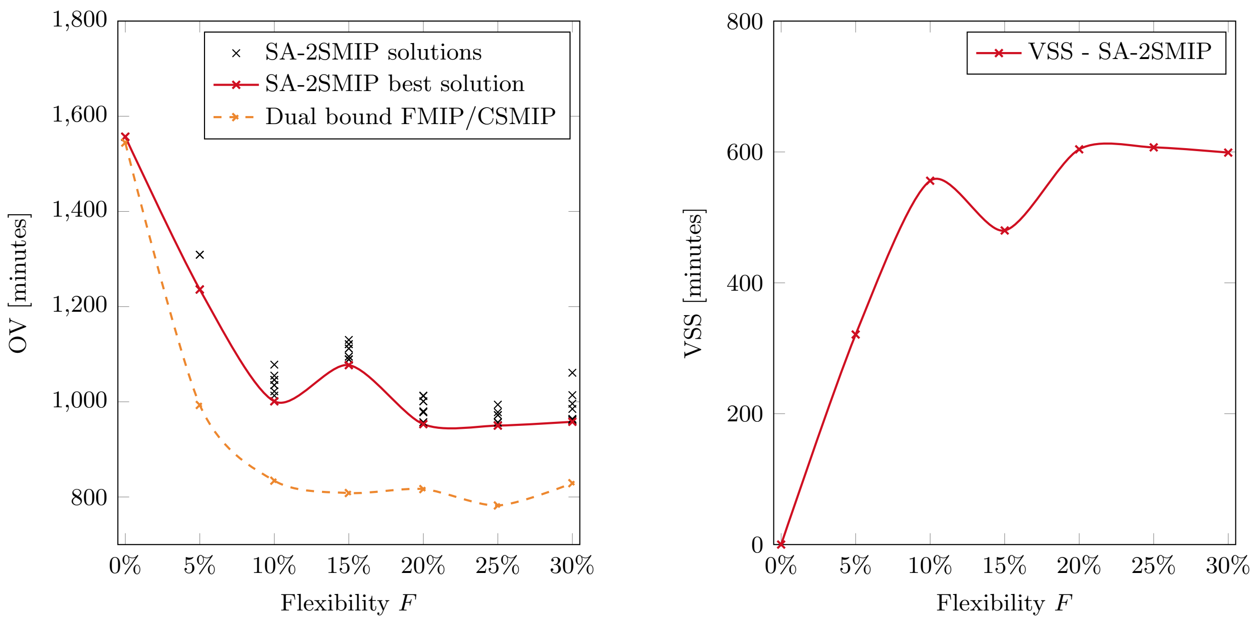
Considering SA-GCH-2SMIP spent 1.5 hours evaluating the best first-stage solution with 2SMIP before reaching an optimality gap below 1%, using SA-2SMIP is not an option for this instance. For the much smaller instance tested in Section 8.4.2, SA-2SMIP spent on average 20-32 seconds evaluating a first-stage solution to optimality, and that resulted in runs of 6-10 hours. If each iteration instead takes 1.5 hours, SA-2SMIP is not a suitable solution method.

8.5 Value of Flexibility

While the focus until now has been on the technical aspects of the optimization problem, we spend this last section investigating the practical implications of flexibility in the MSSP. More concretely, we apply the solution methods implemented in order to estimate how and to what degree flexible slots in an MSS can increase the throughput of patients at the Clinic of Surgery at St. Olavs Hospital.

8.5.1 Value of Flexible Slots with Original Bed Ward Capacity

Figure 8.12a and 8.12b show results from runs of SA-2SMIP on instances with $|\mathcal{G}|= 9$ surgery groups, $N = 250$ scenarios, original bed ward capacities $\beta = 1.0$ and a range of flexibility levels F. For every flexibility level $F \in \{0\%, 5\%, \ldots, 30\%\}$, the SA-2SMIP is run ten times, and the best objective found in each run is plotted as a cross in Figure 8.12a.



(a) Objective value

(b) VSS

Figure 8.12: Value of flexibility illustrated in terms of objective value in (a) and VSS in (b). The red graph in (a) is interpolated thorugh the best solutions obtained by SA-2SMIP for instances with $|\mathcal{G}|= 9$ surgery groups, a scenario tree $\tilde{\boldsymbol{\xi}}^1_{250}$ of size $N = 250$ generated from seed $m = 1$, original bed ward capacity $\beta = 1.0$ and different flexibility levels F. The lowest objective value at each level F is used for calculating the VSS in (b). The dual bounds in (a) are the best obtained dual for each flexibility level found by the FMIP or CSMIP. Values are tabulated in Table D.3-D.4 and Table C.3-C.4 in the appendices.

A red line is interpolated between the best solutions at every flexibility level in Figure 8.12a, displaying how the objective value changes as a function of flexibility. The trend line suggests that the objective value improves when increasing the level of flexibility up until 10%. Beyond this, there is no clear benefit of added flexibility. The orange dashed line displays the best dual bounds found by running the FMIP from Chapter 5 and the alternative cutting stock MIP (CSMIP) from Appendix B for 12 hours. While the FMIP from Chapter 5 generally finds better primal solutions, the CSMIP has proven to often find better dual bounds in the same amount of time. Runs of the two model formulations are compared in Appendix C. The dual bounds give credibility to the general trend of the best solutions from the SA-2SMIP. We can tell that for $F = 0\%$, the SA-2SMIP has found very near-optimal solutions. This, in combination with the fact that the dual bounds are lower for higher levels of flexibility, means that the value of flexibility can potentially be even higher than what the red line in (b) shows.

The initial benefit of allowing for some flexible slots is quite intuitive, because it allows for supply to adapt to changes in demand by assigning flexible slots to the specialties that need it the most in each scenario. Understanding why the incremental benefit diminishes from a flexibility level of 10% and beyond, is not as intuitive. One possible explanation could be that there is simply not enough variance in the demand for individual surgery groups. Perhaps 10% flexibility is enough to shift supply so that we rarely observe a scenario where a slot is not packed with patients. The fact that the objective function does not converge to zero as the flexibility increases, can be a bit confusing. Since we are measuring unmet demand, it would seem as if the queues of patients grow bigger over time if we expect 1000 minutes of unmet demand in an average planning period, even with higher levels of flexibility. This could be a result of some scenarios where the demand for many surgery groups is higher than the average, so that it is not possible to perform all surgeries in those scenarios. While in reality this backlog would at some point be handled in periods with lower demand, our model does not allow for taking on backlog patients in scenarios with low demand.

In Figure 8.12b the value of the stochastic solution (VSS) for the best solutions in Figure 8.12a is plotted. Since our objective function measures the expected number of unoperated minutes, the VSS measures how many minutes we expect to reduce unmet demand by over a planning period. In other words, the VSS measures the value of using the stochastic solution as opposed to the deterministic EVS.

One reason that the VSS is so low in the case of no flexibility, could be the fact that the bed ward capacity downstream is basically non-binding in Figure 8.12. We hypothesize that with idle capacity in the bed wards downstream, it is not significant when each surgery is performed, so long as it is performed during the planning period. Therefore, the objective function value is mainly dependent on how many slots each specialty is assigned and not when these slots are placed in time. This would explain why the stochastic solution (SS) and EVS perform the same. The EVS is the solution to the deterministic problem where all surgery groups have demand equal to their expected demand. We would expect the SS to assign approximately the same number of slots to each specialty as the EVS, since the scenarios on average reflect the expected demand of each surgery group.

Despite the VSS being zero when we do not incorporate flexibility, the VSS increases as we increase the flexibility. The shape of the graph in Figure 8.12b is closely related to the shape of the graph in Figure 8.12a. The increase in VSS when increasing flexibility from 0% to 10% is the same as the reduction in the best objective of Figure 8.12a when making the same jump in flexibility. It is fair to assume that the increasing VSS is a result of the SS having the advantage of being able to use flexible slots as recourse actions. The deterministic EVSs, on the other hand, contain only fixed slots, and therefore cannot adapt supply to demand.

8.5.2 Comparing Value of Flexibility at Different Levels of Bed Ward Capacity

In this section, we investigate how introducing flexible slots affects the objective function value at three different levels of bed ward capacity. These are the same levels that were presented in Section 8.1, namely $\beta \in \{1.0, 0.6, 0.4\}$. These levels are intended to represent instances where the capacity is large, medium and small. They result in a weekday utilization at the bed wards of $\approx 45\%$, $\approx 80\%$ and $\approx 100\%$ for the instances tested in this section. The number of surgery groups remains $|\mathcal{G}|= 9$.

The number of scenarios needs to be reduced in this part of the analysis compared to Section 8.5.1, because of the added complexity of the reduced bed ward capacity, discussed in Section 8.1. Instances with little capacity in bed wards have proven difficult for the SA heuristics to handle. To get comparable results across different values of β, we will proceed to use the SAA method presented in Section 3.4.3.

As time is a limited resource for this analysis, we choose to test more instances rather than increasing the run time and maintaining large scenario trees. She SAA problems z_N^m in (8.5) are solved for $N = 10$ and $m \in \{1, 2, \dots 30\}$ at different levels of flexibility F and bed ward capacity β. Even with the number of scenarios reduced to $N = 10$, however, we experience SAA problems where we are unable to get tight optimality gaps. Therefore, we use the extended FMIP-2SMIP in order to obtain better primal bounds.

First, we run the FMIP for 1200 seconds, before fixing the first-stage solution and running the 2SMIP for 120 seconds. The method is as follows for each SAA problem:

1. Run the FMIP model for 1200 seconds

$$z_N^m = \min_{\boldsymbol{\phi} \in \Phi} \hat{f}(\boldsymbol{\phi}; \breve{\boldsymbol{\xi}}_N^m) \tag{8.1}$$

$$\hat{f}(\boldsymbol{\phi}; \breve{\boldsymbol{\xi}}_N^m) := \frac{1}{N} \sum_{n=1}^{N} F(\boldsymbol{\phi}; \breve{\xi}_{n,N}^m) \tag{8.2}$$

2. Fix the obtained first stage solution and run it again for 120 seconds

$$z_N^{m,fix} = \min_{\boldsymbol{\phi} \in \Phi} \hat{f}(\boldsymbol{\phi}; \boldsymbol{\varphi}_N^{m*}, \breve{\boldsymbol{\xi}}_N^m) \tag{8.3}$$

$$\boldsymbol{\varphi}_N^{m*} := \Big\{ \{\boldsymbol{\gamma}, \boldsymbol{\lambda}\} : \operatorname*{argmin}_{\boldsymbol{\phi} \in \Phi} \hat{f}(\boldsymbol{\phi}; \breve{\boldsymbol{\xi}}_N^m) \Big\} \tag{8.4}$$

3. Calculate new primal bound

$$z_N^m = \min(z_N^{m,fix}, z_N^m) \tag{8.5}$$

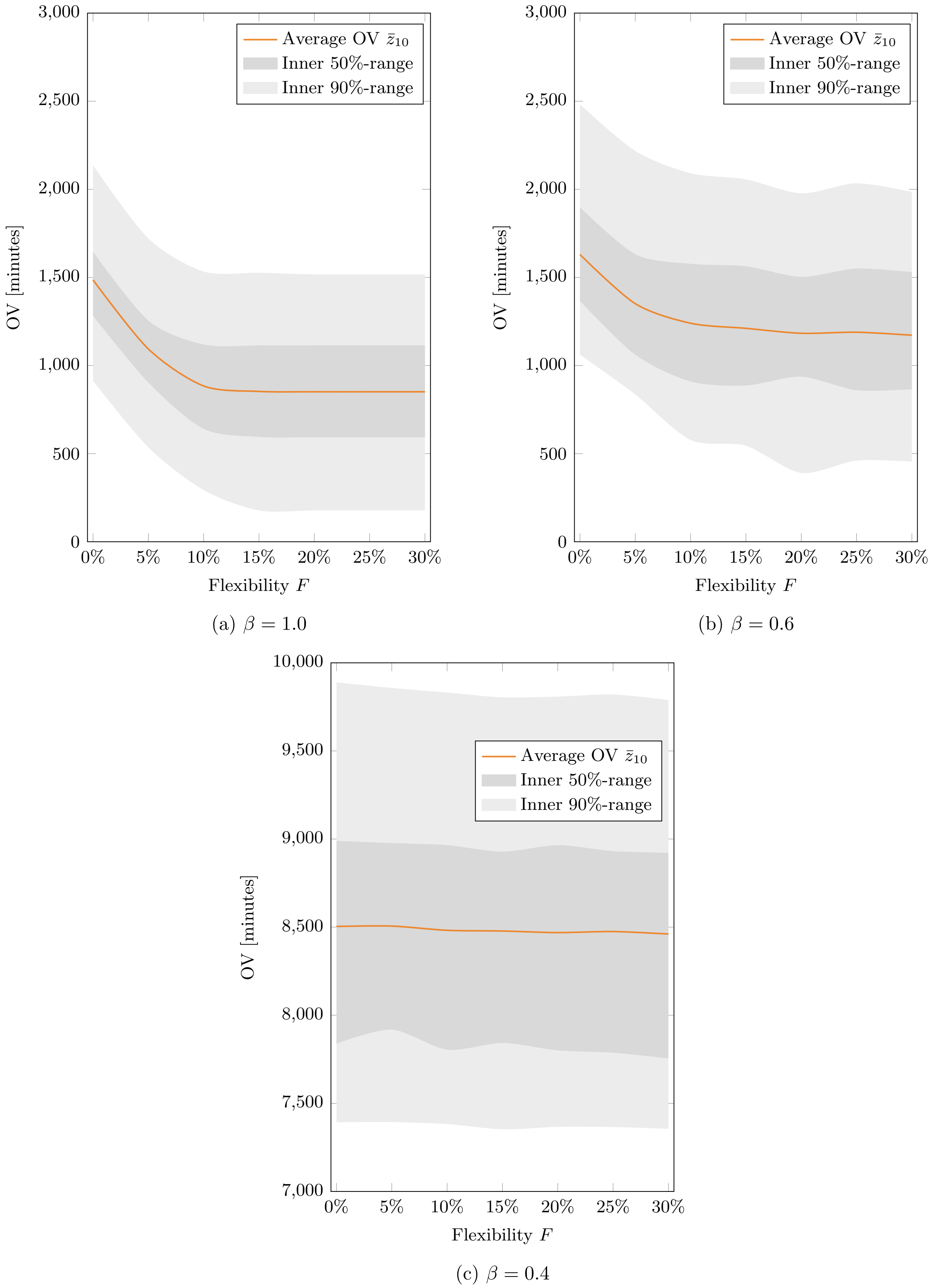
Here $\hat{f}$ is the overall objective function. $F(\boldsymbol{\phi}; \breve{\xi}_{n,N}^m)$ is the objective function value in the n-th scenario, $\breve{\xi}_{n,N}^m$, of the scenario tree, $\breve{\boldsymbol{\xi}}_N^m$, of size N and sample m. $\boldsymbol{\phi}$ is the set of decision variables, $\{\boldsymbol{\gamma}, \boldsymbol{\lambda}, \boldsymbol{\delta}, \boldsymbol{x}\}$, $\boldsymbol{\varphi}$ are the first stage decision variables $\{\boldsymbol{\gamma}, \boldsymbol{\lambda}\}$ and Φ is the feasible set of $\boldsymbol{\phi}$, defined in the mathematical model in Chapter 5. In our formulation, a scenario $\breve{\xi}_{n,N}^m$ is a vector of $\{Q_{1,n}, \dots Q_{|\mathcal{G}|,n}\}$, where $Q_{g,n}$ is the number of patients from group g in line for surgery in scenario n.

Figure 8.13 displays results for the three different bed ward capacities. Looking at (a), we see the same trend and similar objective function values as in Figure 8.12 of Section 8.5.1. This gives credibility to the results of the SAA method used in Figure 8.13.

As β decreases to 0.6, we see the trend line become less steep, and at 0.4, there is no clear benefit to incorporating flexibility at all. Comparing the highest and lowest value of β exemplifies the impact of capacity constraints downstream of the ORs. At $\beta = 1.0$ we are able to reap the full benefit of flexibility, since there is seldom a problem with bed wards being full. When $\beta = 0.4$, however, the bed wards act as bottlenecks downstream of the ORs. As such, it does not matter whether or not a slot is flexible or fixed. Even with a completely fixed schedule at $F = 0$ %, the model seems to be able to make the most out of the downstream bed ward capacity. It is obvious that the utilization of ORs is far lower for $\beta = 0.4$ than for higher values, since the objective function value is far higher. This means that fewer surgeries are being planned than for the other two values of β. Taking this into consideration, $\beta = 0.4$ probably represents an unrealistically small capacity of bed wards relative to the OR capacity in our instances.

To verify the analysis above, we illustrate the occupation of bed wards for each value of β in Figure 8.14. Each subfigure shows the average number of occupied beds at the MC ward and IC ward relative to their capacity. The occupation is the average of solutions across all flexibility levels F and scenario trees m solved in Figure 8.13. As is evident in Figure 8.14a, there is an abundance of capacity in the MC ward, while the occupation is close to the capacity in the IC ward on Fridays.

As β decreases in Figure 8.14b and c, the occupation line progressively comes closer to the capacity at both wards. An interesting observation is that the number of unoccupied beds is greatest during weekends. The capacity of both wards drops to a lower level on Fridays, and stays at this level throughout each weekend. This is arguably not the best way to distribute the capacity, however. Since surgeries are only performed on weekdays, there is no way to make use of the idle capacity at wards on Saturdays and Sundays. Surgeries have to be planned such that the wards are not overfilled on Friday night, but this will inherently result in idle capacity on the two days that follow. In terms of patient throughput, moving some capacity from Sundays to Fridays would likely allow for higher utilization of bed wards. We will not pursue this topic any further in this thesis.

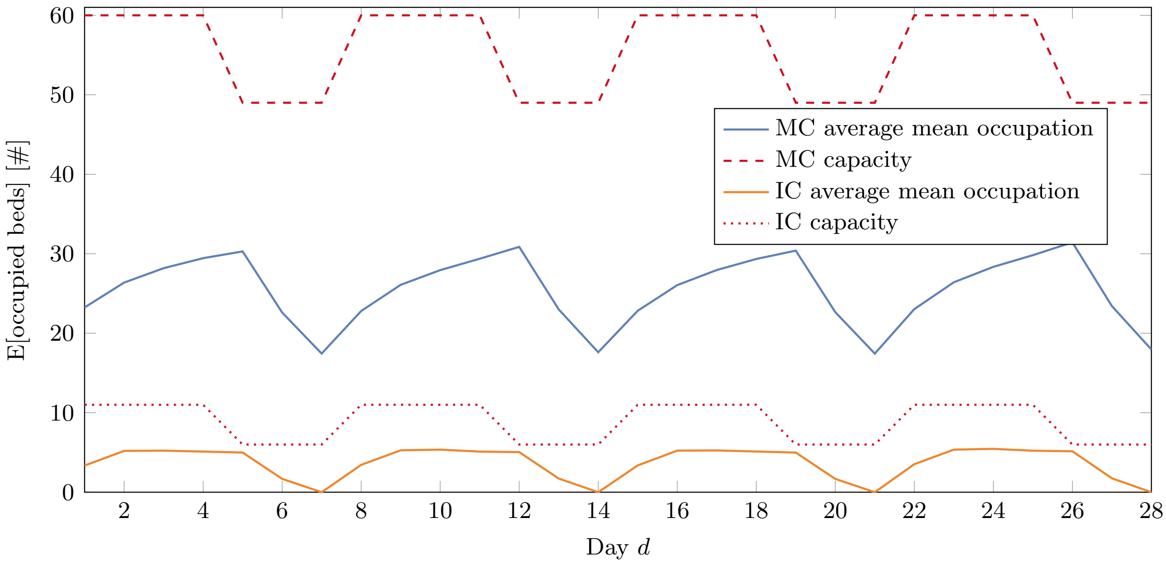


(a) $\beta = 1.0$

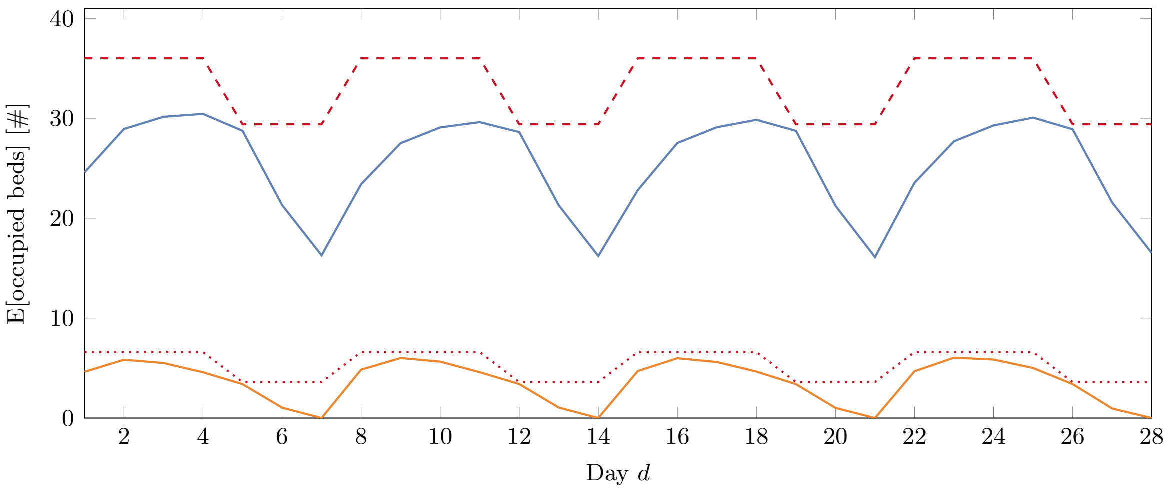
(b) $\beta = 0.6$

(c) $\beta = 0.4$

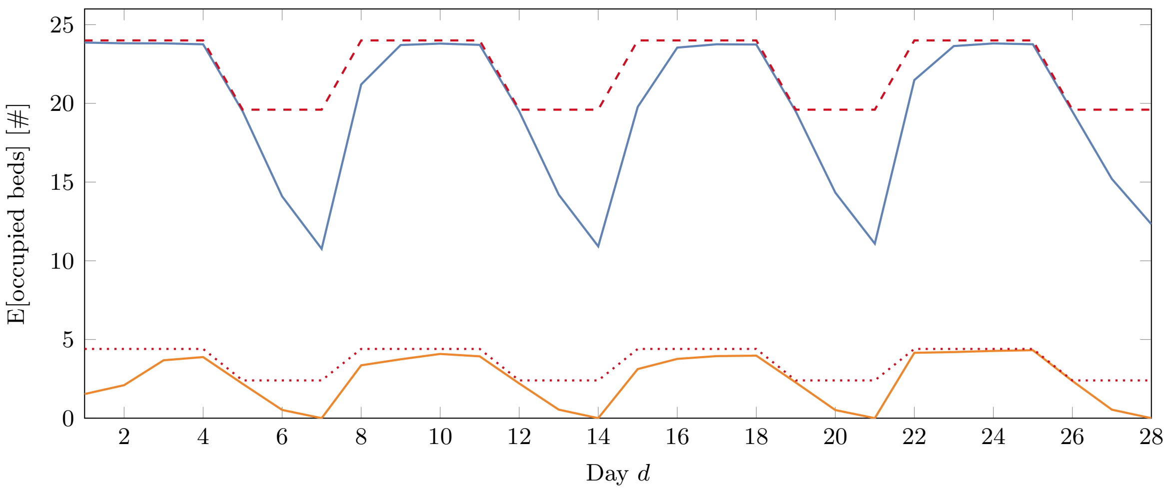
Figure 8.13: The plots show the relation between percentage of flexible slots during a planning period F and the objective value of the SAA problems z_{10}^m from Equation 8.5. The SAA problems are solved for $M = 30$ different scenario trees $\breve{\boldsymbol{\xi}}_{10}^m$ of size $N = 10$ generated from seed $m \in \{1, 2, \ldots, 30\}$ at three different levels of bed ward capacities $\beta = \{1.0, 0.6, 0.4\}$ for each flexibility level F.



(a) $\beta = 1.0$



(b) $\beta = 0.6$



(c) $\beta = 0.4$

Figure 8.14: The plots show the average bed occupancy at both wards for three different bed ward capacities $\beta \in \{1.0, 0.6, 0.4\}$. The plots are based on the results from the same SAA problems as in Figure 8.13. The SAA problems are solved for $M = 30$ different scenario trees $\breve{\boldsymbol{\xi}}_{10}^{m}$ of size $N = 10$ generated from seed $m \in \{1, 2, \ldots, 30\}$ at three different levels of bed ward capacities $\beta = \{1.0, 0.6, 0.4\}$ for each flexibility level F.

8.6 Limitations of the Study and Future Research

To finish Chapter 8, we highlight the limitations of the study, and point to some interesting topics for future research. To give context to this discussion, we reiterate the overall purpose of this thesis, which is to find both the value of taking uncertainty in patient demand into account and the value of incorporating flexibility when assigning ORs in surgical scheduling.

We have reason to believe that the input data used in the implementation of our model does not properly reflect the reality at the Clinic of Surgery at St. Olav. The data set mentioned in Section 7.1.1 was not straightforward to process in order to fit the needs of our model, and several assumptions have been made during implementation. The most significant one is perhaps our assumption that the demands for different surgery groups follow Poisson distributions and are independent of each other. Even though we have used the average throughput of patients in 2019 as our mean demand, we have also assumed that the variance of each surgery group's demand is equal to its mean. This last assumption is made for practical reasons and due to a lack of data. If the demand of each surgery group follows a Poisson distribution, we have indirectly assumed that the time between arrivals of patients in the queue is exponentially distributed. In reality, a patient typically consults their doctor before being referred to a specialist at the hospital, who decides if this patient should be put in line for surgery. Whether or not the time between each patient a specialist puts in line for surgery is exponentially distributed is hard to tell from the 2019 data set. As a result, our model may be of limited use as a planning tool for the Clinic of Surgery at St. Olav.

A topic that has been discussed a lot throughout Chapter 8, is the capacity of bed wards. As we have seen, the complexity of our problem is highly dependent on this capacity. We choose to model the occupation of bed wards as hard constraints in our optimization model, but others in the literature have done so using soft constraints. Our choice to merge the different MC and IC wards into one MC ward and one IC ward, has been an attempt to relax the bed ward capacity constraints somewhat. However, letting these constraints be soft in the objective function could allow for interesting analyses too. This could potentially help tackle the abrupt reduction in the capacity of bed wards that we have on weekends in our test instances.

In this thesis, we define value through an objective function that measures the amount of unmet demand. As we have seen in Section 8.5, we find value in flexibility in OR scheduling primarily when there is sufficient capacity in downstream bed wards. Other definitions of value could, however, yield other results. While flexibility may not reduce unmet demand when there is little capacity in bed wards, it could provide other benefits. An example of such benefits is fairness in patient waiting time. Guaranteeing a maximum waiting time before surgery could give patients more predictability and a sense of fairness. Thus, if the objective function minimized the maximum waiting time of patients in line for surgery, this could make flexibility valuable.

A large part of this thesis concerns solution methods for our stochastic two-stage model, and in Section 8.4, we saw the two SA heuristics deliver solid results. The SA-2SMIP found the best solutions, but is significantly more time-consuming than the SA-GCH-2SMIP. An interesting path for future research could therefore be to further develop the heuristic evaluation of first-stage solutions in SA-GCH-2SMIP. For example, extending the GCH with an improvement heuristic could give closer-to-optimal second-stage solutions. This would probably be more time-consuming, but still less so than the 2SMIP. Another way to improve the SA heuristics could be through breaking symmetry in first-stage solutions. The SA heuristics currently evaluate a number of first-stage solutions that are practically identical. Breaking symmetry in the first-stage could therefore make both of the SA heuristics more efficient, and more likely to end up near the optimal solution.

The main challenge of our solution methods is to find good solutions when the bed ward capacity is low. Consequently, alternative solution methods are also interesting to investigate. The structure of the problem could make decomposition methods well suited. For instance, the scheduling could be split into a master problem, where fixed slots are assigned, and subproblems, where flexible slots are assigned and surgeries planned.

Chapter 9

Concluding Remarks

In order to meet the steady increase in hospital patient demand in Norway and many other countries, we must better exploit the available resources. One way of doing so is through efficient surgical scheduling. In this report, we have developed a two-stage stochastic model that outputs a modified MSS. We do this in an attempt to reap the benefits of both a cyclic surgical schedule and the benefits of flexibility. The modified MSS allows us to incorporate flexibility in the surgical schedule in the form of flexible slots. As opposed to fixed slots, flexible slots are assigned to specialties in the second-stage, when patient demand is known. By varying the input data, we were mainly interested in two things. First, we wanted to find the value of taking uncertainty in patient demand into account. Second, we looked at the value of incorporating flexibility in surgery scheduling. Due to the stochastic optimization problem being computationally demanding, we have developed three heuristic solution methods. The first, FMIP-2SMIP, is a simple matheuristic that fixes the best first-stage solution found with a mixed integer program (FMIP) after a given time, and then optimizes the second-stage problem (2SMIP). The second, SA-2SMIP, combines simulated annealing (SA) in the first-stage with an exact method (2SMIP) in the second-stage. The third, SA-GCH-2SMIP, combines SA in the first-stage with a greedy construction heuristic (GCH) in the second-stage, before optimizing the best first-stage solution found at the end with an exact method (2SMIP). In the order presented, the heuristics become increasingly heuristic, but also more time-efficient. This makes them appropriate for different problem instances. In creating these heuristic solution methods, we contribute to the literature on efficient tactical surgical scheduling and create a prototype for a planning tool at surgical clinics.

We were not able to detect any value in planning for uncertainty in patient demand without incorporating flexibility. However, allowing for flexible slots reduces the expected unmet demand, as long as bed wards are not bottlenecks. With our test instances, the value of flexibility stagnates at around 10% flexibility. At this level of flexibility, we saw a reduction of 35.7% in unmet demand compared to the solution with no flexibility. On average, this amounts to 556 surgery minutes during a four-week planning period, with our input data. We deduce that the value of flexibility stagnates when the level of flexibility outgrows the variance in patient demand. Furthermore, the flexibility in the assignment of ORs requires sufficient capacity in downstream resources in order to reduce unmet demand.

In conclusion, planning for uncertainty can help reduce unmet demand if also incorporating flexibility in an MSS. We can not rule out the possibility that other definitions of value would have revealed additional benefits. We hope that this outcome incentivizes other researchers to incorporate flexibility when striving to make surgical scheduling more efficient. Furthermore, we hope that the heuristic solution methods developed can act as stepping stones for others when developing a fully functioning planning tool for surgical clinics.

Bibliography

Mohamed Amine Abdeljaouad, Zled Bahroun, Anders N. Gullhav, Nour El Houda Saadani, and Belhassen Zouari. A simulated annealing for a daily operating room scheduling problem under constraints of uncertainty and setup. *INFOR: Information Systems and Operational Research*, 58(3), 2020.

Ivo Adan, Jos Bekkers, Nico Dellaert, Jan Vissers, and Xiaoting Yu. Patient mix optimisation and stochastic resource requirements: A case study in cardiothoracic surgery planning. *Health Care Management Science volume*, 12(129):129–141, 2009.

Ivo Adan, Jos Bekkers, Nico Dellaert, Jully Jeunet, and Jan Vissers. Improving operational effectiveness of tactical master plans for emergency and elective patients under stochastic demand and capacitated resources. *European Journal of Operational Research*, 213(1):290–308, 2011.

Shabbir Ahmed and Alexander Shapiro. The sample average approximation method for stochastic programs with integer recourse. 2002. School of Industrial & Systems Engineering, Georgia Institute of Technology.

Henrik Andersson. Matheuristics, April 2021. [Unpublished Paper] Norwegian University of Science and Technology.

Sigrún Andradóttir and Andrei A. Prudius. Balanced explorative and exploitative search with estimation for simulation optimization. *INFORMS Journal on Computing*, 21(2):193–208, 2009.

Robert Newton Anthony. Planning and control systems: A framework for analysis. pages 129–175, 1965.

Øyvind Asplin, Erlend Johan Corneliussen, and Katarina Van de Pontseele. Stochastic programming for flexible resource planning at the clinic of surgery at st. olavs hospital. 2021. [Unpublished Paper] Norwegian University of Science and Technology.

Ana Batista, Jorge Vera, and David Pozo. Multi-objective admission planning problem: a two-stage stochastic approach. *Health Care Management Science*, 23:51–65, 2020.

Jeroen Beliën and Erik Demeulemeester. Building cyclic master surgery schedules with leveled resulting bed occupancy. *European Journal of Operational Research*, 176(2):1185–1204, 2007.

Leonora Bianchi, Marco Dorigo, Luca Maria Gambardella, and Walter J. Gutjahr. A survey on metaheuristics for stochastic combinatorial optimization. *Nat Comput*, 8:239–287, 2008.

Christian Blum and Andrea Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3):268–308, 2003.

Thomas Reiten Bovim, Marielle Christiansen, Anders N. Gullhav, Troels Martin Range, and Lars Hellemo. Stochastic master surgery scheduling. *European Journal of Operational Research*, 285(2): 695–711, 2020.

Justin Britt, M. Fazle Baki, Ahmed Azab, Abderrahmane Chaouch, and Xiangyong Li. A stochastic hierarchical approach for the master surgical scheduling problem. *Computers & Industrial Engineering*, 158, 2021.

M. E. Bruni, P. Beraldi, and D. Conforti. A stochastic programming approach for operating theatre scheduling under uncertainty. *IMA Journal of Management Mathematics*, 26(1):99–119, 2015.

Paolo Cappanera, Filippo Visintin, and Carlo Banditori. Comparing resource balancing criteria in master surgical scheduling: A combined optimisation-simulation approach. *International Journal of Production Economics*, 158:179–196, 2014.

H. Fei, N. Meskens, and C. Chu. A planning and scheduling problem for an operating theatre using an open scheduling strategy. *Computers & Industrial Engineering*, 58(2):221–230, 2010.

Andreas Fügener. An integrated strategic and tactical master surgery scheduling approach with stochastic resource demand. *Journal of Business Logistics*, 36(4):374–387, 2015.

Andreas Fügener, Erwin W. Hans, Rainer Kolisch, Nikky Kortbeek, and Peter T. Vanberkel. Master surgery scheduling with consideration of multiple downstream units. *European Journal of Operational Research*, 239(1):227–236, 2014.

Cheng Guo, Merve Bodur, Dionne M. Aleman, and David R. Urbach. Logic-based benders decomposition and binary decision diagram based approaches for stochastic distributed operating room scheduling. *INFORMS Journal on Computing*, 33(4):1551–1569, 2021.

Mehdi Heydari and Asie Soudi. Predictive / reactive planning and scheduling of a surgical suite with emergency patient arrival. *Journal of Medical Systems*, 40(30), 2016.

Peter J.H. Hulshof, Nikky Kortbeek, Richard J. Boucherie, Erwin W. Hans, and Piet J.M. Bakker. Taxonomic classification of planning decisions in health care: a structured review of the state of the art in or/ms. *Health Systems*, 1:129–175, 2012.

Pia Irgens Isaksen and Martine Svagård. A simulation-optimization approach to tactical surgery scheduling. 2021. [Unpublished Master thesis] Norwegian University of Science and Technology.

Angel A. Juan, Peter Keenan, Rafael Martí, Seán McGarraghy, Javier Panadero, Paula Carroll, and Diego Oliva. A review of the role of heuristics in stochastic optimisation: from metaheuristics to learnheuristics. *Annals of Operations Research*, 2021.

Mehdi A. Kamran, Behrooz Karimi, and Nico Dellaert. Uncertainty in advance scheduling problem in operating room planning. *Computers & Industrial Engineering*, 126:252–268, 2018.

Michael Kaut and Stein W. Wallace. Evaluation of scenario-generation methods for stochastic programming. *Stochastic Programming E-print Series (SPEPS)*, 2003, 2003.

Kibaek Kim and Sanjay Mehrotra. A two-stage stochastic integer programming approach to integrated staffing and scheduling with application to nurse management. *Operations Research*, 63(6):1431–1451, 2015.

Seung-Chul Kim and Ira Horowitz. Scheduling hospital services: the efficacy of elective-surgery quotas. *Omega*, 30(5):335–346, 2002.

Alan J. King and Stein W. Wallace. *Modeling with Stochastic Programming.* Springer New York, NY, 2003.

Ashwani Kumar, Alysson M. Costa, Mark Fackrell, and Peter G.Taylor. A sequential stochastic mixed integer programming model for tactical master surgery scheduling. *European Journal of Operational Research*, 270(2):734–746, 2018.

Wasakorn Laesanklang and Dario Landa-Silva. Decomposition techniques with mixed integer programming and heuristics for home healthcare planning. *Annals of Operations Research*, 256:93–127, 2017.

Mehdi Lamiri, Frédéric Grimaud, and Xiaolan Xie. Optimization methods for a stochastic surgery planning problem. *International Journal of Production Economics*, 120(2):400–410, 2009.

Jan Lundgren, Mikael Rönnqvist, and Peter Värbrand. *Optimization*. Studentlitteratur AB, 2010.

Salma Makboul, Said Kharraja, Abderrahman Abbassi, and Ahmed El Hilali Alaoui. A two-stage robust optimization approach for the master surgical schedule problem under uncertainty considering downstream resources. *Health Care Management Science*, 2021.

Carlo Mannino, Eivind J. Nilssen, and Tomas Eric Nordlander. A pattern based, robust approach to cyclic master surgery scheduling. *Journal of Scheduling*, 15:553–563, 2012.

Rym M'Hallah and Filippo Visintin. A stochastic model for scheduling elective surgeries in a cyclic master surgical schedule. *Computers & Industrial Engineering*, 129:156–168, 2019.

Ministry of Health and Care Services. Hospitals. https://www.regjeringen.no/en/topics/health-and-care/hospitals/id10935/, 2020a. [accessed: 08.12.2021].

Ministry of Health and Care Services. Nasjonal helse- og sykehusplan 2020-2023. https://www.regjeringen.no/no/dokumenter/nasjonal-helse--og-sykehusplan-2020-2023/id2679013/, 2020b. [accessed: 08.12.2021].

Saba Neyshabouri and Bjorn P. Berg. Two-stage robust optimization approach to elective surgery and downstream capacity planning. *European Journal of Operational Research*, 260(1):21–40, 2017.

Mariana Oliveira, Filippo Visintin, Daniel Santos, and Inês Marques. Flexible master surgery scheduling: combining optimization and simulation in a rolling horizon approach. *Flexible Services and Manufacturing Journal*, 2021.

Samik Raychaudhuri. Introduction to monte carlo simulation. In *2008 Winter Simulation Conference*, pages 91–100. Winter Simulation Conference, 2008.

Vahid Roshanaei and Bahman Naderi. Solving integrated operating room planning and scheduling: Logic-based benders decomposition versus branch-price-and-cut. *European Journal of Operational Research*, 293:65–78, 2021.

Michael Samudra, Carla Van Riet, Erik Demeulemeester, Brecht Cardoen, Nancy Vansteenkiste, and Frank E. Rademakers. Scheduling operating rooms: achievements, challenges and pitfalls. *Journal of Scheduling*, 19:493–525, 2016.

A. J. Thomas Schneider, J. Theresia van Essen, Mijke Carlier, and Erwin W. Hans. Scheduling surgery groups considering multiple downstream resources. *European Journal of Operational Research*, 282(2): 741–752, 2020.

Alexander Shapiro and Arkadi Nemirovski. On complexity of stochastic programming problems. In Valthilingam Jeyakumar and Alexander Rubinov, editors, *Continuous Optimization: Current Trends and Modem Applications*, chapter 4, pages 111–144. Springer Science + Business Media, 233 Spring Street, New York, NY 10013, USA, 2005.

Alexander Shapiro and Andy Philpott. A tutorial on stochastic programming. 2007.

Belinda Spratt and Erhan Kozan. Waiting list management through master surgical schedules: A case study. *Operations Research for Health Care*, 10:49–64, 2016.

St. Olavs Hospital. Bryst og endokrinkirurgisk seksjon. https://stolav.no/avdelinger/kirurgisk-klinikk/bryst-og-endokrinkirurgi, 2016a. [accessed: 09.12.2021].

St. Olavs Hospital. Gastroenterologiske kirurgiske seksjoner. https://stolav.no/avdelinger/kirurgisk-klinikk/gastroenterologisk-kirurgi, 2016b. [accessed: 09.12.2021].

St. Olavs Hospital. Kirurgisk klinikk. https://stolav.no/avdelinger/kirurgisk-klinikk, 2016c. [accessed: 08.12.2021].

St. Olavs Hospital. Om oss. https://stolav.no/om-oss, 2016d. [accessed: 08.12.2021].

St. Olavs Hospital. Urologisk seksjon. https://stolav.no/avdelinger/kirurgisk-klinikk/urologisk-seksjon, 2016e. [accessed: 09.12.2021].

St. Olavs Hospital. Karkirurgisk seksjon. https://stolav.no/avdelinger/kirurgisk-klinikk/karkirurgi, 2016f. [accessed: 09.12.2021].

J. Theresia van Essen, Joël M. Bosch, Erwin W. Hans, Mark van Houdenhoven, and Johann L. Hurink. Reducing the number of required beds by rearranging the or-schedule. *OR Spectrum*, 36:585–605, 2014.

Jeroen M. van Oostrum, M. Van Houdenhoven, J. L. Hurink, E. W. Hans, G. Wullink, and G. Kazemier. A master surgical scheduling approach for cyclic scheduling in operating room departments. *OR Spectrum*, 30:355–374, 2008.

Peter T. Vanberkel, Richard J. Boucherie, Erwin W. Hans, Johann L. Hurink, Wineke van Lent, and Wim van Harten. Accounting for inpatient wards when developing master surgical schedules. *Anesthesia & Analgesia*, 112(6):1472–1479, 2011.

Filippo Visintin, Paola Cappanera, and Carlo Bandatori. Evaluating the impact of flexible practices on the master surgical scheduling process: an empirical analysis. *Flexible Services and Manufacturing Journal*, 28:182–205, 2016.

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. *International Journal of Production Economics*, 158:28–36, 2014.

Appendices

Appendix A

Base Model Formulation

Sets and Indices

$\mathcal{W}$	Wards, indexed w
$\mathcal{S}$	Specialties, indexed s
$\mathcal{S}_r^{\mathcal{R}}$	Specialties suitable for OR r
$\mathcal{G}$	Surgery groups, indexed g
$\mathcal{G}_w^{\mathcal{W}}$	Surgery groups that can receive postoperative care at ward w, indexed g
$\mathcal{G}_s^{\mathcal{S}}$	Surgery groups that can receive treatment from specialty s, indexed g
$\mathcal{R}$	ORs, indexed r
$\mathcal{R}_s^{\mathcal{S}}$	ORs suitable for specialty s, indexed r
$\mathcal{R}_g^{\mathcal{G}}$	ORs suitable for group g, indexed r
$\mathcal{D}$	Days in planning period, indexed d
$\mathcal{C}$	Scenarios, indexed c

Parameters

Π_c	Probability of scenario c occurring
C_g	Unit cost of not meeting the demand of surgery group g
T^C	Cleaning time post surgery
L_g^{SD}	Surgery duration of a patient in surgery group g
F	Maximum percentage of flexible number of slots
N_d	Total number of available ORs on day d
U_s^X	Maximum number of times a specialty s may extend its opening hours during a cycle
I	Number of cycles in the planning horizon
K_{sd}	Number of surgical teams from specialty s available on day d
H_d	Default amount of time available in a slot if it is assigned at day d
E	Additional time available if a slot's opening hours are extended
Q_{gc}	Number of patients from surgery group g in line for surgery in scenario c
P_{gwd}	Probability that a patient from surgery group g occupies a bed in ward w, on the night d days after surgery
J_w	Maximum number of nights a patient may stay in ward w
B_{wd}	Number of available beds at ward w on the night following day d
Y_{wd}	Expected number of occupied beds in ward w the night following day d in the current planning period by patients operated on in the previous planning period

Variables

First stage decision variables:

γ_{srd} $\begin{cases} 1 & \text{if specialty } s \text{ is assigned a fixed slot in room } r \text{ on day } d \\ 0 & \text{Otherwise} \end{cases}$

λ_{srd} $\begin{cases} 1 & \text{if specialty } s \text{ extends opening hours in room } r \text{ on day } d \\ 0 & \text{Otherwise} \end{cases}$

Second stage decision variables:

δ_{srdc} $\begin{cases} 1 & \text{if specialty } s \text{ is assigned a flexible slot in room } r \text{ on day } d \text{ in scenario } c \\ 0 & \text{Otherwise} \end{cases}$

x_{grdc} Number of patients from surgery group g operated in room r on day d in scenario c

Auxiliary variables:

a_{gc} Number of patients from surgery group g waiting in line for surgery, but not scheduled for surgery in scenario c

Model formulation

$$\min \sum_{c\in\mathcal{C}} \Pi_c \sum_{g\in\mathcal{G}} \mathrm{C}_g a_{gc} \quad \text{(A.1)}$$

$$\sum_{s\in\mathcal{S}} \sum_{r\in\mathcal{R}^{\mathcal{S}}_s} \sum_{d\in\mathcal{D}} \gamma_{srd} - \left\lceil (1-F) \sum_{d\in\mathcal{D}} N_d \right\rceil \geq 0 \quad \text{(A.2)}$$

$$\lambda_{srd} \leq \gamma_{srd} \qquad s\in\mathcal{S}, r\in\mathcal{R}^{\mathcal{S}}_s, d\in\mathcal{D} \quad \text{(A.3)}$$

$$\sum_{r\in\mathcal{R}^{\mathcal{S}}_s} \sum_{d\in\mathcal{D}} \lambda_{srd} \leq U^{\mathrm{X}}_s \qquad s\in\mathcal{S} \quad \text{(A.4)}$$

$$\sum_{s\in\mathcal{S}} \gamma_{srd} + \delta_{srdc} \leq 1 \qquad r\in\mathcal{R}, d\in\mathcal{D}, c\in\mathcal{C} \quad \text{(A.5)}$$

$$\sum_{r\in\mathcal{R}^{\mathcal{S}}_s} \gamma_{srd} + \delta_{srdc} \leq \mathrm{K}_{sd} \qquad s\in\mathcal{S}, d\in\mathcal{D}, c\in\mathcal{C} \quad \text{(A.6)}$$

$$\sum_{s\in\mathcal{S}} \sum_{r\in\mathcal{R}^{\mathcal{S}}_s} \gamma_{srd} + \delta_{srdc} \leq \mathrm{N}_d \qquad d\in\mathcal{D}, c\in\mathcal{C} \quad \text{(A.7)}$$

$$\sum_{g\in\mathcal{G}^{\mathcal{S}}_s} (\mathrm{L}_g + \mathrm{T}^{\mathrm{C}}) x_{grdc} \leq \mathrm{H}_d(\gamma_{srd} + \delta_{srdc}) + \mathrm{E}\lambda_{srd} \qquad s\in\mathcal{S}, r\in\mathcal{R}^{\mathcal{S}}_s, d\in\mathcal{D}, c\in\mathcal{C} \quad \text{(A.8)}$$

$$\sum_{r\in\mathcal{R}} \sum_{d\in\mathcal{D}} x_{grdc} + a_{gc} = \mathrm{Q}_{gc} \qquad g\in\mathcal{G}, c\in\mathcal{C} \quad \text{(A.9)}$$

$$\sum_{s\in\mathcal{S}^{\mathcal{R}}_r} \delta_{srdc} \leq \sum_{g\in\mathcal{G}} x_{grdc} \qquad r\in\mathcal{R}, d\in\mathcal{D}, c\in\mathcal{C} \quad \text{(A.10)}$$

$$\sum_{g\in\mathcal{G}^{\mathcal{W}}_w} \sum_{r\in\mathcal{R}} \sum_{\delta=\max\{1,d+1-\mathrm{J}_w\}}^{d} \mathrm{P}_{wg(d-\delta+1)} x_{gr\delta c} \leq \mathrm{B}_{wd} - Y_{wd} \qquad w\in\mathcal{W}, d\in\mathcal{D}, c\in\mathcal{C} \quad \text{(A.11)}$$

$$\gamma_{srd} = \gamma_{sr(\frac{|\mathcal{D}|}{I}+d)} \qquad s\in\mathcal{S}, r\in\mathcal{R}^{\mathcal{S}}_s, d=1,...,|\mathcal{D}| - \frac{|\mathcal{D}|}{I} \quad \text{(A.12)}$$

$$\lambda_{srd} = \lambda_{sr(\frac{|\mathcal{D}|}{I}+d)} \qquad s\in\mathcal{S}, r\in\mathcal{R}^{\mathcal{S}}_s, d=1,...,|\mathcal{D}| - \frac{|\mathcal{D}|}{I} \quad \text{(A.13)}$$

$$\gamma_{srd} = 0 \qquad s\in\mathcal{S}, r\in\{\mathcal{R}\backslash\mathcal{R}^{\mathcal{S}}_s\}, d\in\mathcal{D} \quad \text{(A.14)}$$

$$\delta_{srdc} = 0 \qquad s\in\mathcal{S}, r\in\{\mathcal{R}\backslash\mathcal{R}^{\mathcal{S}}_s\}, d\in\mathcal{D}, c\in\mathcal{C} \quad \text{(A.15)}$$

$$x_{grdc} = 0 \qquad g\in\mathcal{G}, r\in\{\mathcal{R}\backslash\mathcal{R}^{\mathcal{G}}_g\}, d\in\mathcal{D}, c\in\mathcal{C} \quad \text{(A.16)}$$

$$\gamma_{srd}, \lambda_{srd} \in \{0,1\} \qquad s\in\mathcal{S}, r\in\mathcal{R}, d\in\mathcal{D} \quad \text{(A.17)}$$

$$\delta_{srdc} \in \{0,1\} \qquad s\in\mathcal{S}, r\in\mathcal{R}, d\in\mathcal{D}, c\in\mathcal{C} \quad \text{(A.18)}$$

$$x_{grdc} \in \mathbb{Z}^+ \qquad g\in\mathcal{G}, r\in\mathcal{R}, d\in\mathcal{D}, c\in\mathcal{C} \quad \text{(A.19)}$$

$$a_{gc} \in \mathbb{Z}^+ \qquad g\in\mathcal{G}, c\in\mathcal{C} \quad \text{(A.20)}$$

Constraint description

(A.1) Is the objective function, and minimizes the total expected cost of unmet demand.
(A.2) Ensures a minimum percentage of fixed slots.
(A.3) Ensure that a specialty can only extend a slot if it is assigned to that slot in the fixed schedule.
(A.4) Ensure that each specialty does not extend more days during the planning period than they are allowed.
(A.5) Ensure that no ORs can be double booked.
(A.6) Ensure that no specialty is assigned more ORs than they have teams available on that day.
(A.7) Ensure that we can not assign more ORs than there are ORs available each day.
(A.8) Ensure that the total planned operating and cleaning time in a slot does not exceed the slot's available time. They also ensure that patients can only be planned for surgery in a slot assigned to the specialty that they belong to.
(A.9) Keep track of the unmet demand of surgery group g in scenario c through the variable a_{gc}.
(A.10) Ensure that a flexible slot is only assigned to a specialty if there is at least one planned operation in that slot.
(A.11) Ensure that the expected number of beds occupied in a ward on the night following day d does not exceed the number of beds available. They take into account the expected number of beds occupied by patients operated in the previous period.
(A.12) Ensure that fixed slots repeat themselves in every cycle of the planning period.
(A.13) Ensure that extended slots repeat themselves in every cycle throughout the planning period.
(A.14) and (A.15) ensure that a specialty is not assigned to an OR that is not suitable.
(A.16) ensures that no patients are scheduled in an OR not suitable for them.
(A.17) and (A.18) ensure that the given variables are binary.
(A.19) and (A.20) ensure that the given variables are integer and non-negative.

Appendix B

Cutting Stock Formulation

We propose an alternative formulation inspired by the cutting stock problem by using the patterns of surgery groups described in Section 6.3.1. A solution from the cutting stock formulation can be translated to variables of the base formulation in Chapter 5 through the formula:

$$x_{grdc} = \sum_{m \in \mathcal{M}} A_{mg} \phi_{mrdc} \qquad g \in \mathcal{G}, r \in \mathcal{R}, d \in \mathcal{D}, c \in \mathcal{C}$$

Note that we in this appendix only present additional notation, while most of the notation is the same as in Appendix A. The performance of this formulation is compared to the performance of the base formulation in Appendix C.

New Sets and Indices

$\mathcal{M}$	Patterns of operations, indexed m
$\mathcal{M}^X$	Patterns of operations only suitable for extended days, indexed m
$\mathcal{M}^{NX}$	Patterns of operations suitable for non-extended days, indexed m
$\mathcal{M}_s^S$	Patterns associated with specialty s, indexed m
$\mathcal{R}_m^{\mathcal{M}}$	ORs suitable for pattern m, indexed r

New Parameters

P_{mwd}^{CS}	Accumulated probability that patients in pattern m occupy a bed in ward w, on the night d days after surgery
A_{mg}	Number of patients from group g included in pattern m

New Variables

Second-stage decision variables:

$$\phi_{mrdc} \quad \begin{cases} 1 & \text{if pattern } m \text{ is assigned a to a slot in room } r \text{ on day } d \text{ in scenario } c \\ 0 & \text{Otherwise} \end{cases}$$

Model formulation

$$\min \sum_{c\in\mathcal{C}} \Pi_c \sum_{g\in\mathcal{G}} \mathrm{C}_g a_{gc} \tag{B.1}$$

$$\sum_{s\in\mathcal{S}} \sum_{r\in\mathcal{R}^{\mathcal{S}}_r} \sum_{d\in\mathcal{D}} \gamma_{srd} - \left\lceil (1-F) \sum_{d\in\mathcal{D}} N_d \right\rceil \geq 0 \tag{B.2}$$

$$\lambda_{srd} \leq \gamma_{srd} \qquad s\in\mathcal{S}, r\in\mathcal{R}^{\mathcal{S}}_s, d\in\mathcal{D} \tag{B.3}$$

$$\sum_{r\in\mathcal{R}^{\mathcal{S}}_s} \sum_{d\in\mathcal{D}} \lambda_{srd} \leq \mathrm{U}^{\mathrm{X}}_s \qquad s\in\mathcal{S} \tag{B.4}$$

$$\sum_{s\in\mathcal{S}} \gamma_{srd} + \delta_{srdc} \leq 1 \qquad r\in\mathcal{R}, d\in\mathcal{D}, c\in\mathcal{C} \tag{B.5}$$

$$\sum_{r\in\mathcal{R}^{\mathcal{S}}_s} \gamma_{srd} + \delta_{srdc} \leq \mathrm{K}_{sd} \qquad s\in\mathcal{S}, d\in\mathcal{D}, c\in\mathcal{C} \tag{B.6}$$

$$\sum_{s\in\mathcal{S}} \sum_{r\in\mathcal{R}^{\mathcal{S}}_s} \gamma_{srd} + \delta_{srdc} \leq \mathrm{N}_d \qquad d\in\mathcal{D}, c\in\mathcal{C} \tag{B.7}$$

$$\sum_{m\in\mathcal{M}^{\mathcal{S}}_s} \phi_{mrdc} \leq \gamma_{srd} + \delta_{srdc} \qquad s\in\mathcal{S}, r\in\mathcal{R}^{S}_s, d\in\mathcal{D}, c\in\mathcal{C} \tag{B.8}$$

$$2\sum_{m\in\mathcal{M}^X} \phi_{mrdc} + \sum_{m\in\mathcal{M}^{NX}} \phi_{mrdc} \leq \sum_{s\in\mathcal{S}} \lambda_{srd} + \gamma_{srd} + \delta_{srdc} \qquad r\in\mathcal{R}, d\in\mathcal{D}, c\in\mathcal{C} \tag{B.9}$$

$$\sum_{m\in\mathcal{M}} \sum_{r\in\mathcal{R}^{\mathcal{G}}_g} \sum_{d\in\mathcal{D}} A_{mg}\phi_{mrdc} + a_{gc} = \mathrm{Q}_{gc} \qquad g\in\mathcal{G}, c\in\mathcal{C} \tag{B.10}$$

$$\sum_{m\in\mathcal{M}^{NX}} \sum_{g\in\mathcal{G}} A_{mg}\phi_{mrdc} \geq \sum_{s\in\mathcal{S}^{\mathcal{R}_r}} \delta_{srdc} \qquad r\in\mathcal{R}, d\in\mathcal{D}, c\in\mathcal{C} \tag{B.11}$$

$$\sum_{m\in\mathcal{M}} \sum_{r\in\mathcal{R}^{\mathcal{M}}_m} \sum_{d'=\max\{1,d+1-\mathrm{J}_w\}}^{d} \mathrm{P}^{CS}_{mw(d-d'+1)}\phi_{mrd'c} \leq \mathrm{B}_{wd} - Y_{wd} \qquad w\in\mathcal{W}, d\in\mathcal{D}, c\in\mathcal{C} \tag{B.12}$$

$$\gamma_{srd} = \gamma_{sr(\frac{|\mathcal{D}|}{\mathrm{I}}+d)} \qquad s\in\mathcal{S}, r\in\mathcal{R}^{\mathcal{S}}_s, d=1,...,|\mathcal{D}|-\frac{|\mathcal{D}|}{\mathrm{I}} \tag{B.13}$$

$$\lambda_{srd} = \lambda_{sr(\frac{|\mathcal{D}|}{\mathrm{I}}+d)} \qquad s\in\mathcal{S}, r\in\mathcal{R}^{\mathcal{S}}_s, d=1,...,|\mathcal{D}|-\frac{|\mathcal{D}|}{\mathrm{I}} \tag{B.14}$$

$$\gamma_{srd}, \lambda_{srd} = 0 \qquad s\in\mathcal{S}, r\in\{\mathcal{R}\backslash\mathcal{R}^{\mathcal{S}}_s\}, d\in\mathcal{D} \tag{B.15}$$

$$\delta_{srdc} = 0 \qquad s\in\mathcal{S}, r\in\{\mathcal{R}\backslash\mathcal{R}^{\mathcal{S}}_s\}, d\in\mathcal{D}, c\in\mathcal{C} \tag{B.16}$$

$$\phi_{mrdc} = 0 \qquad m\in\mathcal{M}, r\in\{\mathcal{R}\backslash\mathcal{R}^{\mathcal{M}}_m\}, d\in\mathcal{D}, c\in\mathcal{C} \tag{B.17}$$

$$\gamma_{srd}, \lambda_{srd} \in \{1,0\} \qquad s\in\mathcal{S}, r\in\mathcal{R}, d\in\mathcal{D} \tag{B.18}$$

$$\delta_{srdc} \in \{1,0\} \qquad s\in\mathcal{S}, r\in\mathcal{R}, d\in\mathcal{D}, c\in\mathcal{C} \tag{B.19}$$

$$\phi_{mrdc} \in \{1,0\} \qquad m\in\mathcal{M}, r\in\mathcal{R}, d\in\mathcal{D}, c\in\mathcal{C} \tag{B.20}$$

$$a_{gc} \in \mathbb{Z}^+ \qquad g\in\mathcal{G}, c\in\mathcal{C} \tag{B.21}$$

Constraint description
(B.1) Is the objective function, and minimizes the total expected cost of unmet demand.
(B.2) Ensures a minimum percentage of fixed slots.
(B.3) Ensure that a specialty can only extend a slot if it is assigned to that slot in the fixed schedule.
(B.4) Ensure that each specialty does not extend more days during the planning period than they are allowed.
(B.5) Ensure that no ORs can be double booked.
(B.6) Ensure that no specialty is assigned more ORs than they have teams available on that day.
(B.7) Ensure that we can not assign more ORs than there are ORs available each day.
(B.8) Ensure that each slot is filled with a legal pattern according to its assigned specialty.
(B.9) Ensure that each slot contains a legal pattern according to its type (extended, non-extended, flexible).
(B.10) Keep track of the unmet demand of surgery group g in scenario c through the variable a_{gc}.
(B.11) Ensure that a flexible slot is only assigned to a specialty if there is at least one planned operation in that slot.
(B.12) Ensure that the expected number of beds occupied in a ward on the night following day d does not exceed the number of beds available. They take into account the expected number of beds occupied by patients operated on in the previous period.
(B.13) Ensure that fixed slots repeat themselves in every cycle of the planning period.
(B.14) Ensure that extended slots repeat themselves in every cycle throughout the planning period.
(B.15) and (B.16) ensure that a specialty is not assigned to an OR that is not suitable.
(B.17) Ensure that a pattern is not assigned to an OR that is not suitable.
(B.18), (B.19) and (B.20) ensure that the given variables are binary.
(B.21) Ensure that the auxiliary variable a_{gc} is integer and non-negative.

Appendix C

Comparing Base and Cutting Stock Formulation

C.1 Problem Size

Table C.1: Problem size and complexity for the base model formulation with different numbers of surgery groups $|\mathcal{G}|$ and scenarios $|\mathcal{C}|$ at flexibility $F = 10\%$ and original bed ward capacity $\beta = 1.0$.

Model	$\lvert\mathcal{G}\rvert$	$\boldsymbol{F}$	$\lvert\mathcal{C}\rvert$	**Prim** [min]	**Dual** [min]	**Gap**	**Presolver** **Row**	**Col**	**Bin**	**Nonzero**	**Time** [s]
BASE	*5*	*10%*	*10*	667	667	0,0%	6310	5439	2810	53197	*124*
BASE	*5*	*10%*	*50*	1087	1087	0,0%	31205	26398	13253	265354	*161*
BASE	*5*	*10%*	*100*	1002	1002	0,0%	62330	52607	26338	530621	*603*
BASE	*9*	*10%*	*10*	310	236	23,7%	6404	7868	5024	84826	**3600*
BASE	*9*	*10%*	*50*	792	565	28,7%	31480	38464	23918	423978	**3600*
BASE	*9*	*10%*	*100*	2126	606	71,5%	62883	76627	47637	852415	**3600*
BASE	*25*	*10%*	*10*	1580	738	53,3%	6571	17876	7222	197845	**3600*
BASE	*25*	*10%*	*50*	3494	225	93,6%	32340	88685	35470	990299	**3600*
BASE	*25*	*10%*	*100*	2190	325	85,2%	64572	176817	70159	1973904	**3600*

All scenario trees have been generated by the same seed $m = 1$.
Row, Col, Bin and Nonzero refer to the constraints, variables, binary variables and non-zero elements in the model after the presolver have been run.
* means that the maximum runtime is reached.

Table C.2: Problem size and complexity for the cutting stock formulation for different numbers of surgery groups $|\mathcal{G}|$ and scenarios $|\mathcal{C}|$ at flexibility $F = 10\%$, with original bed ward capacity $\beta = 1.0$.

Model	$\|\mathcal{G}\|$	F	$\|\mathcal{S}\|$	**Prim** [min]	**Dual** [min]	**Gap**	**Presolver** Row	Col	Bin	Nonzero	**Time** [s]
CS	*5*	*10%*	*10*	667	667	0.0%	7745	9649	9600	121381	*73*
CS	*5*	*10%*	*50*	1087	1057	2.7%	38244	47328	47083	605890	**3600*
CS	*5*	*10%*	*100*	1002	896	10.6%	76373	94417	93928	1211381	**3600*
CS	*9*	*10%*	*10*	310	274	11.7%	7824	16308	16224	216584	**3600*
CS	*9*	*10%*	*50*	951	674	29.1%	38580	81084	80658	1088621	**3600*
CS	*9*	*10%*	*100*	4290	627	85.4%	76983	161967	161117	2177778	**3600*
CS	*25*	*10%*	*10*	1598	1331	16.7%	8771	133195	132213	2093904	**3600*
CS	*25*	*10%*	*50*	6069	950	84.4%	43293	666477	661564	10498566	**3600*
CS	*25*	*10%*	*100*	7232	992	86.3%	86468	1330472	1320638	20954055	**3600*

All scenario trees have been generated by the same seed $m = 1$.
Row, Col, Bin and Nonzero refer to the constraints, variables, binary variables and non-zero elements in the model after the presolver have been run.
* means that the maximum runtime is reached.

C.2 Long Run Performance

Table C.3: Results from 12h runs of the base model formulation of the MIP on $|\mathcal{G}| = 9$ groups solved for a scenario tree $\breve{\boldsymbol{\xi}}^1_{250}$ of size $N = 250$ generated from seed $m = 1$ with original bed ward capacity $\beta = 1.0$ at each flexibility level $F \in \{0\%, 5\%, \ldots 30\%\}$.

Formulation	$\|\mathcal{G}\|$	F	N	β	**Primal**	**Dual**	**Gap**	***Time** [h]*
BASE	*9*	*0%*	*250*	*1.0*	1557.2	1544.6	0.8%	*12*
BASE	*9*	*5%*	*250*	*1.0*	2181.2	994.3	54.4%	*12*
BASE	*9*	*10%*	*250*	*1.0*	1475.1	721.1	51.1%	*12*
BASE	*9*	*15%*	*250*	*1.0*	1517.2	604.7	60.1%	*12*
BASE	*9*	*20%*	*250*	*1.0*	2559..0	554.0	78.4%	*12*
BASE	*9*	*25%*	*250*	*1.0*	2182.7	26.2	98.8%	*12*
BASE	*9*	*30%*	*250*	*1.0*	1357.6	48.0	95.8%	*12*

All scenario trees have been generated by the same seed $m = 1$.

Table C.4: Results from 12h runs of the cutting stock (CS) formulation of the MIP on $|\mathcal{G}| = 9$ groups solved for a scenario tree $\breve{\boldsymbol{\xi}}^1_{250}$ of size $N = 250$ generated from seed $m = 1$ with original bed ward capacity $\beta = 1.0$ at each flexibility level $F \in \{0\%, 5\%, \ldots 30\%\}$.

Formulation	$\|\mathcal{G}\|$	F	N	β	**Primal**	**Dual**	**Gap**	***Time** [h]*
CS	*9*	*0%*	*250*	*1.0*	2101.2	1222.4	41.8%	*12*
CS	*9*	*5%*	*250*	*1.0*	3052.5	956.5	68.7%	*12*
CS	*9*	*10%*	*250*	*1.0*	2208.8	834.2	62.2%	*12*
CS	*9*	*15%*	*250*	*1.0*	4831.8	808.3	83.3%	*12*
CS	*9*	*20%*	*250*	*1.0*	3410.3	816.5	76.1%	*12*
CS	*9*	*25%*	*250*	*1.0*	2145.2	782.1	63.5%	*12*
CS	*9*	*30%*	*250*	*1.0*	3143.8	829.0	73.6%	*12*

All scenario trees have been generated by the same seed $m = 1$.

Appendix D

Tabulated Results

D.1 SA Runs

Table D.1: Results from 10 runs of SA-2SMIP on $|\mathcal{G}| = 9$ groups solved for a scenario tree $\breve{\boldsymbol{\xi}}^1_{250}$ of size $N = 250$ generated with seed $m = 1$, original bed ward capacity $\beta = 1.0$ and flexibility $F = 10\%$ with the same SA parameters $\{T_0, \alpha, I_{max}, T_{min}\}$. The runs are sorted from lowest to highest objective value.

$\|\mathcal{G}\|$	F	N	β	T_0	α	I_{max}	T_{min}	Best OV	Elapsed Time [h]
9	10%	250	1.0	1000	0.9	25	0.01	1001.3	9.5
9	10%	250	1.0	1000	0.9	25	0.01	1001.3	6.4
9	10%	250	1.0	1000	0.9	25	0.01	1013.0	9.0
9	10%	250	1.0	1000	0.9	25	0.01	1022.4	5.8
9	10%	250	1.0	1000	0.9	25	0.01	1035.0	7.0
9	10%	250	1.0	1000	0.9	25	0.01	1035.0	4.9
9	10%	250	1.0	1000	0.9	25	0.01	1045.9	8.0
9	10%	250	1.0	1000	0.9	25	0.01	1045.9	5.2
9	10%	250	1.0	1000	0.9	25	0.01	1055.5	7.5
9	10%	250	1.0	1000	0.9	25	0.01	1078.4	7.6

All scenario trees have been generated by the same seed $m = 1$.

Table D.2: Results from 10 runs of SA-GCH-2SMIP on $|\mathcal{G}| = 9$ groups solved for a scenario tree $\breve{\boldsymbol{\xi}}^1_{250}$ of size $N = 250$ generated with seed $m = 1$, original bed ward capacity $\beta = 1.0$ and flexibility $F = 10\%$ with the same SA parameters $\{T_0, \alpha, I_{max}, T_{min}\}$. The runs are sorted from lowest to highest objective value.

$\|\mathcal{G}\|$	F	N	β	T_0	α	I_{max}	T_{min}	Best OV	Elapsed Time [s]
9	10%	250	1.0	1000	0.9	50	0.01	1092.9	1045
9	10%	250	1.0	1000	0.9	50	0.01	1116.7	989
9	10%	250	1.0	1000	0.9	50	0.01	1136.9	992
9	10%	250	1.0	1000	0.9	50	0.01	1136.9	1008
9	10%	250	1.0	1000	0.9	50	0.01	1152.0	1017
9	10%	250	1.0	1000	0.9	50	0.01	1165.9	1039
9	10%	250	1.0	1000	0.9	50	0.01	1168.9	993
9	10%	250	1.0	1000	0.9	50	0.01	1176.2	979
9	10%	250	1.0	1000	0.9	50	0.01	1176.8	997
9	10%	250	1.0	1000	0.9	50	0.01	1176.8	1009

All scenario trees have been generated by the same seed $m = 1$.

Table D.3: Results from 10 runs of SA-2SMIP on $|\mathcal{G}| = 9$ groups solved for a scenario tree $\breve{\boldsymbol{\xi}}^{1}_{250}$ of size $N = 250$ generated with seed $m = 1$, original bed ward capacity $\beta = 1.0$ and flexibility $F \in \{0\%, 5\%, 10\%, 15\%\}$ with the same SA parameters $\{T_0, \alpha, I_{max}, T_{min}\}$. The runs are sorted from lowest to highest objective value at each flexibility level F.

$\|\mathcal{G}\|$	F	N	β	T_0	α	I_{max}	T_{min}	Best OV	Elapsed Time [h]
9	0%	250	1.0	1000	0.9	25	0.01	1557.2	7.4
9	0%	250	1.0	1000	0.9	25	0.01	1557.2	9.8
9	0%	250	1.0	1000	0.9	25	0.01	1557.2	7.5
9	0%	250	1.0	1000	0.9	25	0.01	1557.2	8.4
9	0%	250	1.0	1000	0.9	25	0.01	1557.2	9.1
9	0%	250	1.0	1000	0.9	25	0.01	1557.2	7.5
9	0%	250	1.0	1000	0.9	25	0.01	1557.2	8.3
9	0%	250	1.0	1000	0.9	25	0.01	1557.2	9.1
9	0%	250	1.0	1000	0.9	25	0.01	1557.2	8.8
9	0%	250	1.0	1000	0.9	25	0.01	1557.2	9.9
9	5%	250	1.0	1000	0.9	25	0.01	1235.8	6.3
9	5%	250	1.0	1000	0.9	25	0.01	1235.8	7.7
9	5%	250	1.0	1000	0.9	25	0.01	1235.8	6.7
9	5%	250	1.0	1000	0.9	25	0.01	1235.8	6.9
9	5%	250	1.0	1000	0.9	25	0.01	1235.8	6.4
9	5%	250	1.0	1000	0.9	25	0.01	1235.8	8.0
9	5%	250	1.0	1000	0.9	25	0.01	1236.4	7.3
9	5%	250	1.0	1000	0.9	25	0.01	1236.4	7.3
9	5%	250	1.0	1000	0.9	25	0.01	1309.4	7.0
9	5%	250	1.0	1000	0.9	25	0.01	1309.4	7.5
9	10%	250	1.0	1000	0.9	25	0.01	1001.3	9.5
9	10%	250	1.0	1000	0.9	25	0.01	1001.3	6.4
9	10%	250	1.0	1000	0.9	25	0.01	1013.0	9.0
9	10%	250	1.0	1000	0.9	25	0.01	1022.4	5.8
9	10%	250	1.0	1000	0.9	25	0.01	1035.0	7.0
9	10%	250	1.0	1000	0.9	25	0.01	1035.0	4.9
9	10%	250	1.0	1000	0.9	25	0.01	1045.9	8.0
9	10%	250	1.0	1000	0.9	25	0.01	1045.9	5.2
9	10%	250	1.0	1000	0.9	25	0.01	1055.5	7.5
9	10%	250	1.0	1000	0.9	25	0.01	1078.4	7.6
9	15%	250	1.0	1000	0.9	25	0.01	1076.8	7.6
9	15%	250	1.0	1000	0.9	25	0.01	1088.9	8.3
9	15%	250	1.0	1000	0.9	25	0.01	1088.9	9.4
9	15%	250	1.0	1000	0.9	25	0.01	1088.9	6.8
9	15%	250	1.0	1000	0.9	25	0.01	1089.3	10.3
9	15%	250	1.0	1000	0.9	25	0.01	1089.3	7.0
9	15%	250	1.0	1000	0.9	25	0.01	1095.1	6.6
9	15%	250	1.0	1000	0.9	25	0.01	1095.1	7.2
9	15%	250	1.0	1000	0.9	25	0.01	1112.3	8.6
9	15%	250	1.0	1000	0.9	25	0.01	1112.3	9.5

All scenario trees have been generated by the same seed $m = 1$.

Table D.4: Results from 10 runs of SA-2SMIP on $|\mathcal{G}| = 9$ groups solved for a scenario tree $\breve{\boldsymbol{\xi}}^1_{250}$ of size $N = 250$ generated with seed $m = 1$, original bed ward capacity $\beta = 1.0$ and flexibility $F \in \{20\%, 25\%, 30\%\}$ with the same SA parameters $\{T_0, \alpha, I_{max}, T_{min}\}$. The runs are sorted from lowest to highest objective value at each flexibility level F.

$\lvert\mathcal{G}\rvert$	F	N	β	T_0	α	I_{max}	T_{min}	Best OV	Elapsed Time [h]
9	20%	250	1.0	1000	0.9	25	0.01	952.5	7.7
9	20%	250	1.0	1000	0.9	25	0.01	957.0	6.6
9	20%	250	1.0	1000	0.9	25	0.01	957.0	7.7
9	20%	250	1.0	1000	0.9	25	0.01	978.4	11.1
9	20%	250	1.0	1000	0.9	25	0.01	978.4	5.6
9	20%	250	1.0	1000	0.9	25	0.01	979.9	8.9
9	20%	250	1.0	1000	0.9	25	0.01	1001.3	8.2
9	20%	250	1.0	1000	0.9	25	0.01	1001.3	7.0
9	20%	250	1.0	1000	0.9	25	0.01	1011.5	8.5
9	20%	250	1.0	1000	0.9	25	0.01	1013.0	10.4
9	25%	250	1.0	1000	0.9	25	0.01	994.3	7.5
9	25%	250	1.0	1000	0.9	25	0.01	949.6	7.7
9	25%	250	1.0	1000	0.9	25	0.01	950.3	10.6
9	25%	250	1.0	1000	0.9	25	0.01	950.9	9.0
9	25%	250	1.0	1000	0.9	25	0.01	950.9	7.5
9	25%	250	1.0	1000	0.9	25	0.01	956.3	8.7
9	25%	250	1.0	1000	0.9	25	0.01	957.0	10.6
9	25%	250	1.0	1000	0.9	25	0.01	971.7	8.4
9	25%	250	1.0	1000	0.9	25	0.01	971.7	6.3
9	25%	250	1.0	1000	0.9	25	0.01	978.4	9.5
9	30%	250	1.0	1000	0.9	25	0.01	1060.8	9.0
9	30%	250	1.0	1000	0.9	25	0.01	957.8	15.9
9	30%	250	1.0	1000	0.9	25	0.01	961.5	8.0
9	30%	250	1.0	1000	0.9	25	0.01	961.5	8.6
9	30%	250	1.0	1000	0.9	25	0.01	962.2	14.3
9	30%	250	1.0	1000	0.9	25	0.01	963.6	14.4
9	30%	250	1.0	1000	0.9	25	0.01	963.6	12.0
9	30%	250	1.0	1000	0.9	25	0.01	984.6	9.7
9	30%	250	1.0	1000	0.9	25	0.01	995.7	10.8
9	30%	250	1.0	1000	0.9	25	0.01	1014.3	8.8

All scenario trees have been generated by the same seed $m = 1$.

D.2 SA Parameter Tuning

Table D.5: Parameter tuning for simulated annealing algorithm. All combinations of parameters have been run 50 times each. The most important elements of the table are summarized in Table 8.4, Chapter 8.

Parameters				Best Objective				End Objective				*Time [s]*	
T_0	α	I_{max}	T_{min}	$\hat{\mu}_b^o$	$\hat{\sigma}_b^o$	$L_b^{5\%}$	$U_b^{95\%}$	$\hat{\mu}_e^o$	$\hat{\sigma}_e^o$	$L_e^{5\%}$	$U_e^{95\%}$	μ_b^t	μ_r^t
10	0.3	25	0.1	1781	144	1532	1928	1781	144	1535	1928	4	17
10	0.3	25	0.01	1745	148	1491	1928	1747	147	1492	1930	5	31
10	0.3	50	0.1	1728	151	1467	1928	1729	151	1467	1930	6	32
10	0.3	50	0.01	1737	146	1496	1928	1738	146	1496	1931	4	60
10	0.6	25	0.1	1757	164	1427	1928	1757	164	1427	1928	4	38
10	0.6	25	0.01	1738	159	1464	1928	1739	158	1476	1937	5	74
10	0.6	50	0.1	1758	152	1520	1928	1760	151	1533	1939	5	74
10	0.6	50	0.01	1765	125	1565	1928	1765	125	1565	1928	4	145
10	0.9	25	0.1	1736	127	1504	1927	1738	126	1505	1934	4	159
10	0.9	25	0.01	1759	161	1460	1928	1760	161	1463	1942	10	314
10	0.9	50	0.1	1770	172	1456	1928	1770	171	1456	1934	4	314
10	0.9	50	0.01	1748	123	1552	1928	1750	123	1552	1945	6	627
10^2	0.3	25	0.1	1717	156	1464	1928	1781	221	1466	2338	9	20
10^2	0.3	25	0.01	1704	160	1454	1928	1747	177	1478	2153	11	34
10^2	0.3	50	0.1	1671	170	1421	1928	1743	206	1443	2237	13	36
10^2	0.3	50	0.01	1677	153	1450	1928	1741	186	1486	2165	13	64
10^2	0.6	25	0.1	1648	168	1382	1908	1699	193	1384	2360	15	43
10^2	0.6	25	0.01	1675	164	1412	1928	1720	198	1430	2200	15	78
10^2	0.6	50	0.1	1624	171	1401	1928	1700	187	1429	2087	21	82
10^2	0.6	50	0.01	1637	160	1379	1918	1714	169	1431	2119	19	152
10^2	0.9	25	0.1	1608	186	1368	1928	1712	178	1424	2009	24	170
10^2	0.9	25	0.01	1632	152	1435	1928	1695	148	1481	1982	26	325
10^2	0.9	50	0.1	1540	138	1381	1830	1639	128	1471	1928	46	332
10^2	0.9	50	0.01	1601	151	1373	1877	1691	163	1463	2051	39	644
10^3	0.3	25	0.1	1763	142	1528	1928	2649	929	1637	5744	8	24
10^3	0.3	25	0.01	1686	146	1479	1928	1969	429	1518	3454	19	42
10^3	0.3	50	0.1	1755	141	1524	1928	2485	650	1735	4147	15	46
10^3	0.3	50	0.01	1627	151	1423	1899	1789	264	1432	2495	45	81
10^3	0.6	25	0.1	1728	124	1528	1927	2191	541	1651	4220	27	56
10^3	0.6	25	0.01	1614	153	1430	1928	1798	354	1483	3476	46	95
10^3	0.6	50	0.1	1667	133	1460	1921	2041	306	1624	2992	69	111
10^3	0.6	50	0.01	1559	127	1389	1784	1697	219	1424	2312	103	188
10^3	0.9	25	0.1	1537	90	1417	1706	1870	221	1609	2366	160	242
10^3	0.9	25	0.01	1483	91	1350	1639	1628	164	1446	2347	207	411
10^3	0.9	50	0.1	1483	83	1368	1635	1808	215	1450	2292	308	479
10^3	0.9	50	0.01	1467	76	1353	1581	1630	134	1424	1975	402	814
10^4	0.3	25	0.1	1840	115	1610	1928	3751	1265	2193	7395	3	24
10^4	0.3	25	0.01	1807	121	1566	1928	3110	1093	1765	5714	8	46
10^4	0.3	50	0.1	1788	133	1511	1928	4198	1400	2218	7972	7	47
10^4	0.3	50	0.01	1774	128	1601	1928	2657	699	1796	4684	21	90
10^4	0.6	25	0.1	1797	135	1580	1928	4203	1439	2093	7539	7	58
10^4	0.6	25	0.01	1751	143	1519	1928	2276	637	1694	5125	28	110
10^4	0.6	50	0.1	1780	131	1537	1928	4062	1307	1977	7359	17	115
10^4	0.6	50	0.01	1648	137	1474	1897	2068	555	1564	4423	140	219
10^4	0.9	25	0.1	1794	124	1576	1928	4040	1530	2129	7983	30	258
10^4	0.9	25	0.01	1523	94	1364	1668	1830	226	1521	2380	407	504
10^4	0.9	50	0.1	1765	127	1586	1928	3580	1112	2246	6072	151	513
10^4	0.9	50	0.01	1489	94	1337	1662	1835	281	1497	2984	824	993

$\hat{\mu}_b^o$, $\hat{\mu}_e^o$: sample average of best and end objectives, $\hat{\sigma}_b^o$, $\hat{\sigma}_e^o$: empirical standard deviation of best and end objectives, $\hat{\mu}_b^t$, $\hat{\mu}_r^t$: sample average of time to find best objective and of total runtime, $L_b^{5\%}$, $L_e^{5\%}$: lower 5%-ile of best and end objectives, $U_b^{95\%}$, $U_e^{95\%}$: Upper 95%-ile of best and end objectives

Appendix E

Overview of Related Literature

Table E.1: Origin of the papers studied in Table 3.1 and Table 3.2. If found through a survey, the name of the survey is presented in the first column. If found through Google Scholar, the search phrase is presented in the second column.

Article	**Paper**	**Search phrase**
van Oostrum et al. (2008)	Hulshof et al. (2012)	
Adan et al. (2009)	Hulshof et al. (2012)	
Fei et al. (2010)	Samudra et al. (2016)	
Cappanera et al. (2014)	Samudra et al. (2016)	
Mannino et al. (2012)	Samudra et al. (2016)	
Beliën and Demeulemeester (2007)	Samudra et al. (2016)	
Vanberkel et al. (2011)	Samudra et al. (2016)	
Fügener et al. (2014)	Samudra et al. (2016)	
Adan et al. (2011)	Samudra et al. (2016)	
Bruni et al. (2015)	Samudra et al. (2016)	
Lamiri et al. (2009)	Samudra et al. (2016)	
Wang et al. (2014)	Samudra et al. (2016)	
Bovim et al. (2020)		master surgery scheduling stochastic
Fügener (2015)		master surgery scheduling stochastic
M'Hallah and Visintin (2019)		master surgery scheduling stochastic
Batista et al. (2020)		two stage stochastic healthcare
Makboul et al. (2021)		two stage stochastic master surgical scheduling
Neyshabouri and Berg (2017)		two stage stochastic master surgical scheduling
Kamran et al. (2018)		two stage stochastic master surgical scheduling
Heydari and Soudi (2016)		two stage stochastic master surgical scheduling
Spratt and Kozan (2016)	Britt et al. (2021)	
Abdeljaouad et al. (2020)	Britt et al. (2021)	
Visintin et al. (2016)		master surgery scheduling swap
van Essen et al. (2014)	Britt et al. (2021)	
Schneider et al. (2020)	Britt et al. (2021)	
Kumar et al. (2018)		surgery scheduling stochastic
Kim and Mehrotra (2015)		surgery scheduling stochastic two stage
Oliveira et al. (2021)		master surgery scheduling flexibility

Appendix F

Input Data

F.1 5 Groups

The following section defines the sets and parameters used in Chapter 8. Flexibility F, demand scenarios Q_{gc} and scenario probabilities Π_c have been excluded from this section as they change throughout the analysis. Flexibility F, the size of the scenario tree N and its seed m and the bed ward capacity factor β are reported along with results.

Sets

Wards:	$\mathcal{W} =$	$\{MC, IC\}$
Specialties:	$\mathcal{S} =$	$\{GN, GO, UR, KA, EN\}$
Groups:	$\mathcal{G} =$	$\{GN\text{-}a, GO\text{-}a, UR\text{-}a, KA\text{-}a, EN\text{-}a, \}$
Days:	$\mathcal{D} =$	$\{1, 2, ..28\}$

Table F.1: $\mathcal{G}_w^{\mathcal{W}}$: Surgery groups that can receive postoperative care at ward w, indexed g. Each row represents one set $\mathcal{G}_w^{\mathcal{W}}$ for the ward w and 1 indicates that the group g is included in the set.

Ward \ Group		GN-a	GO-a	UR-a	KA-a	EN-a
	$w \backslash g$	*1*	*2*	*3*	*4*	*5*
MC	*1*	1	1	1	1	1
IC	*2*	1	1	1	1	1

Table F.2: $\mathcal{R}_s^{\mathcal{S}}$: ORs suitable for specialty s, indexed r. Each row represents one set $\mathcal{R}_s^{\mathcal{S}}$ for the specialty s and 1 indicates that the room r is included in the set.

Specialty \ Room		GA-1	GA-2	GA-3	GA-4	GA-5	GA-6	GA-7
	r\s	*1*	*2*	*3*	*4*	*5*	*6*	*7*
GN	*1*				1	1		1
GO	*2*				1	1		1
UR	*3*	1	1	1			1	
KA	*4*		1					
EN	*5*	1	1					

Parameters

Cleaning time:	$T^C =$	30
Number of cycles in planning period:	$I =$	2
Additional time with extended slots:	$E =$	90

Table F.3: C_g: Unit cost of not meeting the demand of surgery group g.

Cost		GN-a	GO-a	UR-a	KA-a	EN-a
	$w \backslash g$	*1*	*2*	*3*	*4*	*5*
C_g		155	146	253.05	246	173

Table F.4: L_g^{SD}: Surgery duration of a patient in surgery group g.

Duration		GN-a	GO-a	UR-a	KA-a	EN-a
	$w \backslash g$	*1*	*2*	*3*	*4*	*5*
L^{SD}		125	116	223.05	216	143

Table F.5: T_g: Monthly expected demand of for surgery group g.

E[Demand]		GN-a	GO-a	UR-a	KA-a	EN-a
	g	*1*	*2*	*3*	*4*	*5*
T_g		62	44	58	4	56

Table F.6: N_d: Total number of available ORs on day d.

Day	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	...	Sun
d	1	2	3	4	5	6	7	8	...	28
N	7	7	7	7	7	0	0	7	...	0

Table F.7: U_s^X: Maximum number of times a specialty may extend its opening hours during a cycle.

Specialty	GN	GO	UR	KA	EN
s	*1*	*2*	*3*	*4*	*5*
H	8	10	6	1	3

Table F.8: K_{sd}: Number of surgical teams from specialty s available on day d.

Day / Specialty		Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	...	Sun
	$s \backslash d$	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	...	*28*
GN	*1*	2	2	2	2	2	0	0	2	...	0
GO	*2*	2	2	2	2	2	0	0	2	...	0
UR	*3*	4	4	4	4	4	0	0	4	...	0
KA	*4*	2	2	2	2	2	0	0	2	...	0
EN	*5*	2	2	2	2	2	0	0	2	...	0

Table F.9: H_d: Default number of available minutes in a slot on day d.

Day	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	...	Sun
d	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	...	*28*
H	450	450	450	450	450	0	0	450	...	0

Table F.10: B_{wd}: Number of available beds at ward w on the night following day d.

Day / Ward		Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	...	Sun
	$w \backslash d$	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	...	*28*
MC	*1*	60	60	60	60	49	49	49	60	...	49
IC	*2*	11	11	11	11	6	6	6	11		6

Table F.11: J_w: Maximum number of nights a patient may stay in ward w.

Ward	MC	IC
w	*1*	*2*
J	20	2

Table F.12: P_{wgd}: Probability that a patient from surgery group g occupies a bed in ward w, on the night d days after surgery. The equivalent probabilities for the cutting stock formulation can be translated from this table with the formula: $\mathrm{P}^{CS}_{mwd} = \sum_{g \in \mathcal{G}} A_{mg} * P_{gwd} \; \forall \; m \in \mathcal{M}, w \in \mathcal{W}$.

MC	$w \backslash d$	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
GN-a	*1*	0.6281	0.539	0.4791	0.4499	0.3872	0.3357	0.2744	0.2131	0.1616	0.1337	0.1003	0.0864	0.0669	0.0418	0.0292	0.0209	0.0181	0.0125	0.0097	0.0014
GO-a	*2*	0.6158	0.5188	0.3861	0.3347	0.297	0.2713	0.2317	0.2	0.1505	0.1129	0.0911	0.0772	0.0673	0.0634	0.0475	0.0317	0.0257	0.0178	0.0079	0.004
UR-a	*3*	0.6182	0.4163	0.2612	0.1935	0.1459	0.1075	0.0827	0.0632	0.0495	0.0352	0.0287	0.0228	0.0176	0.0117	0.0085	0.0072	0.0065	0.0046	0.0033	0.0007
KA-a	*4*	0.4286	0.1224	0.0816	0.0612	0.0408	0.0408	0.0408	0.0408	0.0408	0.0408	0.0204	0.0204	0	0	0	0	0	0	0	0
EN-a	*5*	0.6035	0.2954	0.1201	0.0742	0.0316	0.0205	0.019	0.0095	0.0047	0.0047	0.0032	0.0032	0.0032	0.0016	0	0	0	0	0	0

IC	$w \backslash d$	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
GN-a	*1*	0.4499	0.2312	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GO-a	*2*	0.3347	0.1743	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
UR-a	*3*	0.1935	0.0977	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KA-a	*4*	0.0612	0.0204	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
EN-a	*5*	0.0742	0.0427	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table F.13: Y_{wd}: Bed occupation from last planning period at ward w, d days into current planning period.

Ward	$w \backslash d$	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
MC	*1*	15.1341	12.0295	9.505	7.4072	5.7522	4.441	3.4253	2.601	1.9586	1.4422	1.0514	0.7689	0.5344	0.3519	0.1955	0.0869	0.0174	0	0	0
IC	*2*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table F.14: Patterns data. Set of patterns $\mathcal{M}$ with indication of whether they are in the subset of extended $\mathcal{M}^X$ or the non-extended patterns $\mathcal{M}^{NX}$ and which specialty they belong to $\mathcal{M}^{\mathcal{S}}_s$. A_{mg} is the numbers in the matrix indicating how many operations from group g that are included in pattern m.

$\mathcal{G} \backslash \mathcal{M}$	$g \backslash \ m$	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*
GN-a	*1*	1	2								3			
GO-a	*2*			1	2	3								
UR-a	*3*						1					2		
KA-a	*4*							1					2	
EN-a	*5*								1	2				3
Duration		155.0	310.0	146.0	292.0	438.0	253.05	246.0	173.0	346.0	465.0	506.1	492.0	519.0
Extended		0	0	0	0	0	0	0	0	0	1	1	1	1
	s	*1*	*1*	*2*	*2*	*2*	*3*	*4*	*5*	*5*	*1*	*3*	*4*	*5*
Specialty		**GN**	**GN**	**GO**	**GO**	**GO**	**UR**	**KA**	**EN**	**EN**	**GN**	**UR**	**KA**	**EN**

F.2 9 Groups

The following section defines the sets and parameters used in Chapter 8. Flexibility F, demand scenarios Q_{gc} and scenario probabilities Π_c have been excluded from this section as they change throughout the analysis. Flexibility F, the size of the scenario tree N and its seed m and the bed ward capacity factor β are reported along with results.

Sets

Wards:	$\mathcal{W} =$	$\{MC, IC\}$
Specialties:	$\mathcal{S} =$	$\{GN, GO, UR, KA, EN\}$
Groups:	$\mathcal{G} =$	$\{GN\text{-}a, GN\text{-}b, GO\text{-}a, GO\text{-}b, UR\text{-}a, UR\text{-}b, KA\text{-}a, EN\text{-}a, EN\text{-}b\}$
Days:	$\mathcal{D} =$	$\{1, 2, ..28\}$

Table F.15: $\mathcal{G}_w^{\mathcal{W}}$: Surgery groups that can receive postoperative care at ward w, indexed g. Each row represents one set $\mathcal{G}_w^{\mathcal{W}}$ for the ward w and 1 indicates that the group g is included in the set.

Group / Ward		GN-a	GN-b	GO-a	GO-b	UR-a	UR-b	KA-a	EN-a	EN-b
	$w \backslash g$	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
MC	*1*	1	1	1	1	1	1	1	1	1
IC	*2*	1	1	1	1	1	1	1	1	1

Table F.16: $\mathcal{R}_s^{\mathcal{S}}$: ORs suitable for specialty s, indexed r. Each row represents one set $\mathcal{R}_s^{\mathcal{S}}$ for the specialty s and 1 indicates that the room r is included in the set.

Room / Specialty		GA-1	GA-2	GA-3	GA-4	GA-5	GA-6	GA-7
	r\s	*1*	*2*	*3*	*4*	*5*	*6*	*7*
GN	*1*				1	1		1
GO	*2*				1	1		1
UR	*3*	1	1	1			1	
KA	*4*		1					
EN	*5*	1	1					

Parameters

Cleaning time:	$T^C =$	30
Number of cycles in planning period:	$I =$	2
Additional time with extended slots:	$E =$	90

Table F.17: C_g: Unit cost of not meeting the demand of surgery group g.

Cost		GN-a	GN-b	GO-a	GO-b	UR-a	UR-b	KA-a	EN-a	EN-b
	$w \backslash g$	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
C_g		344	124	321.25	184	287.45	130	155	171	243.6

Table F.18: L_g^{SD}: Surgery duration of a patient in surgery group g.

Duration		GN-a	GN-b	GO-a	GO-b	UR-a	UR-b	KA-a	EN-a	EN-b
	$w \backslash g$	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
C_g		344	124	321.25	184	287.45	130	155	171	243.6

Table F.19: T_g: Monthly expected demand for surgery group g.

E[Demand]		GN-a	GN-b	GO-a	GO-b	UR-a	UR-b	KA-a	EN-a	EN-b
	g	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
T_g		26	34	16	26	14	118	4	52	2

Table F.20: N_d: Total number of available ORs on day d.

Day	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	...	Sun
d	1	2	3	4	5	6	7	8	...	28
N	7	7	7	7	7	0	0	7	...	0

Table F.21: U_s^X: Maximum number of times a specialty may extend its opening hours during a cycle.

Specialty	GN	GO	UR	KA	EN
s	*1*	*2*	*3*	*4*	*5*
H	8	10	6	1	3

Table F.22: K_{sd}: Number of surgical teams from specialty s available on day d.

Specialty \ Day		**Mon**	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**	**Sun**	**Mon**	**...**	**Sun**
	s\d	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	...	*28*
GN	*1*	2	2	2	2	2	0	0	2	...	0
GO	*2*	2	2	2	2	2	0	0	2	...	0
UR	*3*	4	4	4	4	4	0	0	4	...	0
KA	*4*	2	2	2	2	2	0	0	2	...	0
EN	*5*	2	2	2	2	2	0	0	2	...	0

Table F.23: H_d: Default number of available minutes in a slot on day d.

Day	**Mon**	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**	**Sun**	**Mon**	**...**	**Sun**
d	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	...	*28*
H	450	450	450	450	450	0	0	450	...	0

Table F.24: B_{wd}: Number of available beds at ward w on the night following day d.

Ward \ Day		**Mon**	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**	**Sun**	**Mon**	**...**	**Sun**
	w\d	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	...	*28*
MC	*1*	60	60	60	60	49	49	49	60	...	49
IC	*2*	11	11	11	11	6	6	6	11		6

Table F.25: J_w: Maximum number of nights a patient may stay in ward w.

Ward	**MC**	**IC**
w	*1*	*2*
J	20	2

Table F.26: P_{wgd}: Probability that a patient from surgery group g occupies a bed in ward w, on the night d days after surgery. The equivalent probabilities for the cutting stock formulation can be translated from this table with the formula: $\mathrm{P}^{CS}_{mwd} = \sum_{g \in \mathcal{G}} A_{mg} * P_{gwd} \; \forall \; m \in \mathcal{M}, w \in \mathcal{W}$.

MC	$w \backslash d$	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
GN-a	*1*	0.9811	0.959	0.9401	0.9306	0.8233	0.735	0.6088	0.4732	0.3596	0.2997	0.224	0.1924	0.1483	0.0946	0.0662	0.0473	0.041	0.0284	0.0221	0.0032
GN-b	*2*	0.3491	0.207	0.1147	0.0698	0.0424	0.02	0.01	0.0075	0.005	0.0025	0.0025	0.0025	0.0025	0	0	0	0	0	0	0
GO-a	*3*	0.957	0.9194	0.8172	0.7796	0.7043	0.6398	0.5538	0.4731	0.3495	0.2634	0.2043	0.1667	0.1398	0.1344	0.0968	0.0753	0.0645	0.043	0.0215	0.0108
GO-b	*4*	0.4169	0.2853	0.1348	0.0752	0.0596	0.0564	0.0439	0.0408	0.0345	0.0251	0.0251	0.0251	0.0251	0.0219	0.0188	0.0063	0.0031	0.0031	0	0
UR-a	*5*	0.9885	0.9828	0.9195	0.8793	0.6839	0.546	0.4368	0.3276	0.2471	0.1552	0.1322	0.092	0.069	0.046	0.0287	0.023	0.0172	0.0172	0.0057	0
UR-b	*6*	0.5709	0.3439	0.1771	0.1058	0.0771	0.0514	0.0375	0.0294	0.0242	0.0198	0.0154	0.014	0.011	0.0073	0.0059	0.0051	0.0051	0.0029	0.0029	0.0007
KA-a	*7*	0.4286	0.1224	0.0816	0.0612	0.0408	0.0408	0.0408	0.0408	0.0408	0.0408	0.0204	0.0204	0	0	0	0	0	0	0	0
EN-a	*8*	0.5885	0.2689	0.0869	0.041	0.0164	0.0115	0.0098	0.0049	0.0033	0.0033	0.0016	0.0016	0.0016	0	0	0	0	0	0	0
EN-b	*9*	1	1	1	0.9565	0.4348	0.2609	0.2609	0.1304	0.0435	0.0435	0.0435	0.0435	0.0435	0.0435	0	0	0	0	0	0

IC	$w \backslash d$	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
GN-a	*1*	0.9306	0.489	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GN-b	*2*	0.0698	0.0274	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GO-a	*3*	0.7796	0.4355	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GO-b	*4*	0.0752	0.0219	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
UR-a	*5*	0.8793	0.4655	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
UR-b	*6*	0.1058	0.0507	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KA-a	*7*	0.0612	0.0204	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
EN-a	*8*	0.041	0.023	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
EN-b	*9*	0.9565	0.5652	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table F.27: Y_{wd}: Bed occupation from last planning period at ward w d days into current planning period.

Ward	$w \backslash d$	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
MC	*1*	14.5328	11.5499	9.1267	7.1145	5.5274	4.269	3.2942	2.5017	1.8855	1.3898	1.0143	0.7426	0.5168	0.3405	0.1891	0.0843	0.017	0	0	0
IC	*2*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table F.28: Patterns data. Set of patterns $\mathcal{M}$ with indication of whether they are in the subset of extended $\mathcal{M}^X$ or the non-extended patterns $\mathcal{M}^{NX}$ and which specialty they belong to $\mathcal{M}^{\mathcal{S}}_s$. A_{mg} is the numbers in the matrix indicating how many operations from group g that are included in pattern m.

$\mathcal{G} \backslash \mathcal{M}$		*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	*21*	*22*	*23*	*24*	*25*
GN-a	*1*	1																		1						
GN-b	*2*		1	2	3															1	4					
GO-a	*3*					1																1				
GO-b	*4*						1	2														1				
UR-a	*5*								1		1															
UR-b	*6*									1	1	2	3										4			
KA-a	*7*													1	2									3		
EN-a	*8*															1		2	1							3
EN-b	*9*																1		1						2	
Duration		344	124	248	372	321.3	184	368	287.5	130	417.5	260	390	155	310	171	243	342	414.6	468	496	505.3	520	465.	487.2	513.0
Extended		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
	s	*1*	*1*	*1*	*1*	*2*	*2*	*2*	*3*	*3*	*3*	*3*	*3*	*4*	*4*	*5*	*5*	*5*	*5*	*1*	*1*	*2*	*3*	*4*	*5*	*5*
Specialty		**GN**	**GN**	**GN**	**GN**	**GO**	**GO**	**GO**	**UR**	**UR**	**UR**	**UR**	**UR**	**KA**	**KA**	**EN**	**EN**	**EN**	**EN**	**GN**	**GN**	**GO**	**UR**	**KA**	**EN**	**EN**

F.3 25 Groups

The following section defines the sets and parameters used in Chapter 8. Flexibility F, demand scenarios Q_{gc} and scenario probabilities Π_c have been excluded from this section as they change throughout the analysis. Flexibility F, the size of the scenario tree N and its seed m and the bed ward capacity factor β are reported along with result.

Sets

Wards:	$\mathcal{W} =$	$\{MC, IC\}$
Specialties:	$\mathcal{S} =$	$\{GN, GO, UR, KA, EN\}$
Groups:	$\mathcal{G} =$	{*GN-a, GN-b, GN-c, GN-d, GN-e, GN-f, GO-a, GO-b, GO-c, GO-d, GO-e, GO-f, UR-a, UR-b, UR-c, UR-d, UR-e, UR-f, UR-g, KA-a, EN-a, EN-b, EN-c, EN-d, EN-e*}
Days:	$\mathcal{D} =$	$\{1, 2, ..28\}$

Table F.29: $\mathcal{G}_w^{\mathcal{W}}$: Surgery groups that can receive postoperative care at ward w, indexed g. Each row represents one set $\mathcal{G}_w^{\mathcal{W}}$ for the ward w and 1 indicates that the group g is included in the set.

ward/group		GN-a	GN-b	GN-c	GN-d	GN-e	GN-f	GO-a	GO-b	GO-c	GO-d	GO-e	GO-f	
	w\g	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	
MC	*1*	1	1	1	1	1	1	1	1	1	1	1	1	
IC	*2*	1	1	1	1	1	1	1	1	1	1	1	1	

ward/group		UR-a	UR-b	UR-c	UR-d	UR-e	UR-f	UR-g	KA-a	EN-a	EN-b	EN-c	EN-d	EN-e
	w\g	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	*21*	*22*	*23*	*24*	*25*
MC	*1*	1	1	1	1	1	1	1	1	1	1	1	1	1
IC	*2*	1	1	1	1	1	1	1	1	1	1	1	1	1

Table F.30: $\mathcal{R}_s^{\mathcal{S}}$: ORs suitable for specialty s, indexed r. Each row represents one set $\mathcal{R}_s^{\mathcal{S}}$ for the specialty s and 1 indicates that the room r is included in the set.

Room / Specialty		GA-1	GA-2	GA-3	GA-4	GA-5	GA-6	GA-7
	r\s	*1*	*2*	*3*	*4*	*5*	*6*	*7*
GN	*1*				1	1		1
GO	*2*				1	1		1
UR	*3*	1	1	1			1	
KA	*4*		1					
EN	*5*	1	1					

Parameters

Cleaning time:	$T^C =$	30
Number of cycles in planning period:	$I =$	2
Additional time with extended slots:	$E =$	90

Table F.31: C_g: Unit cost of not meeting the demand of surgery group g.

Cost		GN-a	GN-b	GN-c	GN-d	GN-e	GN-f	GO-a	GO-b	GO-c	GO-d	GO-e	GO-f	
	w\g	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	
C_g	*1*	94	302.5	378.5	382	238	427	97	241	218.5	455	265	302	

Cost		UR-a	UR-b	UR-c	UR-d	UR-e	UR-f	UR-g	KA-a	EN-a	EN-b	EN-c	EN-d	EN-e
	w\g	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	*21*	*22*	*23*	*24*	*25*
C_g	*1*	256	240	91.5	241	136	335	82.5	136	172	164	116	217	255.5

Table F.32: L_g^{SD}: Surgery duration of a patient in surgery group g.

Duration		GN-a	GN-b	GN-c	GN-d	GN-e	GN-f	GO-a	GO-b	GO-c	GO-d	GO-e	GO-f	
	w\g	1	2	3	4	5	6	7	8	9	10	11	12	
L_g^{SD}	1	64	272.5	348.5	352	208	397	67	211	188.5	425	235	272	

Duration		UR-a	UR-b	UR-c	UR-d	UR-e	UR-f	UR-g	KA-a	EN-a	EN-b	EN-c	EN-d	EN-e
	w\g	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	*21*	*22*	*23*	*24*	*25*
L_g^{SD}	1	226	210	61.5	211	106	305	52.5	106	142	134	86	187	225.5

Table F.33: T_g: Monthly expected demand of patient in surgery group g.

E[Demand]		GN-a	GN-b	GN-c	GN-d	GN-e	GN-f	GO-a	GO-b	GO-c	GO-d	GO-e	GO-f	
	g	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	
T_g		38	8	3	3	22	3	22	14	8	3	5	3	

E[Demand]		UR-a	UR-b	UR-c	UR-d	UR-e	UR-f	UR-g	KA-a	EN-a	EN-b	EN-c	EN-d	EN-e
	g	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	*21*	*22*	*23*	*24*	*25*
T_g		8	5	59	11	57	3	30	5	24	11	24	3	5

Table F.34: N_d: Total number of available ORs on day d.

Day	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	...	Sun
d	1	2	3	4	5	6	7	8	...	28
N	7	7	7	7	7	0	0	7	...	0

Table F.35: U_s^X: Maximum number of times a specialty may extend its opening hours during a cycle.

Specialty	GN	GO	UR	KA	EN
s	*1*	*2*	*3*	*4*	*5*
H	8	10	6	1	3

Table F.36: K_{sd}: Number of surgical teams from specialty s available on day d.

Day / Specialty		Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	...	Sun
	s\d	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	...	*28*
GN	*1*	2	2	2	2	2	0	0	2	...	0
GO	*2*	2	2	2	2	2	0	0	2	...	0
UR	*3*	4	4	4	4	4	0	0	4	...	0
KA	*4*	2	2	2	2	2	0	0	2	...	0
EN	*5*	2	2	2	2	2	0	0	2	...	0

Table F.37: H_d: Default number of available minutes in a slot on day d.

Day	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	...	Sun
d	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	...	*28*
H	450	450	450	450	450	0	0	450	...	0

Table F.38: B_{wd}: Number of available beds at ward w on the night following day d.

Day / Ward		Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	...	Sun
	w\d	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	...	*28*
MC	*1*	60	60	60	60	49	49	49	60	...	49
IC	*2*	11	11	11	11	6	6	6	11		6

Table F.39: J_w: Maximum number of nights a patient may stay in ward w.

Ward	MC	IC
w	*1*	*2*
J	20	2

Table F.40: P_{wgd}: Probability that a patient from surgery group g occupies a bed in ward w, on the night d days after surgery. The equivalent probabilities for the cutting stock formulation can be translated from this table with the formula: $\mathrm{P}^{CS}_{mwd} = \sum_{g \in \mathcal{G}} A_{mg} * P_{gwd} \; \forall \; m \in \mathcal{M}, w \in \mathcal{W}$.

MC	$g\backslash d$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
GN-a	*1*	0.2831	0.1292	0.0677	0.0246	0.0092	0.0062	0.0031	0.0031	0	0	0	0	0	0	0	0	0	0	0	0
GN-b	*2*	0.9875	0.9875	0.9875	0.9875	0.8125	0.6875	0.55	0.4	0.25	0.1625	0.125	0.1125	0.1125	0.0875	0.05	0.0375	0.0375	0.0125	0.0125	0
GN-c	*3*	0.5667	0.5	0.0667	0.0333	0.0333	0.0333	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GN-d	*4*	1	1	0.9706	0.9706	0.9706	0.9706	0.9706	0.9118	0.8235	0.7941	0.6765	0.6176	0.4118	0.3235	0.1765	0.0882	0.0882	0.0588	0.0588	0
GN-e	*5*	0.9286	0.8724	0.8265	0.7959	0.6633	0.5357	0.3776	0.25	0.1735	0.1276	0.0714	0.0612	0.051	0.0153	0.0153	0.0153	0.0153	0.0153	0.0102	0.0051
GN-f	*6*	1	1	1	1	1	1	1	0.8837	0.7442	0.6744	0.5581	0.4651	0.3488	0.2093	0.186	0.1395	0.093	0.0698	0.0465	0
GO-a	*7*	0.3424	0.1957	0.1087	0.0652	0.0489	0.0489	0.0435	0.0435	0.038	0.0272	0.0272	0.0272	0.0272	0.0217	0.0163	0	0	0	0	0
GO-b	*8*	0.906	0.8547	0.6752	0.6154	0.5385	0.4786	0.3761	0.3504	0.2479	0.1624	0.1026	0.0598	0.0427	0.0427	0.0427	0.0427	0.0256	0.0171	0	0
GO-c	*9*	0.5116	0.3721	0.1744	0.0465	0.0349	0.0349	0.0233	0.0116	0.0116	0.0116	0.0116	0.0116	0.0116	0.0116	0.0116	0	0	0	0	0
GO-d	*10*	0.4063	0.3438	0.0313	0.0313	0.0313	0.0313	0.0313	0.0313	0.0313	0.0313	0.0313	0.0313	0.0313	0.0313	0.0313	0.0313	0.0313	0.0313	0	0
GO-e	*11*	0.9796	0.9388	0.8776	0.8776	0.8571	0.7755	0.7347	0.5714	0.449	0.3673	0.3469	0.3265	0.3265	0.3265	0.2041	0.1429	0.1224	0.0816	0.0408	0
GO-f	*12*	1	1	1	1	0.8649	0.8108	0.7027	0.5946	0.4324	0.3514	0.2703	0.2432	0.1622	0.1351	0.1081	0.0811	0.0811	0.0541	0.0541	0.0541
UR-a	*13*	0.7143	0.6	0.3143	0.0571	0.0429	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
UR-b	*14*	1	1	1	0.9783	0.7609	0.587	0.4565	0.2609	0.1957	0.0652	0.0652	0.0217	0	0	0	0	0	0	0	0
UR-c	*15*	0.1336	0.0382	0.0191	0.0095	0.0038	0.0038	0	0	0	0	0	0	0	0	0	0	0	0	0	0
UR-d	*16*	0.981	0.9619	0.8476	0.7714	0.5333	0.3333	0.2095	0.1238	0.0571	0.0286	0.0286	0.019	0.0095	0.0095	0.0095	0	0	0	0	0
UR-e	*17*	0.8878	0.495	0.1804	0.0661	0.0441	0.024	0.02	0.018	0.018	0.012	0.01	0.008	0.008	0.006	0.002	0.002	0.002	0.002	0.002	0
UR-f	*18*	1	1	0.9512	0.9512	0.9512	0.9512	0.9512	0.9024	0.7561	0.561	0.439	0.3171	0.2683	0.1707	0.0976	0.0732	0.0488	0.0488	0	0
UR-g	*19*	0.7692	0.5577	0.4154	0.3577	0.2692	0.2	0.1423	0.1077	0.0885	0.0808	0.0615	0.0577	0.0423	0.0269	0.0269	0.0269	0.0269	0.0154	0.0154	0.0038
KA-a	*20*	0.4286	0.1224	0.0816	0.0612	0.0408	0.0408	0.0408	0.0408	0.0408	0.0408	0.0204	0.0204	0	0	0	0	0	0	0	0
EN-a	*21*	0.9058	0.5067	0.1973	0.0807	0.0269	0.0224	0.0179	0.0135	0.009	0.009	0.0045	0.0045	0.0045	0	0	0	0	0	0	0
EN-b	*22*	0.2072	0.009	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
EN-c	*23*	0.5	0.1743	0.0321	0.0275	0.0138	0.0046	0.0046	0	0	0	0	0	0	0	0	0	0	0	0	0
EN-d	*24*	1	0.963	0.8889	0.8519	0.4074	0.2593	0.2593	0.1111	0.037	0.037	0.037	0.037	0.037	0.037	0	0	0	0	0	0
EN-e	*25*	0.3889	0.1667	0.0185	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IC	$g\backslash d$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
GN-a	*1*	0.0246	0.0123	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GN-b	*2*	0.9875	0.85	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GN-c	*3*	0.0333	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GN-d	*4*	0.9706	0.0588	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GN-e	*5*	0.7959	0.2653	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GN-f	*6*	1	0.9302	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GO-a	*7*	0.0652	0.0217	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GO-b	*8*	0.6154	0.2735	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GO-c	*9*	0.0465	0.0116	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GO-d	*10*	0.0313	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GO-e	*11*	0.8776	0.2857	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GO-f	*12*	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
UR-a	*13*	0.0571	0.0143	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
UR-b	*14*	0.9783	0.913	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
UR-c	*15*	0.0095	0.0038	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
UR-d	*16*	0.7714	0.2667	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
UR-e	*17*	0.0661	0.0341	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
UR-f	*18*	0.9512	0.3415	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
UR-g	*19*	0.3577	0.1769	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KA-a	*20*	0.0612	0.0204	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
EN-a	*21*	0.0807	0.0404	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
EN-b	*22*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
EN-c	*23*	0.0275	0.0183	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
EN-d	*24*	0.8519	0.5185	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
EN-e	*25*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table F.41: Y_{wd}: Bed occupation from last planning period at ward w d days into current planning period.

Ward	$w\backslash d$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MC	*1*	12.7463	10.0159	7.8196	6.0185	4.6257	3.5457	2.7226	2.0669	1.5641	1.1611	0.8521	0.6264	0.4366	0.288	0.1591	0.0709	0.0137	0	0	0
IC	*2*	0.0246	0.0123	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table F.42: Patterns data. Set of patterns $\mathcal{M}$ with indication of whether they are in the subset of extended $\mathcal{M}^X$ or the non-extended patterns $\mathcal{M}^{NX}$ and which specialty they belong to $\mathcal{M}_s^{\mathcal{S}}$. A_{mg} is the numbers in the matrix indicating how many operations from group g that are included in pattern m.

$\mathcal{G} \backslash \mathcal{M}$	$g\backslash m$	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	*21*	*22*	*23*	*24*	*25*
GN-a	*1*	1						2	1	1	3	2	4													
GN-b	*2*		1						1																	
GN-c	*3*			1																						
GN-d	*4*				1																					
GN-e	*5*					1				1		1														
GN-f	*6*						1																			
GO-a	*7*													1					2	1	1	1	1		3	2
GO-b	*8*														1					1						1
GO-c	*9*															1					1			2		
GO-d	*10*																									
GO-e	*11*																1					1				
GO-f	*12*																	1					1			
UR-a	*13*																									
UR-b	*14*																									
UR-c	*15*																									
UR-d	*16*																									
UR-e	*17*																									
UR-f	*18*																									
UR-g	*19*																									
KA-a	*20*																									
EN-a	*21*																									
EN-b	*22*																									
EN-c	*23*																									
EN-d	*24*																									
EN-e	*25*																									
Duration		94	302.5	378.5	382	238	427	188	396.5	332	282	426	376	97	241	218.5	265	302	194	338	315.5	362	399	437	291	435
Extended		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	s	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*2*	*2*	*2*	*2*	*2*	*2*	*2*	*2*	*2*	*2*	*2*	*2*	*2*
Specialty		**GN**	**GN**	**GN**	**GN**	**GN**	**GN**	**GN**	**GN**	**GN**	**GN**	**GN**	**GN**	**GO**	**GO**	**GO**	**GO**	**GO**	**GO**	**GO**	**GO**	**GO**	**GO**	**GO**	**GO**	**GO**

$\mathcal{G} \backslash \mathcal{M}$	$g\backslash m$	*26*	*27*	*28*	*29*	*30*	*31*	*32*	*33*	*34*	*35*	*36*	*37*	*38*	*39*	*40*	*41*	*42*	*43*	*44*	*45*	*46*	*47*	*48*	*49*	*50*
GN-a	*1*																									
GN-b	*2*																									
GN-c	*3*																									
GN-d	*4*																									
GN-e	*5*																									
GN-f	*6*																									
GO-a	*7*	2	4																							
GO-b	*8*																									
GO-c	*9*	1																								
GO-d	*10*																									
GO-e	*11*																									
GO-f	*12*																									
UR-a	*13*			1							1	1	1													
UR-b	*14*				1									1	1	1										
UR-c	*15*					1					1			1			2	1	1	1	1					
UR-d	*16*						1											1				1	1			
UR-e	*17*							1				1			1				1			1		2	1	
UR-f	*18*								1											1						1
UR-g	*19*									1			1			1					1		1		1	1
KA-a	*20*																									
EN-a	*21*																									
EN-b	*22*																									
EN-c	*23*																									
EN-d	*24*																									
EN-e	*25*																									
Duration		412.5	388	256	240	91.5	241	136	335	82.5	347.5	392	338.5	331.5	376	322.5	183	332.5	227.5	426.5	174	377	323.5	272	218.5	417.5
Extended		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	s	*2*	*2*	*3*	*3*	*3*	*3*	*3*	*3*	*3*	*3*	*3*	*3*	*3*	*3*	*3*	*3*	*3*	*3*	*3*	*3*	*3*	*3*	*3*	*3*	*3*
Specialty		**GO**	**GO**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**

Table F.43: Patterns data. Set of patterns $\mathcal{M}$ with indication of whether they are in the subset of extended $\mathcal{M}^X$ or the non-extended patterns $\mathcal{M}^{NX}$ and which specialty they belong to $\mathcal{M}_s^{\mathcal{S}}$. A_{mg} is the numbers in the matrix indicating how many operations from group g that are included in pattern m.

$\mathcal{G} \setminus \mathcal{M}$	$g \setminus m$	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
GN-a	1																									
GN-b	2																									
GN-c	3																									
GN-d	4																									
GN-e	5																									
GN-f	6																									
GO-a	7																									
GO-b	8																									
GO-c	9																									
GO-d	10																									
GO-e	11																									
GO-f	12																									
UR-a	13		1	1	1																					
UR-b	14					1	1	1																		
UR-c	15		2	1		2	1		3	2	2	2	1	1	1	1						4	3	3	2	2
UR-d	16									1			1				1									
UR-e	17										1			2	1			3	2	1			1		1	
UR-f	18																									
UR-g	19	2		1	2		1	2				1	1		1	2	2		1	2	3			1	1	2
KA-a	20																									
EN-a	21																									
EN-b	22																									
EN-c	23																									
EN-d	24																									
EN-e	25																									
Duration		165	439	430	421	423	414	405	274.5	424	319	265.5	415	363.5	310	256.5	406	408	354.5	301	247.5	366	410.5	357	401.5	348
Extended		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	s	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Specialty		**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**

$\mathcal{G} \setminus \mathcal{M}$	$g \setminus m$	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
GN-a	1																									
GN-b	2																									
GN-c	3																									
GN-d	4																									
GN-e	5																									
GN-f	6																									
GO-a	7																									
GO-b	8																									
GO-c	9																									
GO-d	10																									
GO-e	11																									
GO-f	12																									
UR-a	13																									
UR-b	14																									
UR-c	15	1	1	1				4	3	2	1															
UR-d	16																									
UR-e	17	2	1		2	1																				
UR-f	18																									
UR-g	19	1	2	3	2	3	4	1	2	3	4	5														
KA-a	20												1	2	3											
EN-a	21															1					2	1	1	1	1	
EN-b	22																1					1				2
EN-c	23																	1					1			
EN-d	24																		1					1		
EN-e	25																			1					1	
Duration		446	392.5	339	437	383.5	330	448.5	439.5	430.5	421.5	412.5	136	272	408	172	164	116	217	255.5	344	336	288	389	427.5	328
Extended		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	s	3	3	3	3	3	3	3	3	3	3	3	4	4	4	5	5	5	5	5	5	5	5	5	5	5
Specialty		**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**KA**	**KA**	**KA**	**EN**	**EN**	**EN**	**EN**	**EN**	**EN**	**EN**	**EN**	**EN**	**EN**	**EN**

Table F.44: Patterns data. Set of patterns $\mathcal{M}$ with indication of whether they are in the subset of extended $\mathcal{M}^X$ or the non-extended patterns $\mathcal{M}^{NX}$ and which specialty they belong to $\mathcal{M}^{\mathcal{S}}_s$. A_{mg} is the numbers in the matrix indicating how many operations from group g that are included in pattern m.

$\mathcal{G} \setminus \mathcal{M}$	$g \setminus m$	*101*	*102*	*103*	*104*	*105*	*106*	*107*	*108*	*109*	*110*	*111*	*112*	*113*	*114*	*115*	*116*	*117*	*118*	*119*	*120*	*121*	*122*	*123*	*124*	*125*
GN-a	*1*													1	1	1		2	3	5						
GN-b	*2*																	1								
GN-c	*3*													1												
GN-d	*4*														1											
GN-e	*5*																2		1							
GN-f	*6*															1										
GO-a	*7*																									
GO-b	*8*																					2	1	1		
GO-c	*9*																						1		1	1
GO-d	*10*																				1					
GO-e	*11*																							1	1	
GO-f	*12*																									1
UR-a	*13*																									
UR-b	*14*																									
UR-c	*15*																									
UR-d	*16*																									
UR-e	*17*																									
UR-f	*18*																									
UR-g	*19*																									
KA-a	*20*																									
EN-a	*21*								1																	
EN-b	*22*	1	1	1						2	1															
EN-c	*23*	1			2	1	1		2	1	2	3	2													
EN-d	*24*		1			1		2					1													
EN-e	*25*			1			1																			
Duration		280	381	419.5	232	333	371.5	434	404	444	396	348	449	472.5	476	521	476	490.5	520	470	455	482	459.5	506	483.5	520.5
Extended		0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
	s	*5*	*5*	*5*	*5*	*5*	*5*	*5*	*5*	*5*	*5*	*5*	*5*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*2*	*2*	*2*	*2*	*2*	*2*
Specialty		**EN**	**EN**	**EN**	**EN**	**EN**	**EN**	**EN**	**EN**	**EN**	**EN**	**EN**	**EN**	**GN**	**GN**	**GN**	**GN**	**GN**	**GN**	**GN**	**GO**	**GO**	**GO**	**GO**	**GO**	**GO**

$\mathcal{G} \setminus \mathcal{M}$	$g \setminus m$	*126*	*127*	*128*	*129*	*130*	*131*	*132*	*133*	*134*	*135*	*136*	*137*	*138*	*139*	*140*	*141*	*142*	*143*	*144*	*145*	*146*	*147*	*148*	*149*	*150*
GN-a	*1*																									
GN-b	*2*																									
GN-c	*3*																									
GN-d	*4*																									
GN-e	*5*																									
GN-f	*6*																									
GO-a	*7*		2	2	1	3	3	5																		
GO-b	*8*					1																				
GO-c	*9*				2		1																			
GO-d	*10*																									
GO-e	*11*	2	1																							
GO-f	*12*			1																						
UR-a	*13*								2	1	1					1	1	1								
UR-b	*14*									1		2	1						1	1	1					
UR-c	*15*															1			1			2	1	1		
UR-d	*16*										1		1	2									1		1	1
UR-e	*17*														1	1	2	1	1	2	1		1		2	1
UR-f	*18*														1							1		1		
UR-g	*19*																	1			1			1		1
KA-a	*20*																									
EN-a	*21*																									
EN-b	*22*																									
EN-c	*23*																									
EN-d	*24*																									
EN-e	*25*																									
Duration		530	459	496	534	532	509.5	485	512	496	497	480	481	482	471	483.5	528	474.5	467.5	512	458.5	518	468.5	509	513	459.5
Extended		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	s	*2*	*2*	*2*	*2*	*2*	*2*	*2*	*3*	*3*	*3*	*3*	*3*	*3*	*3*	*3*	*3*	*3*	*3*	*3*	*3*	*3*	*3*	*3*	*3*	*3*
Specialty		**GO**	**GO**	**GO**	**GO**	**GO**	**GO**	**GO**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**

Table F.45: Patterns data. Set of patterns $\mathcal{M}$ with indication of whether they are in the subset of extended $\mathcal{M}^X$ or the non-extended patterns $\mathcal{M}^{NX}$ and which specialty they belong to $\mathcal{M}_s^{\mathcal{S}}$. A_{mg} is the numbers in the matrix indicating how many operations from group g that are included in pattern m.

$\mathcal{G} \setminus \mathcal{M}$	$g \setminus m$	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173
GN-a	1																							
GN-b	2																							
GN-c	3																							
GN-d	4																							
GN-e	5																							
GN-f	6																							
GO-a	7																							
GO-b	8																							
GO-c	9																							
GO-d	10																							
GO-e	11																							
GO-f	12																							
UR-a	13		1	1	1	1																		
UR-b	14						1	1	1	1														
UR-c	15		3	2	1		3	2	1		3	2	2	1	1			5	4	3	2	2	1	1
UR-d	16										1	1		1		1								
UR-e	17												2		3		3		1	1	2	1	2	1
UR-f	18	1																						
UR-g	19	2		1	2	3		1	2	3		1		2		3	1			1	1	2	2	3
KA-a	20																							
EN-a	21																							
EN-b	22																							
EN-c	23																							
EN-d	24																							
EN-e	25																							
Duration		500	530.5	521.5	512.5	503.5	514.5	505.5	496.5	487.5	515.5	506.5	455	497.5	499.5	488.5	490.5	457.5	502	493	537.5	484	528.5	475
Extended		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	s	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Specialty		**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**

$\mathcal{G} \setminus \mathcal{M}$	$g \setminus m$	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196
GN-a	1																							
GN-b	2																							
GN-c	3																							
GN-d	4																							
GN-e	5																							
GN-f	6																							
GO-a	7																							
GO-b	8																							
GO-c	9																							
GO-d	10																							
GO-e	11																							
GO-f	12																							
UR-a	13																							
UR-b	14																							
UR-c	15			5	4	3	2	1																
UR-d	16																							
UR-e	17	2	1																					
UR-f	18																							
UR-g	19	3	4	1	2	3	4	5	6															
KA-a	20																							
EN-a	21											3	2	2	1	1	1					1		
EN-b	22												1		2	1		3	1	1			1	
EN-c	23													1		1	1		1	1	2	3	3	4
EN-d	24									1							1		1					
EN-e	25									1	2									1	1			
Duration		519.5	466	540	531	522	513	504	495	472.5	511	516	508	460	500	452	505	492	497	535.5	487.5	520	512	464
Extended		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	s	3	3	3	3	3	3	3	3	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
Specialty		**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**UR**	**EN**	**EN**	**EN**	**EN**	**EN**	**EN**	**EN**	**EN**	**EN**	**EN**	**EN**	**EN**	**EN**	**EN**	**EN**

Asplin, Corneliussen, Van de Pontseele

Flexible Resource Management for Surgical Scheduling

NTNU

Norwegian University of
Science and Technology