Creating an optimal OR schedule regarding downstream resources.

M. Carlier

Technische Universiteit Delft

Delft University of Technology

Challenge the future

Creating an optimal OR schedule regarding downstream resources.

by

M. Carlier

in partial fulfilment of the requirements for the degree of

Master of Science
in Applied Mathematics

at the Delft University of Technology,
to be defended publicly on Monday March 6, 2017.

Supervisor:	Dr. ir. J. T. van Essen
Thesis committee:	Dr. ir. J. T. van Essen, TU Delft
	Prof. dr. ir. K. I. Aardal, TU Delft
	Dr. ir. M. B. van Gijzen, TU Delft
	Msc. A. J. Schneider, LUMC

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Preface

After high school, I was convinced that I wanted to be an Architect, it turned out I was wrong. After two years of Architecture at the TU Delft, I quit. On a whim, I decided to start studying mathematics. I started my bachelor in mathematics at Leiden University. During my time at Architecture I wanted more challenges, which is exactly what I got during my bachelor in mathematics. After finishing my bachelor at Leiden University, I chose for a master in Applied Mathematics at the TU Delft. A year ago, I started my graduation project at the LUMC, a university medical centre in Leiden.

First, I want to thank everyone at the LUMC who helped me gather information through interviews and meetings. I would especially like to express my gratitude to Carolien Lagers-Dresselhuys and Thomas Schneider. Carolien, despite your busy schedule, you always made time for me and helped me with relating my research to the real issues within the hospital. Thomas, thank you for introducing me to the interesting world of healthcare logistics and giving me the opportunity to do my graduation project in the LUMC.

A special thanks to my supervisor at the TU Delft, Theresia van Essen. You always took the time to give me feedback on my writing and research and never gave up on correcting me on my bad use of commas.

I also want to thank my partner, for keeping me motivated and building me a super fast computer. The computational results would not all have been here if it were not for you. Now the only thing that kept me from getting results were unscheduled Windows updates that interrupted my experiments on multiple occasions. Finally, I would like to thank my family for supporting me during this last year, but also for your support during the rest of my study.

M. Carlier
Delft, March 2017

Abstract

A high percentage of hospital admissions is due to surgical interventions. The operating theatre, which holds the operating rooms (ORs), is therefore one of the key resources in hospitals. Managing the operating theatre and finding an optimal OR schedule is complex due to the many factors that influence it. Scheduling a surgery in an OR influences downstream facilities like the post anaesthesia care unit, intensive care unit and general patient wards. This research was conducted at Leiden University Medical Centre (LUMC), an academic teaching hospital in Leiden, the Netherlands. During the week, the LUMC experiences a large variation in bed occupancy at the patient wards due to the way surgeries are scheduled. The large variation in bed occupancy causes surgeries to be cancelled, because there are no beds available at the ward. Because the OR theatre is such an expensive resource, we want to find a schedule that utilises the OR optimally during opening times. In this research, we develop a clustering method to cluster surgical procedures into surgery groups based on surgery duration and the length of stay. Then, we extend a model that analytically determines the patient distributions over the wards and intensive care for a given OR schedule. We define a mixed integer programming model with the objective to maximise the OR utilisation and minimise the variation in bed occupancy at the wards and intensive care. The model produces an OR schedule with the defined surgery groups assigned to days in the OR. We use two different methods to solve the model: a global approach and a local search heuristic, i.e., simulated annealing. The model has one nonlinear constraint and a complex objective function. Therefore, we linearise the constraint and the objective function, which results in a mixed integer linear program that is proven to be NP-hard. Both approaches are tested on a dataset provided by the LUMC. Furthermore, several scenarios are evaluated. We conclude that the mixed integer linear programming method performs better and faster than the simulated annealing procedure. To obtain an even better solution it is possible to use a combination of both. By using this method, the OR utilisation of the LUMC can improve by 11% and the variation in bed occupancy can be decreased by 80%.

Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Description
CTS	Cardiothoracic surgery
ED	Emergency department
ENT	Ear, nose, and throat surgery
EYE	Ophthalmic surgery
FN	False negative
FP	False positive
GS	General surgery
GYN	Gynaecology
ICU	Intensive care unit
LoS	Length of stay
LUMC	Leiden University Medical Centre
MCU	Medium care unit
MILP	Mixed integer linear programming
MIP	Mixed integer programming
MSS	Master surgery schedule
NS	Neurological surgery
OB	Obstetrics
OMS	Oral and maxillofacial surgery
OR	Operating room
ORT	Orthopaedic surgery
PACU	Post anaesthesia care unit
PLA	Plastic surgery
POS	Pre-operative screening
RAD	Radiology
SA	Simulated annealing
TN	True negative
TP	True positive
URO	Urology

1

Introduction

In this report, we perform research to find an operating room schedule that improves the operating room utilisation while taking the required number of beds on the intensive care and the general patient wards into account. We develop a tactical operating room schedule for the Leiden University Medical Centre (LUMC). Our scheduling approach is based on the research of Van Oostrum *et al.* [2]. Contrary to [2], we use an analytical model to determine the required number of beds on the intensive care and wards. The analytical model is based on the research of Fügener [3]. An important difference between the model of [3] and our model is that we schedule multiple types of surgeries per day instead of assigning a whole day to a surgical specialty.
Section 1.1 gives background information about the LUMC, the hospital where this research was conducted. The motivation for this research is described in Section 1.2. In Section 1.3, a problem description is given. Section 1.4 states the research goals that follow from the problem description. The scope of the report is described in Section 1.5. Section 1.6 contains the research questions and outline of the report.

1.1. Context

This research was conducted at the Leiden University Medical Centre, an academic teaching hospital in Leiden, the Netherlands. After years of cooperation between university and academic hospital in Leiden, the LUMC was founded in 1996. It is now a modern university medical centre for research, education, and patient care. The LUMC strives to offer the highest quality in medical care as well as in attention and care for its patients. Yearly, over 16.000 patients undergo surgery at the LUMC. The hospital has an operating room complex that consists of 20 operating rooms, where over 12 different specialties perform surgeries. The LUMC has multiple patient wards, i.e., a day treatment patient ward, a short stay patient ward and multiple long stay patient wards. Together these wards house over 400 beds.
The starting point of this research was a request from the LUMC to develop an optimisation model that would find a schedule for the operating room department that would minimise the variation of the bed occupancy on the patient wards.

1.2. Research motivation

The demand for health care services is increasing in industrialised countries due to an ageing society and technological progress [4]. However, hospital resources are under pressure caused by human resource shortages. On the one hand, hospitals want to increase their production, while on the other hand, they do not want to increase the costs. Furthermore, they want to maximise patient satisfaction. Therefore, operations management is becoming increasingly important in supplying health care in hospitals.

A high percentage of hospital admissions is due to surgical interventions. The operating theatre, which consists of a pre-operative holding unit, operating rooms (ORs), recovery beds and post anaesthesia care unit (PACU), is therefore one of the key resources in hospitals. For most hospitals, the

ORs are the most expensive resources, but also generate most revenue [5]. Therefore, the operating theatre has great influence on how hospitals function in general. However, managing the operating theatre and finding an optimal OR schedule is complex due to the many factors that influence it. When generating an OR schedule, we need to consider all the facilities in the operating theatre, but also staff (surgeons, nurses, anaesthetists, etc.), equipment and capacity of downstream resources and facilities. Scheduling a surgery in an OR influences downstream facilities like the PACU, the intensive care unit (ICU) and the general patient wards. Currently, while planning and scheduling the ORs, these downstream facilities are not always considered even though the capacity of these facilities can be limiting on scheduling surgeries. If there is no bed available on the ICU or general patient ward on the day of a patients surgery, the surgery can be cancelled. Only considering the availability of ORs when scheduling can lead to cancellation of surgeries, which leads to dissatisfied patients.

1.3. Problem description

In the current planning of the OR department in the LUMC, there is a regular shortage of capacity being perceived. One of the causes is the unnatural (self-inflicted) variation in bed occupancy on the general patient wards. In this research, we make the distinction between elective and emergency patients. Patients who do not need urgent surgery, but together with their physician choose to have surgery because it is beneficial for them, are called elective patients. Surgeries of elective patients can be scheduled in advance. Emergency patients are in need of urgent surgery and are therefore not scheduled in advance.

The LUMC uses a block-booking system when scheduling surgeries. Each surgical specialty is assigned to blocks (usually a day) in certain ORs. The schedule, which contains all these blocks, is called the master surgery schedule (MSS). The MSS is a cyclic schedule and the MSS of the LUMC has a cycle length of two weeks. The surgical specialties can schedule elective surgeries in the assigned blocks. Many different factors are important when creating an OR schedule, i.e., available surgeons, surgery duration, length of stay (LoS) on the wards and available equipment in ORs. Creating the OR schedule is challenging, because of these many factors, but also because stakeholders may have different priorities. The ORs are an expensive resource and should therefore be used optimally during their opening hours. However, planners do not base surgery durations on data, but on the experience of surgeons. Often, the available OR time is either not completely used or surgeries go on after the opening hours.

Currently, the OR schedule is made without considering consequences on downstream resources, like the bed occupancy on the patient wards. Therefore, scheduling decisions in the OR can cause problems in the downstream resources. In 2015, the number of cancelled elective surgical patients in the LUMC was 1656, of which 312 were cancelled within the 24 hours before their surgery. These numbers do not contain any patient related reasons, only cancellations due to capacity shortages or other planning related reasons. Most frequently, these cancellations came from prioritising emergency patients. However, schedulers in the LUMC notice that many of the cancellations occur because there are no beds available on either the ICU or general patient wards, due to the large variation in bed occupancy on the ICU and wards. Cancelling patients also means rescheduling these patients. This is a nuisance for the patient as well as inefficient for the hospital. Rescheduling patients gives a lot of extra work for hospital staff, in the LUMC this came down to 0.3 fte last year. Patients are often informed of the cancellation on a short notice. The LUMC responded by increasing the capacity of beds, planning less patients and using a fixed quota of admissions a day. However, they would like to start scheduling smarter so that the variation in bed occupancy is reduced.

To tackle the variation in bed occupancy and maximise the utilisation of the OR a new tactical plan for the OR schedule is required. Therefore, this research aims to find an OR schedule that minimises the variation in bed occupancy and maximises the OR utilisation. Other objectives that can be considered for this problem are minimising the number of required beds or minimising costs.

1.4. Research Goals

Stakeholders from the hospital have set the following research goals:

- *Minimise the variation in bed occupancy on the wards during the week.*
- *Minimise the variation in bed occupancy on the ICU during the week.*
- *Maximise the OR utilisation.*
- *Minimise overtime in the ORs.*

We want to optimally use the ORs during the time that they are open. However, this should not effect the downstream resources in a negative way. Bottlenecks are bed capacities in certain long stay wards and the ICU. These capacity problems lead to cancellation of patients, which we try to avoid. Therefore, the new OR schedule should minimise the variation in bed occupancy on these wards. Furthermore, we also want to avoid overtime, i.e., ORs that stay open after their opening hours. Overtime in the ORs is very expensive for the hospital. Therefore, the OR department tries to minimise the overtime. However, if multiple surgeries in the same OR take shorter than expected, you can end up with unused OR time at the end of the day. The staff, instruments and OR are available, but are not being used. This is also something the OR department wants to avoid as much as possible. This means we want to maximise the use of OR capacity within opening hours, but minimise overtime.

1.5. Scope

In operations research, we distinguish three different levels of planning, i.e., strategic, tactical and operational. Strategic plans are designed to determine the future and long term goals of the entire operation. This is the highest level of planning and serves as a framework for the lower levels. Tactical plans support the strategic plans by outlining what various parts of the organisation should do. Operational plans are day-to-day plans that focus on specific procedures or processes. We position this research in the health care planning and control framework of Hans *et al.* [1] at the tactical level of resource capacity planning, see Figure 1.1. We cluster surgical procedures based on factors like surgery duration, LoS and specialty. These clusters of surgical procedures are then be planned in a cyclic way to minimise the variation in bed occupancy. We use a model to analytically determine the required number of beds for a given schedule. This research focuses on elective patient flows through the ORs, ICU and surgical wards of the LUMC. The flow of non-surgical patients are out of the scope of this project. This research provides the hospital with a scheduling strategy to tackle the variation in bed occupancy and improve the OR utilisation.

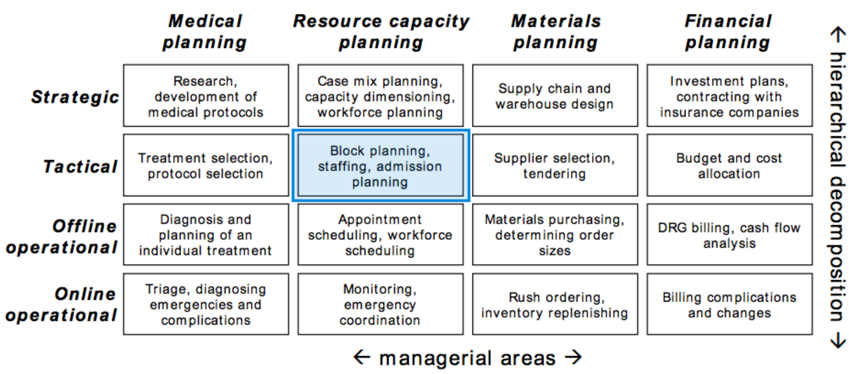

Figure 1.1: Healthcare planning & control framework as described in Hans *et al.* [1].

1.6. Research questions

We have formulated research questions to structure the report. Each research question is associated with a chapter. This is the outline of this report.

What is the current situation and performance? (Chapter 2)
This chapter is based on the experiences of practitioners working in the considered hospital obtained through interviews with surgical and supporting staff. It discusses the clinical course of the patient, the planning process and important resources of the hospital.

What models and solution methods are available in literature? (Chapter 3)
A literature study is carried out to get an overview of OR scheduling and planning. It discusses three main problems on different levels of planning by considering the mathematical models, solution methods and objectives used to optimise the OR schedule.

How should the OR schedule be optimised? (Chapter 4)
This chapter describes the mathematical model to optimise the OR schedule in relation to the bed occupancy on the wards and ICU. Surgical procedures are clustered into surgery groups based on LoS and surgery duration. These surgery groups are then scheduled into the tactical OR schedule. Multiple restrictions and a model to analytically determine the resulting bed occupancy are presented.

What solution methods are used? (Chapter 5 and 6)
We use two methods to find a solution to our model.

1. A global approach where we linearise the nonlinear constraints and objective function and then solve the corresponding mixed integer linear program. (Chapter 5)
2. A local search heuristic, namely simulated annealing. (Chapter 6)

Which method performs better? (Chapter 7)
We determine the best parameter settings for both methods and compare the results of the global approach and the local search heuristic. We also show results for a combination of the two solution methods.

What influence do different scenarios have on the solution? (Chapter 8)
Different scenarios are given and the solutions are compared to the default situation.

- Closing of the short stay ward during the weekend.
- Increasing the importance of certain wards.
- Minimising the bed occupancy.
- Planning more groups with high variance in surgery duration.
- Using different changeover times.
- Without using the master surgery schedule.
- Using data from another hospital.

What are the conclusions and recommendations? (Chapter 9)
In this chapter, conclusions regarding the solution methods and their outcomes are given. Furthermore, recommendations to the management are made and recommendations for further research are given.

2

Situational analysis

In this chapter, an analysis is given of processes at the LUMC that are related to the problems formulated in Chapter 1. The patient flow through the hospital and the OR department are described in Section 2.1. A brief overview of the current planning process is given in Section 2.2. Prior research regarding patient logistics, which was carried out at the LUMC, is described in Section 2.3. This chapter concludes in Section 2.4 by giving performance indicators regarding the OR and the wards.

2.1. Processes

This section describes the flow of surgical patients through the hospital to understand the processes surrounding the problem. Further, a simplified version of the current planning process is given to understand where problems may occur.

2.1.1. Clinical route of the patient

We consider the typical flow of an elective surgical patient who stays longer than a day, see Figure 2.1. A patients route through the hospital generally starts in the outpatient clinic. Outpatient clinics provide diagnoses and care for patients that do not need to stay overnight. If the patient is in need of having surgery, they are put on the waiting list and have a pre-operative screening. On the day of their surgery they are admitted to the ward of their surgical specialty. Within an hour of their surgery, the patient is transferred to the OR complex. Here, they wait in the pre-operative holding area before they are brought to the operating room where they undergo their surgery. After their surgery, patients are brought to the recovery area, PACU, ICU or medium care unit (MCU), depending on their medical condition. When the patients' medical condition is sufficient, they are transferred to their ward. From the ward, they can be discharged to go home, another hospital or a nursing home. Another possibility is that the patient died during their stay in the hospital.

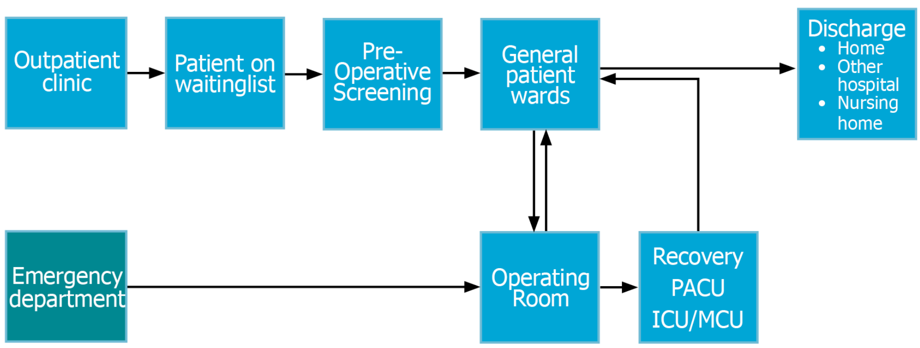

Figure 2.1: Surgical patient process.

We also see the typical flow of emergency surgical patients in Figure 2.1. Usually, they will directly go from the emergency department to the OR complex. From there, they follow the same route as elective patients. During the time in the OR, the patients goes through several steps that are described in Figure 2.2. First, the patient is transferred from the pre-operative holding area to the OR. The moment a patient enters the OR, the *OR in time* is registered. At that time, the surgeon is not yet in the OR. First, if necessary, the anaesthesiologist puts the patient under surgical anaesthesia. This is called *induction*. Then, the patient is prepared for surgery, equipment is connected and instruments are prepared. When the patient is ready to be operated on, the surgeon is called and the surgery starts. This time is registered as the start of the *intervention time*. Once the surgeon is finished and the wound is closed, the end of the *intervention time* is registered. This is the moment the surgeon leaves the OR. Then, equipment is disconnected and the patient is brought out of anaesthesia. The moment the patient leaves the OR and is brought to the recovery area is registered as the *OR out time*. The *OR out time* minus the *OR in time* is called the *session time*, which includes the induction, intervention time and end of the surgery. In this report, the surgery duration is defined as the session time, so the gross OR time of a surgery.

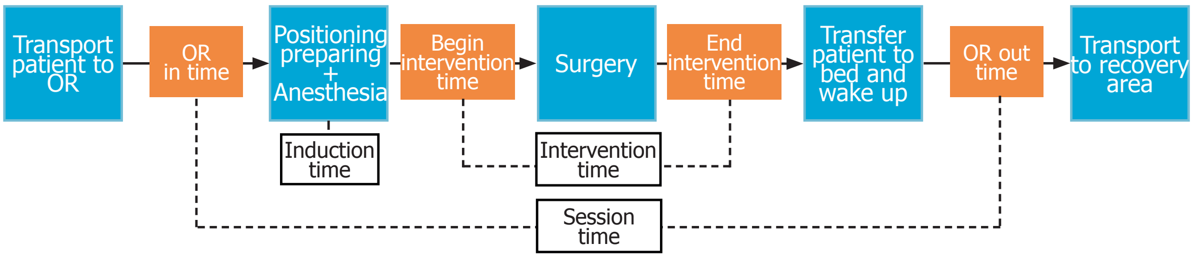

Figure 2.2: Operating room process.

2.1.2. Important departments and areas

In this section, we describe various important departments and areas in the hospital. Table 2.1 explains the abbreviations of surgical specialties used in this section.

Abbreviation	Specialty
CTS	Cardiothoracic surgery
ENT	Ear, nose, and throat surgery
EYE	Ophthalmic surgery
GS	General surgery
GYN	Gynaecology
NS	Neurological surgery
OB	Obstetrics
OMS	Oral and maxillofacial surgery
ORT	Orthopaedic surgery
PLA	Plastic surgery
RAD	Radiology
URO	Urology

Table 2.1: Abbreviations of the surgical specialties in the LUMC.

Outpatient clinic In the outpatient clinic, patients are diagnosed, treated or undergo a small procedure. Because the patients do not stay overnight, they are called outpatients. After their visit at the outpatient clinic, patients are often scheduled to be admitted as inpatients. In case the patient needs surgery, he will be sent to get a pre-operative screening and is placed on the waiting list. Most patients enter the hospital through the outpatient clinic.
Pre-operative Screening (POS) Before surgery, patients have to visit the POS. Here, a health check is done to make sure the patient's condition is sufficient to undergo surgery. The patient also receives

information regarding his surgery, e.g., medicine use and if he has to be sober or not.
Emergency department (ED) The emergency department offers acute care. Patients with a wide range of medical issues arrive unscheduled. The urgency varies between patients. The hospital is able to forecast the demand on the resources of the patients entering the hospital via the emergency department. Emergency surgery patients go through an adapted pre-operative process.
Operating room (OR) In the operating rooms, surgeries take place. The master surgery schedule of the LUMC from 2016 is given in Figure 2.3. There are 20 ORs in the LUMC, that are assigned to a specialty each day of the week. This schedule repeats itself every two weeks. Every week, two of the cardiothoracic ORs, two of the neurosurgery ORs and one OR of the specialty ear, nose, throat have extended opening hours. Once every two weeks, the specialty plastic surgery (PLA) has extended opening hours for one of their ORs. On Thursdays, OR 10 starts later. The other ORs have regular opening hours from 8.00-16.00. The LUMC works with dedicated emergency ORs. In these ORs, nothing is scheduled, so when an emergency patient needs to undergo surgery most of the time there is an OR available. In theory, the ORs are designed in a way that every surgical procedure could take place in every OR. The equipment can be moved, but this is time consuming and in general not very productive. Therefore, the preferences of the specialties are taken into account while scheduling. In Table 2.2, we see which specialties can operate in which OR.

EVEN WEEKS

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Monday	ORT	ORT	URO		GS/ Em.OR	GS	GS	GS	GYN	PLA/ OMS	EYE	Em.OR	NS	NS	ENT	ENT	CTS	CTS	CTS	CTS
Tuesday	ORT	ORT	URO		GS/ Em.OR	GS	GYN	GS	PLA	OMS	EYE	Em.OR	NS		ENT local	ENT	CTS	CTS	CTS	CTS
Wednesday	ORT	ORT	URO	OB/ Em.OR		GS	GS	GS	GYN	PLA/ GS	EYE local	Em.OR	NS	NS	ENT	ENT	CTS	CTS	CTS	CTS
Thursday	ORT	ORT	URO		GS/ Em.OR	GS	GS	GS	GYN	OMS	EYE	Em.OR	NS	NS	ENT	ENT	CTS	CTS	CTS	CTS
Friday		ORT/ Em.OR	URO/ RAD		GS	GS	GS	GS	GYN	PLA/ Em.OR	EYE	Em.OR	NS	NS	ENT	ENT	CTS	CTS	CTS	CTS

ODD WEEKS

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Monday	ORT	ORT	URO		GS/ Em.OR	GS	GS	GS	GYN	PLA/ OMS	EYE local	Em.OR	NS	NS	ENT	ENT	CTS	CTS	CTS	CTS
Tuesday	ORT	GS	URO		GS/ Em.OR	GS	GYN	GS	GYN	OMS	EYE	Em.OR	NS	NS		ENT	CTS	CTS	CTS	CTS
Wednesday	ORT	ORT	URO	OB/ Em.OR		GS	GS	GS	GYN	PLA/ GS	EYE	Em.OR		NS	ENT	ENT	CTS	CTS	CTS	CTS
Thursday	ORT	ORT	URO		GS/ Em.OR	GS	GS	GS	GYN	OMS	EYE	Em.OR	NS	NS	ENT	ENT	CTS	CTS	CTS	CTS
Friday		ORT/ Em.OR	URO/ RAD		GS	GS	GS	GS	GYN	PLA/ Em.OR	EYE local	Em.OR	NS	NS	ENT	ENT		CTS	CTS	CTS

Local (No anaesthesia)
Extended opening hours until 19.00 hour
Extended opening hours until 21.00 hour
Extended opening hours until 23.00 hour
Starts at 11.00 hour

Figure 2.3: Master surgery schedule 2016 of the LUMC.

OR	Specialty	OR	Specialty
1	ORT	**11**	EYE
2	GS, ORT, Emergency OR (Em. OR)	**12**	Em.OR
3	URO	**13**	NS
4	OB, Em. OR	**14**	NS
5	GS, Em. OR	**15**	ENT
6	GS	**16**	ENT
7	GS, GYN	**17**	CTS
8	GS	**18**	CTS
9	GYN, PLA	**19**	CTS
10	OMS, PLA, GS, Em. OR	**20**	CTS

Table 2.2: ORs with the specialties that operate in them.

Recovery & PACU After their surgery, patients are brought to either the recovery or post anaesthetic care unit. Here, they are monitored, while they recover from surgery. This can take half an hour to multiple hours. Once it is clear that the patient is recovering well from the anaesthetics, he is transferred back to his ward. In the recovery there are nurses, but no physicians. Patients who need to be monitored more closely are placed on the PACU. Therefore, there is always one physician present on the PACU.

ICU and MCU The intensive and medium care units are for patients that need urgent care, because of there medical condition. On the ICU and MCU, patients are monitored more closely by more medical personnel than on regular patient wards. Patients are brought to these facilities straight after their surgery.

General patient wards A ward is a facility of a medical specialty in the hospital where patients get specific care. In the LUMC, there are three types of wards: day clinic, short stay (less than five days) and long stay (more than 5 days). However, patients who need specific medical care are placed on the long stay ward even though their stay might be shorter than 5 days. Emergency patients are also placed on the long stay wards. All wards have a certain capacity of available beds. This depends on both the actual beds and on the available staff. We make the distinction between *opened* and *closed* beds. Opened beds are beds that are staffed and closed beds are physically there, but not staffed. In this research, we refer to opened beds as beds. When we mention a shortage of beds, we actually mean a shortage of staffed beds. We consider multiple wards in this research that can be divided into three categories: day treatment, short stay ward and multiple long stay wards. They have the following opening times:

- day treatment: Monday to Friday: 7.00-20.00;
- short stay: Monday to Friday, Saturday until 12.00;
- long stay: every day of the week.

2.2. Planning process

In Figure 2.4, an abstraction of the planning process is shown. The master surgery schedule of the LUMC states which specialties are assigned to certain days in the ORs, i.e., OR-days. Around two weeks before an OR date, a patient is planned in the concept planning. This concept planning is made without regarding any resources other than the OR capacity. In the current situation, there are separate planners for the OR schedule and the general patient wards. Each specialty has someone to make a schedule for the assigned OR-days. This can be a planner, but also a physician, physician's assistant or a nurse. They schedule with intervention time, based on intuition or experience. Resource conflicts are often avoided by experience, but the way to handle restrictions is not formalised. Regular ORs are open for 480 minutes. Of that time, 360 minutes of intervention time is scheduled to accommodate for, e.g., induction, preparation and clean up. This can lead to constant over- or underestimation of surgery durations.

Furthermore, the planners do not have a good overview of the bed occupancy on the wards in this phase of planning. The planners do take several scheduling restrictions into account.

Availability of instruments The amount of instruments determines how many surgeries of the same type can be planned on the same day. The following restrictions influence the planning.

Surgeons capability Surgeons are assigned to certain days in a certain OR. The surgeon scheduled to a certain OR-day needs to be able to perform all surgeries that are scheduled on that day. The LUMC is an academic hospital. Therefore, surgeons have specific skills and preferences towards certain type of surgeries. In general, surgeons cannot perform all surgery types within their specialty. Therefore, surgeons usually only perform a subset of all the surgery types in their specialty. However, they do not work the same shifts every week. This makes it extra difficult for planners.

Capacity of the holding and recovery areas The holding and recovery areas need to house the patients that are scheduled in the OR. If many surgeries end at the same time it may cause a shortage of capacity on the recovery unit.

However, due to limitations in the available data and preferences of stakeholders, these restrictions will not be taken into account.

Once surgeons have agreed on the concept planning and patients agree with their OR date, the week programme is frozen and no further changes are allowed. However, only after this moment bed capacity on the general patient wards and the PACU is considered. Bed availability on the ICU is only known at the day of the surgery, which leads to last minute cancellations.

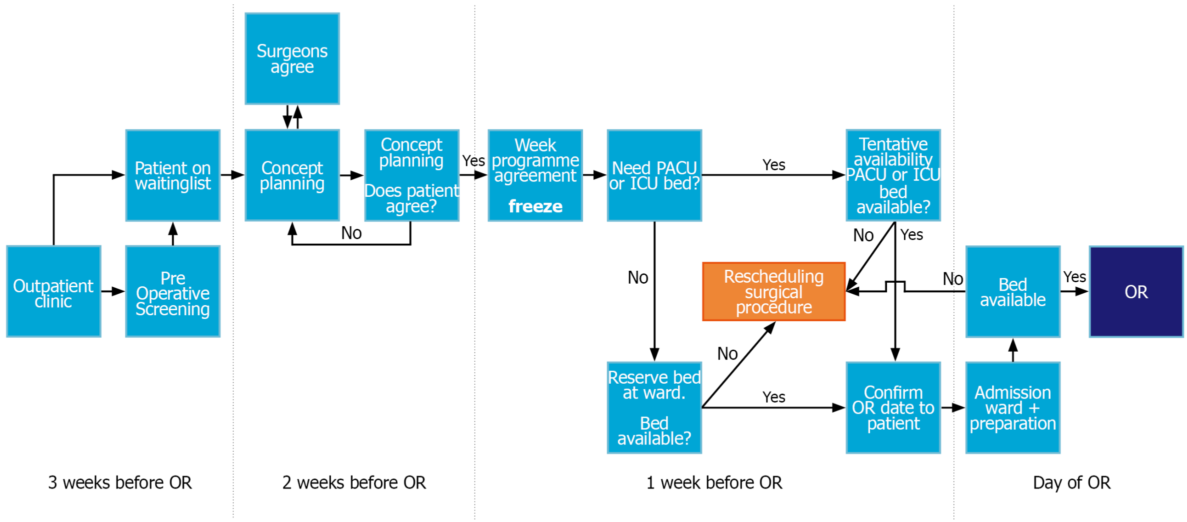

Figure 2.4: Planning process of elective surgical patients.

2.3. Previous research in the LUMC

The LUMC is running multiple projects regarding patient logistics. They include the problems surrounding the bed occupancy on the general patient wards. When the problem was observed, the planners started using a fixed quota of admissions per day. This helped to stabilise the patient inflow. However, it did not control the patient outflow. More beds and staff were placed on the wards with the most capacity problems. This was not a permanent solution to one of the root causes, the OR schedule. Therefore, the LUMC conducted research regarding the MSS. An optimisation model was applied to find an optimal MSS regarding the bed occupancy on the wards. It also regarded the effects of having blocks of half OR-days instead of only blocks of whole OR-days. However, this still lead to problems. For example, a specialty that had an OR-day on Monday and Wednesday performed on average two surgeries on Monday and six on Wednesday, while the model only took the mean number of surgeries into account on each day. Thus, there would still be a peak of patients needing a bed on Wednesdays. Therefore, the outcome of this model was not put into practice.

In the search for a new MSS, only the bed occupancy on the wards was taken into account. However, the bed occupancy on the ICU and MCU are also limiting factors. In this project, we create a schedule that specifies how many and what type of surgical procedures should be scheduled on what day such that the bed occupancy on the wards and ICU stabilise during the week.

2.4. Performance indicators

In this section, we describe the indicators that we use to evaluate the performance of the system. In Section 2.4.1, we discuss the OR utilisation and the problems of under- and overtime. The indicators regarding the bed demand at the wards, ICUs and MCUs are described in Section 2.4.2.

2.4.1. OR utilisation and overtime

ORs with regular opening hours are open from 8.00-16.00 on Monday to Friday for elective patients. Exceptions are the extended opening hours for the cardiothoracic ORs, neurological surgery ORs, plastic surgery OR and the OR of the specialty ear, nose and throat, see Figure 2.3. We are developing a tactical surgery scheduling approach. Therefore, we assume that the availability of OR capacity is known beforehand. The OR department perceive that they are experiencing undertime, i.e., times

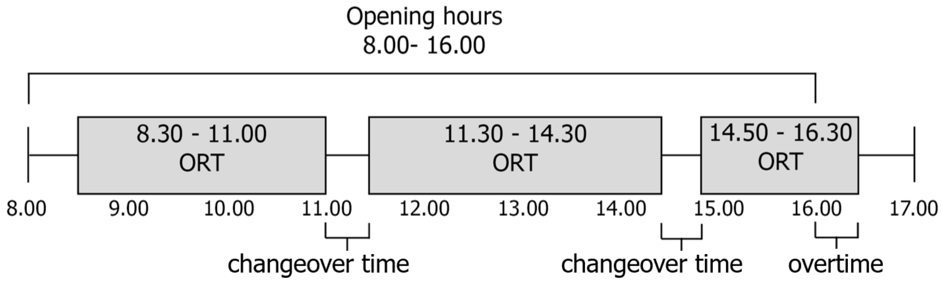

Figure 2.5: Day in OR with overtime and changeover times.

when no procedures are done during opening hours. These times are usually at the start or at the end of the day. The planners schedule with mean intervention times. Because surgery durations are not deterministic this can cause problems. Often surgeries take shorter or longer than expected. If multiple surgeries scheduled in the same OR take longer than expected, either the last planned patient will be cancelled or the OR has to stay open after opening hours. Overtime and undertime are expensive for hospitals. Therefore, we are interested in the utilisation and overtime of the ORs. The intervention time does not accommodate for, e.g., induction and preparation. Furthermore, time is needed between surgeries to clean the OR and bring in the next patient. The time between a patient leaving the OR and the next patient entering the OR is called the changeover time, see Figure 2.5. Currently, every OR-day gets a fixed amount of time for changeover, induction, preparation and clean up. However, the time needed for this can be very different for differently scheduled OR-days.

We define the OR utilisation as:

$$\text{OR utilisation} = \frac{\sum \text{(OR out - OR in) during opening hours}}{\text{Total available OR time}} \cdot 100\%.$$

For the OR utilisation, we only consider session times, i.e., the time from a patient entering the OR until the patient leaving the OR, during the opening hours of the OR. In Figure 2.5, we see an example of an OR-day. During opening hours the sum of session times is equal to 400 minutes, the last 30 minutes are not taken into account because they happen outside the opening hours of the OR. The OR utilisation for this OR-day is given by $400/480 \cdot 100\% = 83.3\%$.

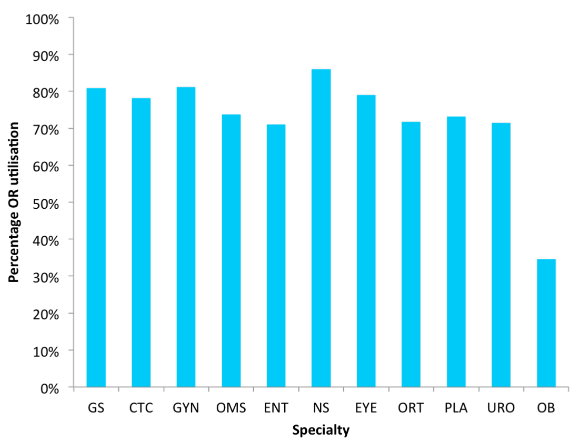

Figure 2.6: Percentage OR utilisation per specialty.

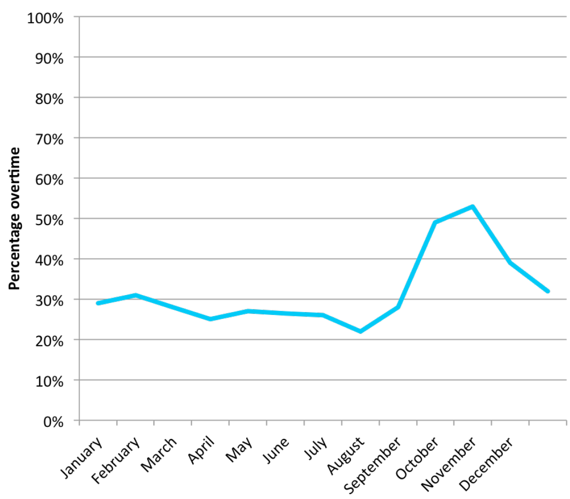

Figure 2.7: Percentage of OR utilisation per month.

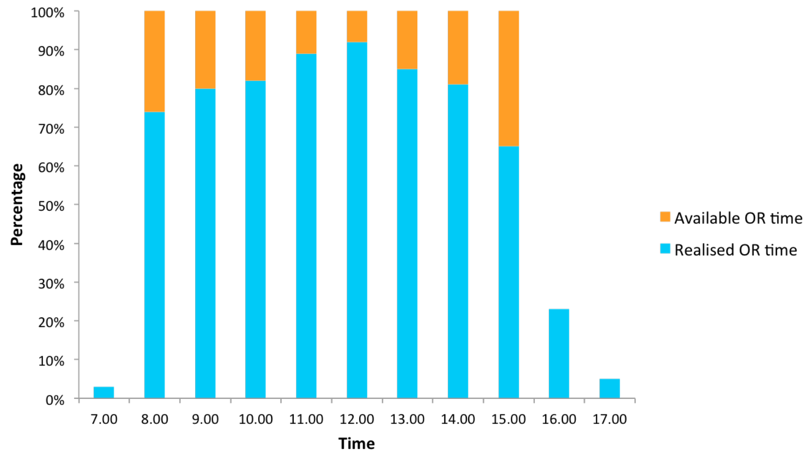

Figure 2.8: Average OR utilisation per hour of the specialty ENT versus the scheduled OR utilisation.

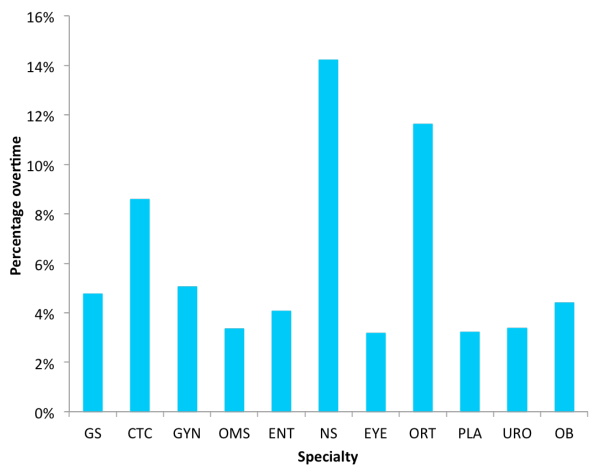

Figure 2.9: Percentage of overtime per specialty. Number of hours overtime devided by the total scheduled hours.

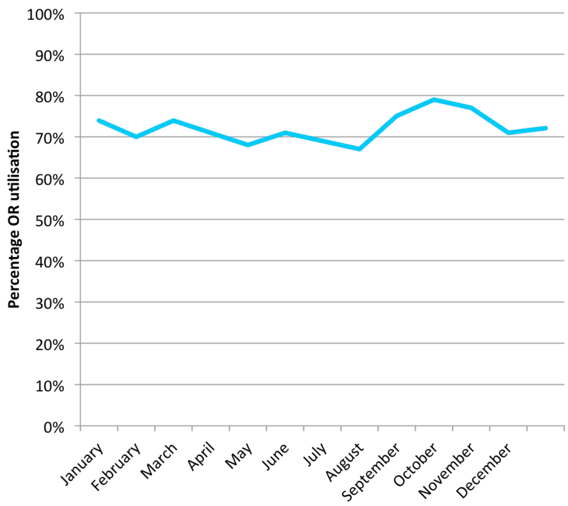

Figure 2.10: Percentage of days with overtime per month.

OR utilisation In Figure 2.6, we see the average OR utilisation per specialty. We only take the OR-days into account that are on weekdays. This percentage is based on the sum of session times during the opening hours of the OR divided by the total available hours. Holidays, closed ORs and extended opening times are taken into account. These figures are based on data from 2015. The ORs have an average OR utilisation of 73%. Between surgeries, there is a certain changeover time, when the ORs are cleaned and a new patient is brought in. However, this should only take 15 minutes, according to OR personnel. Planning for a 100% OR utilisation leads to overtime due to variation in surgery durations and the need for changeover time. From interviews with management we set the preferred OR utilisation at 85%.
In Figure 2.7, we see the overall OR utilisation per month. In July and August, the LUMC works with a reduced OR schedule due to the summer holiday. This is taken into account when calculating the OR utilisation. However, the OR utilisation is still slightly lower during these months. At the end of the

year, the OR utilisation increases slightly.
In Figure 2.8, we see the average OR utilisation per hour in the OR for the specialty ENT. Only 70% of the surgeries start on time. In the LUMC, one anaesthesiologist is responsible for two ORs. However, all ORs open at the same time. This may be one of the reasons why the ORs do not all start on time. Other possible reasons were concluded from interviews with staff. Surgical staff has daily morning meetings that can cause personnel to be later at the OR. Furthermore, the start of the OR-day is not clearly defined, sometimes it is denoted as the moment a patient enters the OR other times as the moment a surgeon should start the surgery.
In Figure 2.8, we also see a lower OR utilisation at the end of the day. This is caused by inaccurate estimation of intervention times and change over times instead of using quantitative criteria when scheduling. To minimise overtime, the OR management has set the rule that no new surgeries can start after 14.00 hour, when they do not expect to be done before 16.00 hour. This can also lead to undertime.
Overtime in OR In Figure 2.9, we see the percentage of OR-days where overtime occurred per specialty. In Figure 2.10, we see that the percentage of overtime at the end of the year increases. Production goals for each specialty are determined at the beginning of the year. To reach these goals before the end of the year, more surgeries are planned during the last months of the year, which leads to more overtime. The average overtime when not regarding the emergency ORs is 32%. Overtime is very expensive for the hospital and should therefore be minimised.

2.4.2. Bed demand

We discuss two indicators concerning the wards: the required number of beds on the wards and ICU/MCU and the bed demand on the short stay ward during the weekend.

Required numbers of beds In Figure 2.11 and Figure 2.12, we see the bed occupancy of elective patients for two random weeks in the year 2015 on the wards VCH1 and VBS3 respectively. There is a lot of variability in the bed occupancy. During the weekend, there are no elective surgeries, so we see lower bed occupancies. With the current planning process, it is difficult to estimate the number of required beds. The LoS of patients can vary and is sometimes difficult to predict. The variation in bed occupancy on the wards causes patients to be cancelled for surgery, because there is no bed available on the ward. With the beds on the ICU, it is even more difficult. Patients that need a bed on the ICU after their surgery need to wait until the day of their surgery to know if there is a bed available.

In Figure 2.13 and 2.14, we see the bed occupancy on the ward VCH1, for the beginning of the year 2015 and the end of the year 2015. As with the OR utilisation and overtime, we see that at the end of the year the bed occupancy is higher. There is also a clear difference between the days of the week. Near the end of the week the bed occupancy goes up.

The goal of this research is to minimise the variation in bed occupancy during the week. This should lead to less cancellations of surgeries and a more balanced workload for staff at the wards. We determine the mean variation in bed occupancy during the week for each ward, based on data from 2015. Since no surgeries are performed on Saturdays and Sundays, we expect the bed occupancy to be lower during the weekend. The main problem with the variation in bed occupancy comes from cancelling elective patients due to a shortage of beds on the wards. Therefore, we do not take Saturdays and Sundays into account when calculating the variation in bed occupancy. We define the variation in bed occupancy in one week as the difference between the maximum and minimum bed occupancy during that week. In Table 2.3, we see the mean variation in bed occupancy during the week. This is calculated by determining the variation in bed occupancy for each week of the year and then calculating the mean for each ward. The sum over the variation in bed occupancy over all wards is 53 beds. The aim of this research is to find an OR schedule that reduces this variation.

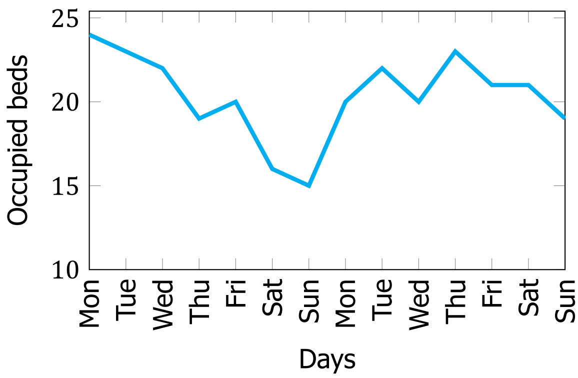

Figure 2.11: Bed occupancy on the ward VCH1 from two weeks in March in 2015.

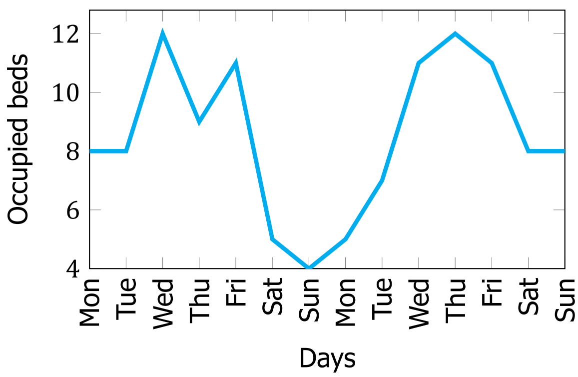

Figure 2.12: Bed occupancy on the ward VBS3 from two weeks in September in 2015.

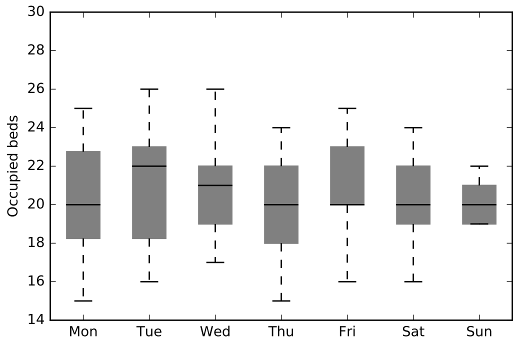

Figure 2.13: Bed occupancy for each day of the week from January to July in 2015 on ward VCH1.

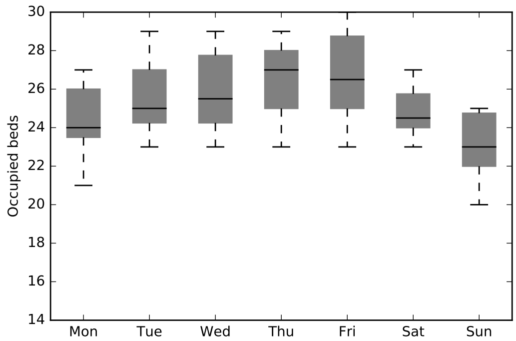

Figure 2.14: Bed occupancy for each day of the week from September to December on ward VCH1.

Ward	Mean variation in bed occupancy	Minimum variation in bed occupancy	Maximum variation in bed occupancy
DBVL	9	4	18
KVVL	13	3	24
VBS3	5	1	10
VCH1	4	1	8
VCH2	4	1	13
VCH3	5	1	12
VERL	2	0	7
VIG1	1	0	3
VPL2	2	0	4
VRVK	4	1	10
ICU	4	1	8

Table 2.3: Mean variation in bed occupancy during the week.

Short stay ward during the weekend The short stay ward of the LUMC houses patients that have an expected LoS of less than five days. It closes every weekend, because of personnel costs. However, patients can have a longer LoS than expected. Patients that cannot be discharged before Saturday need to be transferred to one of the long stay wards. If they expect that a patient is not going to be discharged on time, a bed is reserved on the long stay ward on Friday. Sometimes the reserved bed is not even necessary, because the patient is discharged in time. Stakeholders in the LUMC are interested in the benefits of keeping the short stay ward open during weekends versus the costs. However, this is out of the scope of this research. Therefore, transferring patients from the short stay ward during weekends should be avoided as much as possible.

3
Literature review

This literature review focuses on OR scheduling and planning. OR scheduling can be divided into three problems, namely the case mix problem, the master surgery scheduling problem and the surgery scheduling problem. In this literature review, we consider the MSS problem, which is located at the tactical level of planning. Section 3.1 gives a brief overview of what has been studied in OR scheduling and planning with consideration of downstream facilities and resources, while using block scheduling. Section 3.2 focuses on surgical case scheduling and section 3.3 covers the assigning and sequencing of surgical procedures within the allocated time blocks. Section 3.4 describes the construction of surgery types.

Over the years, much research has been conducted on OR planning and scheduling. Multiple review papers have been written on this topic. Cardoen *et al.* [6] and Guerriero and Guido [5] both take a different approach to provide an overview of related papers. Cardoen *et al.* [6] propose a review that is structured using descriptive fields. The fields are either related to the problem setting (e.g., patient characteristics or performance measures) or technical features (e.g., research methodology or uncertainty incorporation). They conclude that researchers should put effort into using stochastic activity durations instead of focusing on deterministic approaches. Furthermore, a better integration of OR downstream and upstream facilities and resources should be favoured. However, both those suggestions would lead to an increase of the computational complexity. Guerriero and Guido [5] give a more general survey, where they aim to provide an overview of how operational research can be applied to OR scheduling and planning. They pay particular attention to the most interesting mathematical models and solution approaches. Their classification is based on a more traditionally used approach where a distinction is made of three different hierarchical decision levels, i.e., strategic (long term), tactical (medium term), and operational (short term). Cardoen *et al.* [6] chose to use a different type of classification because of the ambiguity of these hierarchical decision levels and their lack of detail. Both papers conclude that an efficient OR schedule is obtained when not only the demand of OR related resources are considered, but also other resources, such as the capacity of the ICU and other wards.

This research focuses on minimising the variation of the bed occupancy on the wards when creating an OR schedule. Therefore, this literature review does not take articles into account that consider OR planning and scheduling without considering downstream facilities and resources. We conduct our literature search using search engines PubMed, Web of Science, Embase and Scopus. With a query based on Van Oostrum *et al.* [2], Vanberkel *et al.* [7] and Fügener *et al.* [8] we found 294 unique articles. By first selecting on title and then on abstract we found 27 relevant articles. We divided this literature review into four subjects: developing an MSS, surgical case planning, assigning surgeries to blocks and sequencing these surgeries and classification of surgeries.

3.1. Developing master surgery schedules

Search terms: operating room planning, operating room scheduling, bed occupancy, peak bed occupancy, master surgery schedule, block scheduling.

Operations research is extensively applied in health care and received considerable attention in literature. The operating room is a very expensive resource that most patients, who are in hospital, use. Managing the operating room is a difficult task because of conflicting preferences of stakeholders, scarcity of expensive resources and conflicting priorities [6]. Therefore, a large amount of literature on operating room scheduling is available. In this section we focus on articles that develop MSSs, while considering downstream resources. The found articles are divided by the type of optimisation model and main solution approach they use.

The term MSS is used in different ways. Beliën and Demeulemeester [9] see it as a schedule that is specialty specific, meaning OR time is assigned to a surgical specialty. The surgical specialties then determine the patients that they schedule in their OR blocks. However, Van Oostrum *et al.* [2] uses the term MSS for a more specific schedule where OR blocks are being allotted to specific surgical procedures. In [10], the term MSS is not used and instead they use the term surgical block scheduling. In this research we use the term MSS as in [9], assigning blocks of OR time to a speciality.

3.1.1. Mixed integer programming

Beliën and Demeulemeester [9] describe the first models that consider the bed occupancy at the wards when creating an OR schedule. The models for building surgery schedules they propose and evaluate should result in leveled bed occupancy. Elective surgeries of the same type are scheduled in blocks. These blocks are assigned to days in the planning horizon while minimising either the highest expected bed occupancy peak or the highest bed occupancy variance. They also propose a model that minimises a combination of both. They take a stochastic LoS for each operated patient into account. The LoS is assumed to be a multinomial distribution function. Two approaches are used: a mixed integer programming (MIP) based approach, linear as well as quadratic, and a metaheuristic approach (simulated annealing). They conclude that the best results concerning solution quality and computation time are obtained by a combination of simulated annealing and a quadratic programming model. In the extension Beliën *et al.* [11] three objectives are taken into account: leveling the bed occupancy at the patient wards, minimising the sharing of ORs between different surgeon groups and making the MSS as repetitive as possible. The theoretical model is extended so it includes multiple recovery wards, which should lead to a more realistic model. They use the same solution approaches as [9].
In [9], the number of patients per operating room block was assumed to be deterministic dependent on type of surgery and fixed for each surgeon, while in [11] they assume a multinomial distribution function for this variable. The extension further consists of making flexible block sizes and allowing the allocation of OR time to individual surgeons instead of surgeon groups. They modify the mixed integer programming models such that they take the new objectives into account. Simultaneous incorporation of all three objectives leads to a large mixed integer program that becomes computationally intractable. Therefore, they use a modified version, that includes the new objectives, of the simulated annealing procedure described in [9]. The observations made with these procedures leads to two hierarchical goal programming approaches that both consist of two goal programming models that are solved successively. They develop a decision support system that incorporates all these solution approaches to develop an MSS. None of the solution approaches give an overall best solution, but the writers believe that the real power of the tool lies in visualising the OR-schedule and the resulting bed occupancy so managers can make informed decisions about their schedules.
Santibanez *et al.* [10] also develop a mixed integer programming model to schedule surgical blocks for each specialty. They study the best trade-off among OR time availability, downstream resources and patient waiting lists. In contrast to [11], they use averaged values for the LoS. The MIP model was used to deal with two different situations: maximising daily bed utilisation and determining maximum throughput and mix of patients. For the first objective the total amount of OR time and throughput of patients is given, for the second objective the availability of resources is known. They considered eight hospitals for scenario analysis. The results showed that they could reduce resource requirements such as beds, while maintaining the throughput of patients.
Yahia *et al.* [12] present a mixed integer linear programming (MILP) model that levels the daily beds

and nurses' workload, while considering surgeons preferences. They define a parameter that is a weighting factor between leveling the beds and leveling the nurses' workload. The MILP model is solved for different values of this parameter to find a desired solution. In this paper, no uncertainty in variables like the LoS is considered, which makes the model less realistic.
Fügener [3] combines a strategic with a tactical MSS problem, i.e.,they do not only decide what type of OR blocks to schedule when, but also decide how many blocks each specialty should get assigned. Their approach maximises hospital revenues while considering the ICU and general patient wards. A MILP model is set up to assign OR blocks to medical specialties, where the resource demand in ICs and wards is calculated by a stochastic patient flow model based on Fügener *et al.* [8]. By optimising revenues generated by surgeries the author hopes to give a reliable input for management decisions.

3.1.2. Analytical approach

Vanberkel *et al.* [7] and Vanberkel *et al.* [13] assign OR time to specialties, as in Beliën and Demeulemeester [9]. They believe there are downsides to mixed integer programming models as well as simulation models. They think that the scope of mixed integer programming models is limited, because it needs to remain solvable. And even though simulation models can capture a broad scope, they believe them to be inexact and the development time consuming. Therefore, they take an analytical approach that can be used to quickly evaluate a proposed MSS for additional factors. Their model computes the ward occupancy distributions, the patient admission/discharge distributions and the distributions for the ongoing interventions/treatments required by recovering patients. The model was applied at the Netherlands Cancer Institute-Antoni van Leeuwenhoek Hospital. In [13], a new MSS was developed by the authors together with stakeholders in the hospital. They chose the best MSS by swapping OR blocks and surgical specialty assignments. According to the authors a better MSS may have been found by a search heuristic, but this would lead to a very complex model that may not have gotten the same level of staff understanding and support. The authors want their model to be used as a decision support tool for both tactical and operational decision levels.
Fügener *et al.* [8] based their research on [13]. They also take an analytical approach, given an MSS they calculate the exact distributions of patients at downstream resources. Contrary to [13] they take multiple wards and the ICU into account, because this can be an important bottleneck. The objective is leveling bed demand and reducing weekend bed requests, so that the costs of downstream resources are minimised. For the exact objective function several heuristic solution methods are considered: swapping MSS blocks, a simulated annealing approach and a simple branch-and-bound approach. The latter is only applied to small instances. They also tested solution methods where they approximated the objective function and then used an exact solution method. However, for large instances the computation time became long. Fügener [3] based their patient flow model on the algorithm presented in [8] to calculate the exact distributions of patients at downstream resources. They extend it by including multiple ICs and outpatient flows.
Van Essen *et al.* [14] also use the analytical approach of [13] to determine the number of required beds. The developed model consists of a list of linear constraints and a complex nonlinear objective function. The objective of the model is to minimise the number of required beds, which should lead to a leveled bed occupancy. Two solution methods are used, a global approach: integer linear programming, and a local search heuristic: simulated annealing. To use the global approach, the objective function in linearised by replacing the objective function by the maximum of expected required beds. This yields better solutions than simulated annealing. It performs faster and gives better results. This is different from the conclusion of [9], where simulated annealing performs better. The paper also addresses what happens when you relax certain constraints. For example, allowing surgeries during the weekend gives a lower number of required beds.
As in [3], Fügener *et al.* [15] based their patient flow model on [8]. They extend it by including outpatients and emergency surgeries in weekends. The research aims at analysing the effect of three different MSS options. The first MSS represents the current practice at the hospital. The two other options are generated based on discussions with the management. One MSS is designed to avoid bed demand in the weekend, because working on weekends is more expensive. The last MSS is a compromise between the first and the second to avoid high peaks in bed occupancy during the week and also avoid bed occupancy at the weekend. The developed model determines, from historical data, the probabilities that certain number of patients are in the ward or ICU at certain days in the planning horizon. From this, they calculate the exact quantiles of the bed occupancy in the wards and the ICU for

these three MSS options. It is assumed that patients from the same specialty have they have the same chance of going through the ICU and the that they have the same LoS. With the three different MSSs, simulations were performed to find the impact on the bed occupancy at the downstream resources.

3.1.3. Other studies

Bekker and Koeleman [16] analyse the impact of variability in admissions and LoS on the required amount of bed capacity with an approximation method. When a weekly admission pattern is given, their method determines the mean bed occupancy of each day. They determine the admission quota for elective patients with a quadratic programming model that minimises the difference between the desired and expected occupancy. Another aim is to gain practical insights from their observations. They find that more variation in admissions leads to a higher variability in bed occupancy. Reducing the variability of the LoS only leads to less variable bed occupancy when the admissions variability is low. They conclude that arrivals should be evenly spread over the week to get the most time-stable performance.

3.2. Surgical case scheduling

Search terms: Surgical scheduling, operating room schedule, bed occupancy, variation
In this section we consider articles that schedule surgical procedures instead of assigning blocks to specialties. This is still on a tactical level, because types of surgical procedures are scheduled.

3.2.1. Mixed integer programming

Van Oostrum *et al.* [2] plan elective surgical types that are frequently performed in a cyclic schedule. The objective is to minimise the OR capacity and to level the requirement of hospital beds. Planned slack is used to incorporate the stochastic nature of surgery durations. However, the LoS is assumed to be deterministic. The solution approach consists of two phases. In the first phase the requirement of hospital beds is ignored and an integer linear program (ILP) is formulated and solved with an implicit column generation approach. In the second phase the problem is formulated as a MILP with the objective to minimise the maximum needed hospital beds, which can be solved within reasonable time bounds. They incorporated three different types of hospital beds by giving a priority factor to a certain type of bed.
Adan *et al.* [17] assign surgeries to a day in the cyclic schedule, as in [2]. However, they use a stochastic LoS, which outperforms a deterministic LoS. Multiple downstream resources are included: medium care beds, ICU beds and ICU-nursing staff. In contrast to [2] and [9], Adan *et al.* [17] are less interested in finding the best mathematical solution and more aimed at offering support and information that can be implemented in practice. The objective from [17] also differs from the objective of [2] and [9] as they do not minimise the required beds, but assume a fixed amount of beds and minimise the over- and under-utilisation of these beds. Their MILP with the LoS expressed stochastic was not solved to optimality, but the writers believe their solution is good. The extension of this model, Adan *et al.* [18], also accounts for emergency patients instead of just elective patients. They plan a higher number of patients than needed for the elective stream, so they adapt the tactical plan to the actual stream of patients. They solve their model to obtain solutions to the tactical problem for elective patients. Then the obtained elective schedule is modified using the flexibility so it also includes emergency patients. They use simulation to create an operational schedule based on the obtained tactical schedule with emergency patients.

3.2.2. Simulation

Min and Yih [19] investigate a stochastic surgery scheduling problem while considering ICU beds. Surgery durations and LoS on the ICU are assumed stochastic with known distributions. An important assumption they make is assuming all patients in the same surgical service group follow identical distribution of surgery duration. The LoS is generated randomly, where they use the same mean value and discrete distributions per surgical group. They formulate two solutions: one from a stochastic optimisation problem and one from a deterministic expected value problem. With a simulation study, they compare the performance of the two models. They claim the benefits of the stochastic optimisation problem over the deterministic expected value problem outweigh the higher computation costs.
Banditori *et al.* [20] propose a MIP model to find an MSS. A block scheduling approach is assumed

where a number of OR blocks is assigned to a specialty on a monthly basis. Patients are assigned to surgery groups within the specialties, see 3.4. The solution of the proposed model indicates, for each day of the month, how many cases from each surgery group need to be treated. Their objective is to maximise the number of surgeries planned while minimising violating due dates. However, instead of minimising variation of bed occupancy at the wards as in [11] and [2], they minimise the number of patients not being given a bed at their own ward. Next to the MIP model, they also do a simulation study to test the model solution's robustness. To find a good mix of robustness and efficiency, they combine the optimisation and simulation approach.
Chow *et al.* [21] developed a Monte Carlo simulation to predict the impact of an MSS on bed occupancy at the ward. The simulation model predicts the daily demand of beds in an uncapacitated system. The writers choose a trace-driven simulation, a simulation based on historical data. They also developed an MIP model based on [11] that levels bed occupancy. However, unlike other models, the model schedules both surgery blocks and the patient mix in each block. The objective is to minimise the total maximum bed occupancy of all wards. They developed two integer optimisation models, one to assign surgical blocks to days and one to determine the patient mix in each block.

3.2.3. Dynamic model

Given an MSS, Astaraky and Patrick [22] provide scheduling policies for all surgeries that minimises the time a patient spends on the waiting list, overtime in the ORs, and congestion in the wards. They model the scheduling problem as a Markov Decision Process (MDP). However, any realistic problem would make the model too large to solve. Therefore, it is necessary to use a version of simulation based approximate dynamic programming to solve the MDP. To demonstrate the success of the resulting policy, they test their model on data from a local hospital.

3.3. Assigning surgeries and sequencing surgeries

Search terms: Surgical scheduling, OR scheduling/planning, bed occupancy, bed demand, variation.
In this section we consider the articles that plan surgeries on an operational level. Two levels can be distinguished. During the first, specific surgeries are assigned an OR block and day in the planning horizon. The second consists of sequencing the assigned surgeries of each block. This is all done, such that the impact on downstream resources is limited.

3.3.1. Heuristics

Cardoen *et al.* [23] focus on surgical case scheduling on an operational level, instead of on a tactical level like [2] and [17]. Instead of making a cyclic schedule consisting of surgical cases, they start from a schedule where operating rooms are already assigned to specialties on particular surgery days. The strategy that they use in this paper consists of assigning patterns to the OR, while minimising multiple objectives using a branch and price methodology. Because the sequencing step depends on the assignment policies, [23] propose to enlarge the surgical case scheduling problem to find a qualitative surgery schedule without first assigning the ORs to specialities and surgery days.
Fei *et al.* [24] also determine a sequence of operations in each day, taking the availability of recovery beds into account. However, they assume a fixed amount of beds and maximise the utilisation. They use a column-generation-based heuristic procedure to construct weekly schedules and solve their daily scheduling problem by a hybrid genetic algorithm. As in [23], Aringhieri *et al.* [25] start with a given MSS filled with blocks assigned to specialties. The problem they address is determining the surgery date and operating room assigned to each patient from a set of patients waiting to be operated. They use expected duration of surgery and expected LoS. A Variable Neighbourhood Search methodology is used to explore the solution of the problem. The only resources taken into account are the ORs and recovery beds, where they aim at leveling the bed occupancy.

3.3.2. Stochastic model

Jebali and Diabat [26] develop a two stage stochastic program for scheduling the operating room while considering the beds at the wards and IC. With their model they want to avoid cancellations as much as possible. The paper could be viewed as an extension of [19], where they also propose a two-stage stochastic program that takes the beds at the wards into account. They extend it by also taking OR overtime capacity into account. They plan on an operational level, so they assume patients already have a hospitalisation date. The objective of the model is to minimise overall costs, where penalty costs are administered when patients' hospitalisation dates are moved. The first stage of the model assigns elective surgery cases to a day of the week. The second stage includes decisions regarding overtime and undertime of the OR and the utilisation of the ICU and the ward beds. The surgery durations are assumed to follow a log-normal distribution with known mean and standard deviation.

3.4. Clustering surgical procedures

When scheduling surgical types as in 3.2 often a classification is used to distinguish different patients and group them together. These surgery types consist of different surgeries with similar characteristics, for example duration of surgery, specialty and LoS. We cannot only group patients who undergo the same surgery, because this would lead to too many different groups. Therefore, a classification is needed to construct these surgery types. In this section we discuss different approaches in literature regarding this problem. Van Oostrum *et al.* [2] received data from the Erasmus MC in Rotterdam. In cooperation with surgeons they grouped medically homogeneous procedures.

Adan *et al.* [17] and the extension Adan *et al.* [18] take a different approach. Their research only takes the cardiothoracic ORs into account and using data from the Thorax centre in the Erasmus MC they made a distinction based on surgery duration and intensive care resources. They used a fixed classification of short/long surgery duration and short/medium/long ICU use. They find six adult patient groups, for which they gathered data based on medical guidelines. For the stochastic LoS they compare schedules obtained using empirical distributions and schedules using parametrised distributions fitted to the sample mean and sample standard deviation of the the LoS. Jebali and Diabat [26] use the same method as [17] and [18].

In Banditori *et al.* [20] patients are assigned to surgery groups, where each surgery group consists of procedures that have similar surgery durations, LoS and belong to the same specialty. In contrast to the distributions used by [18], they use expected values for LoS and surgery duration. They take three kind of wards into account where they assume that the wards are characterised by cases with the same LoS, i.e., short stay beds (LoS less than one day), medium stay beds (LoS equal to two days) and long stay beds (LoS more than 2 days). The surgery durations are also divided over three groups,i.e., short-, medium- and long-lasting surgeries.

Van Oostrum *et al.* [27] wrote a paper on clustering surgical procedures. The resulting surgery types should have the same characteristics like surgery duration and LoS. To get a feasible schedule, the characteristics of the surgical procedures should not differ too much. Together with surgeons, the surgery types were discussed, for example, they discussed if the surgeries within one type could all be performed by one surgeon. Surgery types that do not occur often enough to be scheduled in a cyclic schedule are grouped in so-called dummy surgeries. If the surgery types are narrowed to reduce the variability within the characteristics, this will lead to more dummy surgeries. They propose a method to obtain surgery types that minimises the number of dummy surgeries and the variability of surgery types. The solution approach they develop is based on Ward's Hierarchical Clustering Method [28]. They performed a case study at a Dutch hospital.

4

Mathematical model

In the previous chapter, we described different models to optimise the bed occupancy at the wards as found in literature. In this chapter, we describe the construction of our model. The LUMC has worked with a two week cyclical schedule for years. To increase the acceptance within the LUMC we continue with a tactical schedule that repeats itself every two weeks. However, instead of scheduling whole OR-days assigned to a specialty, we make a schedule that specifies per specialty what types of procedures they should plan on what day. In Section 4.1, the concept of the model is explained. In Section 4.2, a method to cluster surgical procedures based on LoS and surgery duration is given. In Section 4.3, we formulate the constraints for our basic model. Then we describe a model that analytically determines the distribution of post-operative patients on the ICU and wards from a given cyclic schedule, in Section 4.4. Lastly, we describe the objective function of the model with a focus on minimising the bed demand and maximising OR utilisation.

4.1. Conceptual model

Creating cyclic schedules is often done by block scheduling where the OR blocks are assigned to different groups. The groups assigned to them can range from being defined as an entire specialty to being defined as specific surgical procedures. For example, in the first case, all types of procedures within cardiothoracic surgery will be put into one group and in the second case, a bypass surgery and an aortic valve replacement are in different groups, all within the cardiothoracic surgical specialty. The appropriate level of aggregation depends on the quality and volume of available information. In most literature, whole specialties are assigned to OR blocks based on the average number of surgeries they perform during one OR block. Often, the OR blocks are defined as whole days in an OR. This means the specialties decide how many surgeries are performed on each day. As described in Section 2.3, scheduling with OR blocks, where multiple surgeries are performed in one block, can lead to problems when put into practice. Using the average number of patients that have surgery in one OR block can over- or underestimate the actual number of patients that have surgery in that OR block. During the week, this can cause unwanted variation in the bed occupancy.

Prior research was conducted in the LUMC where groups were defined as whole specialties and OR blocks defined as whole or half OR-days. The downside of this approach is the diversity within the specialties. The LUMC is an academic hospital that performs more specialised procedures than a non academic hospital. According to the data of 2015, about 7% of all surgeries are unique surgeries that happen only once a year. Surgeries that happen less than ten times a year cover 40% of all surgeries in a year. Therefore, defining groups as particular surgical procedures will lead to groups with surgeries that do not occur enough to be scheduled in a cyclic schedule of two weeks and defining groups as whole specialties leads to a large diversity within the groups. Thus, we formulate groups somewhere in between these cases, based on LoS and surgery duration. We denote these groups by the term surgery groups.

In order to address the main problems, OR utilisation and bed occupancy, we propose to apply an approach to create a cyclic schedule on the tactical level based on the research of Fügener [3] and Van Oostrum *et al.* [2].

First, we define surgery groups that contain surgical procedures with similar expected surgery durations and expected LoS, see Section 4.2. The surgery groups consist of surgical procedures of one specialty that are similar in surgery duration and LoS. For the surgery duration of a surgery group, we use the session time, i.e., the time between a patient entering and leaving the OR. The surgery duration of a surgery group is not set, but is a stochastic variable based on historical data of all procedures in that group.

After defining the surgery groups, we use a model to find an optimal OR schedule, see Section 4.3. In the model, multiple restrictions are taken into account, e.g., the need for specific ORs due to available equipment, the total available OR time during opening hours and the number of scheduled surgeries per surgery group.

We define an OR-day as a combination of a day in the cyclic schedule and an OR. The model assigns surgery groups to ORs on each day of the cyclic schedule, see Figure 4.1. If a surgery group is assigned to OR 1 every Monday, this means that in the operational planning a patient with a surgery from that surgery group is scheduled in OR 1 on Monday. Multiple surgery groups can be assigned to the same day in the same OR, because multiple patients can be operated on in the same OR on the same day. Assigning a surgery group to an OR-day in the schedule means a single patient with a surgery in that surgery group will be operated on during that day in that OR. Therefore, surgery groups can be assigned multiple times to the same OR on the same day. Then, multiple patients with procedures from the same surgery group are operated on that day. The surgery duration depends on the surgery group. Thus, a variable amount of surgery groups can be scheduled on an OR-day in the cyclic schedule, depending on the length of the surgeries from the assigned surgery groups.

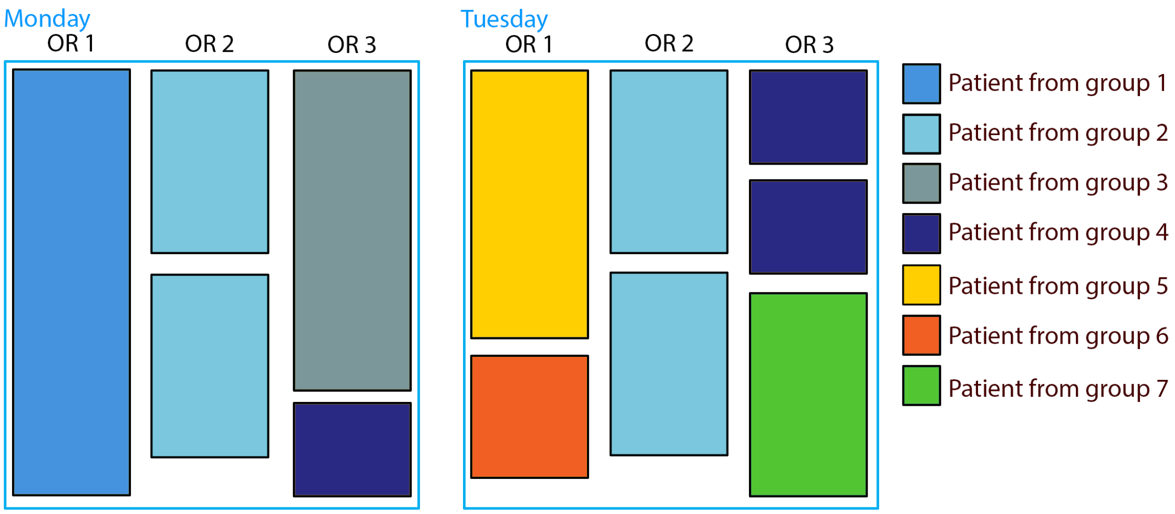

Figure 4.1: OR schedule example.

The model finds feasible solutions for the OR schedule. However, we also want to take the downstream resources into account. We assume that the LoS of surgery types is stochastic. The bed occupancy at the ICUs and wards for a given schedule is determined based on the stochastic LoS by using a model based on Fügener [3]. The objective of the model is to minimise the bed occupancy at the wards and ICUs, while maximising the OR utilisation.

The output of this model is a two weekly cyclic schedule for all ORs and specialties. The schedule is optimised for normal weeks, so reduction weeks, e.g., summer and Christmas, are not included.

4.2. Clustering

In this section, we present a method to determine surgery groups based on LoS and surgery duration. Section 4.2.1 gives a short introduction into datamining techniques. In Section 4.2.2, we define the method to cluster surgical procedures into surgery groups.

4.2.1. Literature

Datamining techniques can be split into two main categories: supervised learning and unsupervised learning. With supervised learning there is labeled training data, i.e., a response variable that labels each instance. The other variables are predictor variables. With the classification, response variables are explained with predictor variables [29]. We would like to use the variables *specialty* and *procedure* to explain the variables *surgery duration* and *LoS*. The response variable would then be a certain class like short LoS and short surgery duration. However, these labels are dependent on the classification we would like to make, and are therefore, not available. The other category, unsupervised learning, assumes unlabelled data and does not split the variables into response and predictor variables. Common clustering algorithms examine the data to find groups with similar instances. We would like instances with the same specialty and procedure to be in one cluster, so they can easily be planned. However, most clustering algorithms assume instances are independent. Moreover, in most clustering algorithms we cannot specify what type of clusters we want. This means that one cluster can consists of instances where the dispersion of LoS is small and the dispersion of surgery duration is very large, but another cluster can consist of instances where the dispersion of LoS is large and that of the surgery duration is small, see Figure 4.2. Therefore, we take a different approach which is explained in Section 4.2.2.

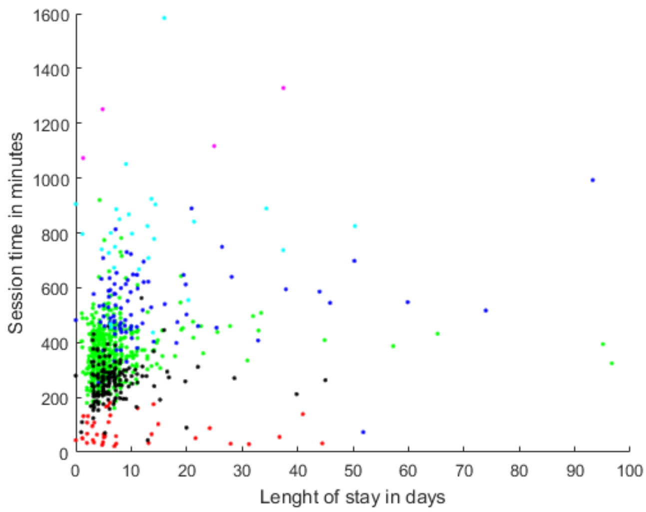

Figure 4.2: Clusters made for the specialty cardiothoracic surgery with a k-means algorithm in MATLAB with $k = 6$, based on the session times and lenght of stay.

To evaluate the performance of our determined classes, we use a confusion matrix, see Table 4.1. In the confusion matrix, the rows represent the instances in a predicted class, while the columns represent the instances in the actual class. In the standard confusion matrix, we define two classes: positive (P) and negative (N). If an instance belongs to a certain group, then it should be labeled positive. If the instance does not belong to the group, it should be labeled negative. How the groups are defined influences the labeling of instances. For each cell in the matrix this leads to fields as true positives, false positives, false negatives and true negatives. These are defined as:

- False positive (FP): Falsly predicting a positive label.
- False negative (FN): Falsly predicting a negative label.
- True positive (TP): Correctly predicting a positive label.
- True negative (TN): Correctly predicting a negative label.

The accuracy of the clustered groups is given by the percentage of correctly classified instances divided by the total number of instances (TP + TN/ Total population). However, when a dataset has significantly more instances in one class than in the other class, then the accuracy is not representative of the true performance. For example, suppose there are 90 instances that are negative and only 10 that are positive. If the clustering is biased towards the negative class and all the instances are in the class negative, then the accuracy is still 90%, while the accuracy of the class positive is 0%. Therefore, it is important to also take other metrics into account. The specificity (TN ratio) is the proportion of instances that are labeled negative (TN) of all the instances that are negative (TN + FP). How higher the specificity, the fewer instances that are positive are labeled as negative. The sensitivity (TP ratio), is the proportion of instances that are labeled positive (TP) of all the instances that are actually positive (TP + FN). With higher sensitivity, less positive instances are classified as negative. The precision, another important metric, is the fraction of correctly labeled instances that are relevant given by the proportion of true positives (TP) of all predicted positives (TP + FP). These metrics help us to evaluate the clustering of procedures.

	P′ (Predicted)	N′ (Predicted)
P (Actual)	True Positive (TP)	False Negative (FN)
N (Actual)	False Positive (FP)	True Negative (TN)

Table 4.1: Confusion Matrix

4.2.2. Approach

We cluster surgical procedures into surgery groups based on historical data. These surgery groups need to be specialty specific, so procedures of only one specialty may occur in a surgery group. To make sure we can fill a whole OR-day with surgeries from one specialty. We use the average surgery duration and the median LoS to cluster procedures into groups. Then, we use a confusion matrix to determine how well the clustering method performed. To find the best clusters, we compare the accuracy and precision of the confusion matrices of different outcomes.

Let B be the set of all patients that are operated on in 2015. Let P be the set of all procedures that have been performed. Procedures are registered with an open text field. This leads to a lot of double entries for the same procedure, e.g., the following inputs are registered as three different procedures:

ABOVE KNEE - AMPUTATION
ABOVE KNEE - AMPUTATION - LEFT
ABOVE KNEE - AMPUTATION - RIGHT

To minimise the number of unique procedures, we discussed the procedure lists with surgeons of the main surgical specialties. This lead to a shortened list of codes C that each defines multiple comparable procedures as the three procedures mentioned above. Subset P_c includes the procedures that belong to code c. Each procedure p should be assigned to exactly one code c. Therefore, $P_c \cap P_{\hat{c}} = \emptyset$ for all $c \neq \hat{c}$ and $\bigcup_{c \in C} P_c = P$. However, this still leads to over 2000 procedure unique codes. Because we cannot schedule all of these separately in our tactical schedule, we cluster the procedure codes based on LoS and surgery duration.

We first divide the procedures based on their LoS. We define two groups: short stay and long stay. These two groups are chosen in consultation with surgeons and ward management. They recognise these two types of patients. The ward personnel finds that either a patient recovers well and leaves the hospital after a short stay, or the medical condition of the patient does not improve, which often means that the patient becomes a long stay patient. The groups we define here do not define the ward a patient goes to. A short stay patient may go to another ward than the short stay ward. The LoS is typically highly skewed as described in Carter and Potts [30]. It has a long tail on the right side, see Figure 4.4, which is the length of stay of patients of the specialty cardiothoracic surgery. Therefore, using the median LoS instead of the mean LoS leads to better results. We performed the algorithm with both the mean and the median and compared the precision and accuracy of the outcomes. When using the median LoS to define the two groups, both the accuracy and the precision were overall higher.

For every specialty, we want to find a definition of the short stay and long stay group. For example, we could say that every surgery that has a median LoS less than 4 days is assigned to the short stay group while every surgery that has a median LoS of more than 4 days is assigned to the long stay group. However, we want to define this threshold between the short stay and long stay group per specialty based on the historical data. Therefore, we use an algorithm to find a threshold between the groups that leads to the highest precision for both groups. An optimal threshold is determined for each specialty. This threshold is then used to define the short stay and long stay groups. A pseudocode for the algorithm can be found in Algorithm 1. We let a threshold iterate between the smallest value for which the group size of the short stay group is still 20% of the total to the largest value for which the size of the long stay group is still 20% of the total. This is done to ensure that the size of both groups does not become too small. Groups that are too small cannot be scheduled in a cyclic schedule. For each threshold, we determine the short stay and long stay group. Every procedure with a median less than or equal to the threshold is assigned to the short stay group, otherwise, the procedure is assigned to the long stay group. Then, we determine the corresponding precision of both groups. We want to use the threshold with the highest precision for both the groups. If we sum the precisions, one might be very high while the other is low. Therefore, we look at the minimum of the precision of the short stay group and the precision of the long stay group. The threshold for which this minimum is the highest is chosen. The procedures are divided into the groups based on their median length of stay. In Figure 4.3, we see for each possible threshold the percentage of instances that are in the short stay group based on the median LoS of the procedure, but who had a realised LoS of more than the threshold. We also see the percentage of instances that are in the long stay group based on the median LoS, but had a realised LoS of less than the threshold. For this example, 7.7 days would be the best possible threshold between the long stay and short stay group. To determine if the resulting groups are significantly different we use a two-sample *t*-test with a 5% significance level. The null hypothesis is that the groups have equal medians, the alternative hypothesis is that groups come from populations with unequal medians.

```
Data: Historical data of a specialty. List of performed procedures with LoS.
Result: Procedures clustered into day, short and long stay.
initialization;
a = threshold group size short 20% of total;
b = threshold group size long 20% of total;
for a ≤ threshold ≤ b, per 0.1 day do
    group all procedures with median LoS less than or equal to the threshold in short stay;
    group all procedures with median LoS greater than the threshold in long stay;
    precision in short stay = #instances in short stay with a LoS less than threshold divided by
      the total in short stay;
    precision in long stay = #instances in long stay with a LoS greater than threshold divided by
      the total in long stay;
    accuracy = (correctly classified short + correctly classified long) divided by total number of
      instances;
    precision overall = minimum( precision in long stay, precision in short stay);
end
find threshold with maximum precision overall;
divide procedures into groups for that threshold;
```

Algorithm 1: Find the optimal grouping of procedures into short stay and long stay groups.

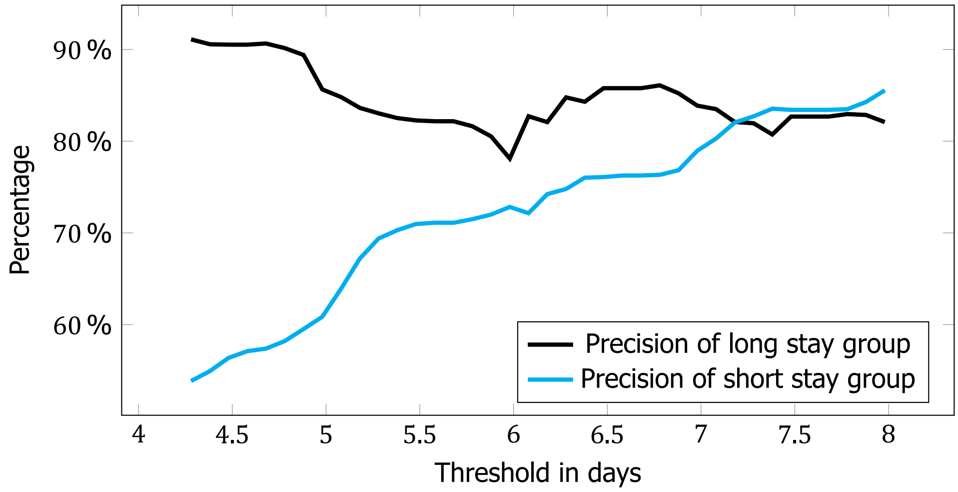

Figure 4.3: For each possible threshold between short stay and long stay, the precision of the short stay group and the long stay group.

In Figure 4.4, we see a normalised histogram based on historical data, that depicts the LoS of patients from the specialty cardiothoracic surgery. In Figure 4.5, we see the same data, but divided into short stay and long stay groups. For this specialty, the threshold found by the algorithm was 7.7 days.

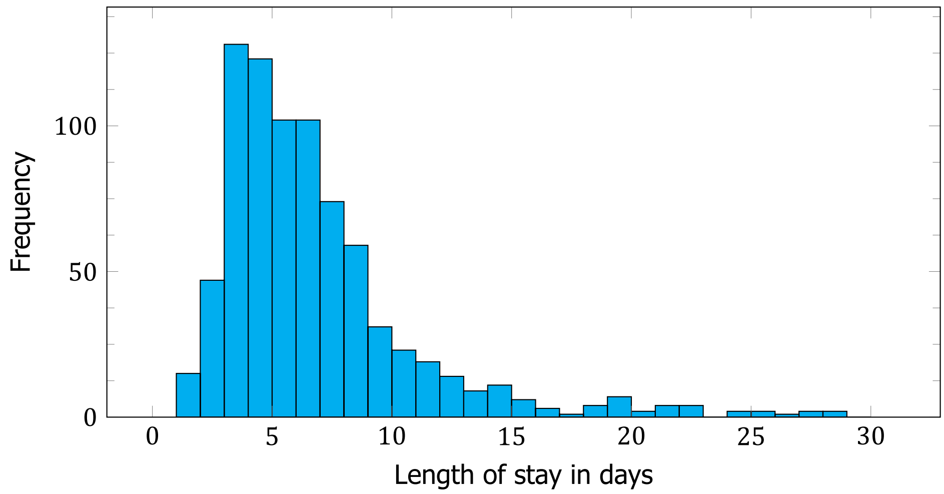

Figure 4.4: Probability distribution of the number of days a patient of the specialty cardiothoracic surgery stays in the ward based on historical data.

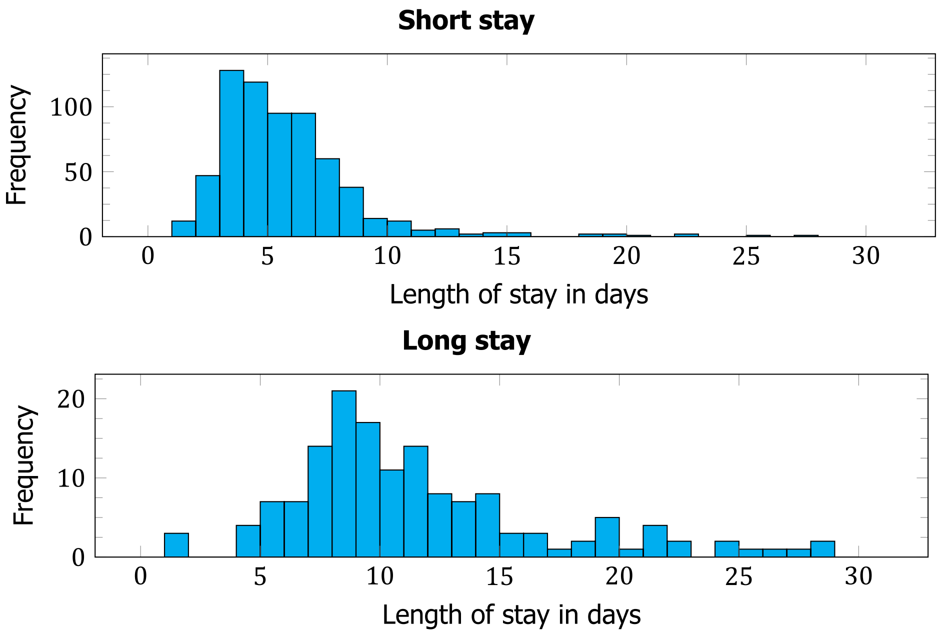

Figure 4.5: Probability distribution of the number of days a patient of the specialty cardiothoracic surgery stays in the ward based on historical data for each LoS group.

After dividing the procedures into LoS groups, we divide each group again into different groups based on surgery duration. This is done similarly to defining the LoS groups. However, when dividing the surgical procedures based on surgery duration, the mean surgery duration was a better predictor than the median surgery duration. The accuracy of the groups was higher when we used the mean surgery duration instead of the median surgery duration. For each group, short stay and long stay, thresholds are determined based on historical data. We define three groups, so two thresholds, based on surgery duration: short surgery duration, middle surgery duration and long surgery duration. To determine if the resulting groups are significantly different we use a two-sample t-test with a 5% significance level. The null hypothesis is that the groups have equal means, the alternative hypothesis is that groups come from populations with unequal means.
If the difference between the longest surgery duration and the shortest surgery duration in a LoS group is less than 180 minutes, creating three surgery duration groups does not lead to a rejection of the null hypothesis. This means the groups are not significantly different. Therefore, we only define two groups based on surgery duration in that case. The pseudocode for defining three groups is shown in Algorithm 2. This gives on average 6 groups, namely, the combinations of short, middle or long surgery duration and short or long length of stay.

In literature, it is often assumed that patients from the same specialty or surgery group all go to one specific ward or ICU. However, hospital data indicates this is not the case. Patients who have had the same surgery do not always go to the same ward. For example, patients who are in good condition might go to the short stay ward, while an older patient who had the same surgery is transferred to the long stay ward, because the LoS is expected to exceed the maximum LoS of the short stay ward. Therefore, patients from the same surgery group do not go to the same ward in our model. For each group, the chance to go to a certain ward is calculated from historical data.

Data: Historical data of a specialty. List of performed procedures with surgery duration in minutes.
Result: Procedures clustered into short, middle and long surgery duration.
initialization;
a = threshold group size short 20% of total;
b = threshold group size short 60% of total;
d = threshold groups size long 80% of total;
for *a* $\leq$ *threshold1* $\leq$ *b, per 5 minutes* **do**
    c = threshold2 such that group size middle at least 20% of total with respect to threshold1;
    for *c* $\leq$ *threshold2* $\leq$ *d, per 5 minutes* **do**
        group all procedures with mean surgery duration less than or equal to threshold1 in short surgery duration;
        group all procedures with mean surgery duration greater than threshold1 and less than or equal to threshold2 in middle surgery duration;
        group all procedures with mean surgery duration greater than threshold2 in long surgery duration;
        precision in short = # instances in short with a surgery duration less than or equal to threshold1 divided by the total in short;
        precision in middle = # instances with mean surgery duration greater than threshold1 and less than or equal to threshold2 divided by the total in middle;
        precision in long = # instances in long with a surgery duration greater than threshold2 divided by the total in long;
        accuracy = (correctly classified short + correctly classified middle + correctly classified) long divided by total number of instances;
        precision overall = minimum(precision in short, precision in middle, precision in long);
    end
end
find thresholds with maximum precision overall;
divide procedures into groups for that threshold based on mean surgery duration;

Algorithm 2: Find the optimal grouping of procedures into surgery duration groups.

4.3. Restrictions

In this section, we describe several restrictions on the OR-schedule that are relevant for the LUMC. All indices and sets, parameters and variables, shown in Tables 4.2, 4.3 and 4.4 respectively, are also explained in the text. Let $\mathcal{O}$ be the set of given ORs and $\mathcal{K}$ be the set of OR-days in the cyclic schedule. An OR-day (o, k) is defined as a combination of day k of the cyclic schedule and OR o. The set of given surgery groups $\mathcal{J}$ are assigned to a day $k \in \mathcal{K}$ in OR $o \in \mathcal{O}$, so OR-day (o, k). Multiple surgery groups can be assigned to the same day in the same OR and a surgery group can be assigned multiple times to one OR on the same day, see OR 2 in Figure 4.1. This means that we do not have a predefined number of surgeries for each OR-day. The schedule we generate does not specify the order in which these surgeries should take place. This is determined by physicians during the operational planning.
We use the integer decision variables $z_{o,k,j}$ to specify the number of times surgery group j is scheduled on OR-day (o, k). The model of Van Oostrum minimises the needed OR capacity, while we want to maximise the OR utilisation. Therefore, we do not schedule a fixed number of surgeries per surgery type, but set a lower bound β_j on the number of scheduled surgeries per surgery type. By doing this, we assume that waiting lists are long enough to schedule a variable amount of surgery types in each cycle of the cyclic schedule. The following constraint ensures that all groups $j \in \mathcal{J}$ are scheduled a minimum of β_j times.

$$\sum_{\substack{o \in \mathcal{O}, \\ k \in \mathcal{K}}} z_{o,k,j} \geq \beta_j, \qquad \forall j \in \mathcal{J}. \tag{4.1}$$

Our basic schedule is based on the current MSS. On OR-days that are assigned to a specific specialty, we schedule only surgery groups of that specialty. Let $\mathcal{S}$ be the set of specialties and $J_s \subseteq \mathcal{J}$ the surgery groups of specialty s. We introduce binary decision variables $u_{o,k,s}$, which are equal to one if surgery groups of specialty s are planned on OR-day (o, k) and zero otherwise. Furthermore, we introduce the binary parameter $\epsilon_{o,k,s}$, which is equal to one if specialty s is assigned to OR-day (o, k) in the current MSS and zero otherwise. Constraint 4.2 ensures that only the surgery groups of the specialty that is assigned to OR-day (o, k) are scheduled on that day.

$$u_{o,k,s} \leq \epsilon_{o,k,s}, \qquad \forall o \in \mathcal{O}, \forall k \in \mathcal{K}, \forall s \in \mathcal{S} \tag{4.2}$$

The relation between $z_{o,k,j}$ and $u_{o,k,s}$ is given by constraint 4.3, where M_s is the maximum number of surgeries of a specialty s on one OR-day.

$$\sum_{j \in \mathcal{J}_S} z_{o,k,j} \leq M_s \cdot u_{o,k,s}, \qquad \forall o \in \mathcal{O}, \forall k \in \mathcal{K}, \forall s \in \mathcal{S} \tag{4.3}$$

ORs are only open for a couple of hours each day. The number of surgeries we can schedule on an OR-day is limited by the surgery duration of the surgery groups we assign to the OR-day. We denote the surgery duration by ζ_j for each surgery group $j \in \mathcal{J}$. This is a stochastic variable with mean μ_j and variance σ_j^2. Let $g_{o,k}$ denote the probability distribution of the total session time of the surgery types that are scheduled on OR-day (o, k), for solution ψ. The available OR time on day k in OR o is denoted by $\tau_{o,k}$. We introduce constraint 4.4, to ensure that the probability that overtime occurs is less than α. This constraint ensures that a certain amount of slack is planned to buffer against overtime. Overtime occurs when the total time needed for all surgeries in the OR is more than the total available time in that OR.

$$P(g_{o,k} \leq \tau_{o,k}) \geq 1 - \alpha, \qquad \forall o \in \mathcal{O}, k \in \mathcal{K} \tag{4.4}$$

On a regular OR-day, it is required that only one surgeon performs all surgeries scheduled that day. The surgeons are highly specialised and can only perform a subset of procedures of their specialty. Surgeons have regular days in the OR, however due to other responsibilities, e.g., teaching, shifts in the outpatient clinic or working in affiliated hospitals they are available on an irregular basis. The surgeons schedules change too much during the year to take them into account for a tactical schedule. Therefore, we use a more general constraint to ensure that a whole OR-day only gets assigned surgery groups from the same specialty. We introduce binary parameter $\chi_{o,k}$, which denotes if an OR o is open on day k. Constraint 4.5 ensures that only surgery groups of the same specialty are planned on an OR-day.

$$\sum_{s \in S} u_{o,k,s} \leq \chi_{o,k}, \qquad \forall o \in \mathcal{O}, \forall k \in \mathcal{K} \tag{4.5}$$

Furthermore, a surgery group cannot be assigned to a random OR, because, special equipment might be needed for procedures in that surgery group. We define a set of OR types $\mathcal{R}$ and for all surgery types $j \in \mathcal{J}$ we denote the subset of surgery groups that can be performed in OR type $r \in \mathcal{R}$ by $J_r \subseteq J$. We introduce binary decision variables $v_{o,k,r}$, which are equal to one if surgery groups of type r are scheduled on OR-day (o, k) and zero otherwise. The number of available ORs of type $r \in \mathcal{R}$ is different each day and is given by $\eta_{k,r}$. This leads to the following constraint:

$$\sum_{o \in O} v_{o,k,r} \leq \eta_{k,r}, \qquad \forall k \in \mathcal{K}, \forall r \in \mathcal{R}. \tag{4.6}$$

We relate decision variable $v_{o,k,r}$ to $z_{o,k,j}$ with the following constraint:

$$\sum_{j \in \mathcal{J}_r} z_{o,k,j} \leq N_r v_{o,k,r}, \qquad \forall o \in \mathcal{O}, \forall k \in \mathcal{K}, \forall r \in \mathcal{R}, \tag{4.7}$$

where N_r is the maximum number of surgeries belonging to OR type r on one OR-day.

Index ∈ set	Description
$o \in \mathcal{O}$	ORs
$k \in \mathcal{K}$	Days in cyclic schedule on which ORs are open
$j \in \mathcal{J}$	Surgery groups
$s \in \mathcal{S}$	Specialties
$j \in \mathcal{J}_s$	Surgery groups from specialty s
$r \in \mathcal{R}$	OR types
$j \in \mathcal{J}_r$	Surgery groups with OR type r

Table 4.2: Sets and indices.

Parameters	Description
β_j	How many times surgery group j needs to be scheduled
$\epsilon_{o,k,s}$	Binary parameter that equals one when specialty s is assigned to OR-day (o, k)
M_s	Maximum number of surgeries of specialty s on one OR-day
α	Probability of overtime
$\tau_{o,k}$	Time available in OR o on day k
$\chi_{o,k}$	Binary parameter that denotes if an OR is open on day k
$\eta_{k,r}$	Number of available OR types r on day k
N_r	Maximum number of surgeries needing OR type r on one OR-day

Table 4.3: Parameters

Variables	Description
$z_{o,k,j}$	Integer decision variables that denote the number of times surgery group j is scheduled on OR-day (o,k)
$u_{o,k,s}$	Binary decision variables, which are equal to one if specialty s is scheduled on OR-day (o,k)
$g_{o,k}$	Probability distribution of total session time of the surgery types on day k in OR o
$v_{o,k,r}$	Binary decision variables, which are equal to one if on OR-day (o,k) OR type r is needed

Table 4.4: Variables

4.4. Patient distribution

In this section, we describe a model that calculates the distribution of post-operative patients in the downstream units for a given cyclic schedule. Determining the patient distribution is done in three steps, where we include multiple ICUs and wards. The first step is to calculate the distribution of recovering patients in the ICUs and wards for each surgery group, see Section 4.4.1. Step two is to calculate the distribution of patients resulting from scheduling a surgery group in a cyclical schedule, see Section 4.4.2. Here we take patients into account that had surgery in one cycle, but are discharged in a later cycle. They influence all the cycles where they are still occupying a bed. The last step is to combine all surgeries from a cyclic schedule and calculate the occupancy levels on the ICUs and wards, see Section 4.4.3. This last step needs to be repeated for every cyclic schedule. The first two steps can be done beforehand. Section 4.4.4, describes the resulting objective function.

Index ∈ Set	Description
$i \in \mathcal{I}$	ICUs
$w \in \mathcal{W}$	Wards
$j \in \mathcal{J}_i$	Surgery groups that go to ICU i
$j \in \mathcal{J}_w$	Surgery groups that go to ward w
$n \in \{1, ..., N_j^I\}$	Days in the ICU
$n \in \{1, ..., N_j^{WS}\}$	Days in ward after surgery
$n \in \{1, ..., N_j^{WI}\}$	Days in ward after ICU
$l \in \mathcal{L}$	Days in cyclic schedule

Table 4.5: Sets and indices.

Table 4.5 presents a summary of sets and indices that are used in the model. In Table 4.6 and 4.7, parameters are presented. The parameters and sets and indices are also presented in the text. This approach is based on the approach of Fügener [3]. It is extended by assuming that patients of the same surgery group do not all go to the same ward, but have a probability of going to a certain ward versus another. The same is assumed for ICUs. Furthermore, [3] use OR blocks that cover a whole OR-day and specialties are assigned to an OR-day. In our model we assign multiple surgery groups to one OR-day (o,k). For each assigned surgery group, one patient will be scheduled during operational planning.

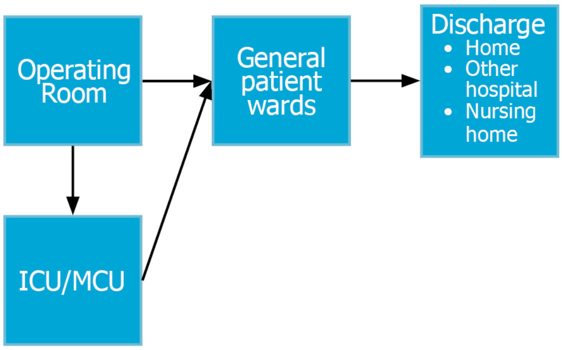

Figure 4.6: Most common patient flows from operating room to discharge.

We assume that most patients take one of two paths: from surgery directly to a ward (WS), or from surgery first to the ICU (I) and then to the ward (WI), see Figure 4.6. After staying on the ward, patients are discharged. We assume that patients do not go back from the ward to the ICU. Patients that are directly discharged from the ICU are modelled by a ward stay of zero days. A few patient flows are not covered in this approach. They are redefined such that the stay at the ICU is directly after their surgery. As [3] assumes that all patients from one surgery group are admitted in the same ICU or ward, they do not distinguish between different ICUs and wards. However, as explained in Section 4.2, patients who belong to the same surgery group do not always go to the same ward. Therefore, we do distinguish between different ICUs and wards. Let the set $\mathcal{I}$ denote all ICUs and MCUs and let the set $\mathcal{W}$ denote all the surgical patient wards. For all $j \in \mathcal{J}$ we define the subsets $\mathcal{J}_i \subseteq \mathcal{J}$ and $\mathcal{J}_w \subseteq \mathcal{J}$ as the surgery groups who go to ICU i and ward w respectively. The LoS (in days) in ICU i or ward w of each surgery group is modelled by discrete empirical distributions obtained from LUMC records. The empirical distribution of the LoS are the same for patients from the same surgery group, regardless of which ward they are transferred to. The patient flow model requires additional empirical data that can be obtained from hospital information systems. The following input parameters are required for every surgery group $j \in \mathcal{J}$:

- $a_{i,j}$ represents the probability that a patient of surgery group j is transferred to ICU i after surgery.
- $b_{w,j}$ represents the probability that a patient from surgery group j is transferred to a certain ward w after surgery or ICU.
- $c^{I}_{j,n}$ represents the probability that a patient from surgery group j stays exactly n days in the ICU after surgery.
- $c^{WS}_{j,n}$ represents the probability that a patient from surgery group j stays exactly n days in the ward after surgery.
- $c^{WI}_{j,n}$ represents the probability that a patient from surgery group j stays exactly n days in the ward after a stay on the ICU.

Here, the probability a patient from surgery group j goes to the ICU is given by $\sum_{i\in I} a_{i,j}$. The probability a patient from surgery group j goes to the ward is given by $1 - \sum_{i\in I} a_{i,j}$. The probabilities $c^{I}_{j,n}$, $c^{WS}_{j,n}$ and $c^{WI}_{j,n}$ are not given separately for every ward/ICU, because for every surgery group j the probability of a patient staying exactly n days on the ward/ICU is the same for every ward/ICU. Their LoS does not depend on the ward/ICU they are transferred to. In our model we assume that when a patient is discharged on day n he is still on the ward or ICU on that day. Often patients are only discharged in the afternoon, while admissions are during the morning.

4.4.1. Single surgery group

We start by calculating the conditional probabilities $d^{I}_{j,n,w}$, $d^{WS}_{j,n}$ and $d^{WI}_{j,n}$ assuming surgery took place on day 1. Equation (4.8) is the conditional probability that a patient from surgery group j who is still in the ICU on day $n+1$, which is n days after surgery, is transferred to the ward on day $n+1$. We calculate this by dividing the probability that the patient stays exactly n days by the sum of the probabilities that the patients stays n days, $n+1$ days, $n+2$ days and so on until the maximum number of days that patients from surgery group j can stay in the ICU, N_j^I. Following the same logic, we calculate the probability that a patient from surgery group j, who is in the ward on day n, is discharged on day n, see Equation (4.9). Here $d_{j,1}^{WS}$ gives the probability that a patient from surgery group j leaves the hospital directly after surgery on day 1 (e.g. in case of death, transfer to another hospital or nursing home). Equation (4.10), is the probability that a patient from surgery group j, who is in the ward on day n after being transferred from the ICU, is discharged on day n, where we assume the patient is transferred from the ICU on day 1. N_j^{WS} and N_j^{WI} are the maximum number of days patients from surgery group j stay in the ward after surgery and the ward after a stay in the ICU, respectively. We define $N_j^W = \max(N_j^{WS}, N_j^I + N_j^{WI})$ as the maximum number of days a patient from surgery group j stays in the hospital, where the ICU and ward are both included.

$$d_{j,n+1}^I = \frac{c_{j,n}^I}{\sum_{k=n}^{N_j^I} c_{j,k}^I}, \qquad j \in \mathcal{J}, n \in \{1, ..., N_j^I\} \tag{4.8}$$

$$d_{j,n+1}^{WS} = \frac{c_{j,n}^{WS}}{\sum_{k=n}^{N_j^{WS}} c_{j,k}^{WS}}, \qquad j \in \mathcal{J}, n \in \{0, ..., N_j^{WS}\} \tag{4.9}$$

$$d_{j,n+1}^{WI} = \frac{c_{j,n}^{WI}}{\sum_{k=n}^{N_j^{WI}} c_{j,k}^{WI}}, \qquad j \in \mathcal{J}, n \in \{0, ..., N_j^{WI}\} \tag{4.10}$$

With $d_{j,n}^I$, $d_{j,n}^{WS}$ and $d_{j,n}^{WI}$ we determine the probabilities $e_{j,n}^I$, $e_{j,n}^{WS}$ and $e_{j,n}^{WI}$. $e_{j,n}^I$, the probability that a patient from surgery group j who had surgery on day 1, is still in the ICU on day n is determined in Equation (4.11). For $n = 1$ this is simply the probability that the patient is transferred to the ICU after surgery. We assume a patient stays at least one day in the ICU, otherwise it is modelled as going directly to the ward. Therefore, for $n = 2$ we have the same probability as for $n = 1$. For $n \in \{3, ..., N_j^I + 1\}$ this is the probability that the patient was not transferred to the ward the day before, day $n-1$, multiplied with the probability that the patient was still in the ICU the day before. Following the same line of thought we find Equation (4.12), which gives the probability that a patient from surgery group j who had surgery on day 1, is still in the ward on day n. In Equation (4.13) we determine the probability that after an ICU stay of m days, a patient from surgery group j is still in the ward on day n. Here, m can range from day one to the maximum number of days patients of surgery group j can stay in the ICU. The number of days a patient stays in the ward after a stay of m days in the ICU is given by $n-m$. Therefore, n ranges from $n = m$ to $n = m + N_j^{WI} + 1$. For $n = m$ and $n = m+1$, $e_{j,n,m}^{WI}$ is given by the probability that a patient is not being discharged straight after their stay of m days in the ICU $(1 - d_{j,1}^{WI})$, multiplied with the probability that the patient was in the ICU on day m ($e_{j,m}^I$) multiplied by the probability that he was transferred from the ICU on day m ($d_{j,m}^I$). For $n \in \{m+2, ..., m + N_j^{WI} + 1\}$, we multiply the probability that a patient from surgery group j is not discharged after a stay of $n-m$ days on the ward, so $n-1$ days after the surgery, with the probability that the patient was in the ward on day $n-1$.

$$e_{j,n}^I = \begin{cases} \sum_{i \in I} a_{ij}, & \text{for } n = 1, 2 \\ \left(1 - d_{j,n-1}^I\right) e_{j,n-1}^I, & \text{for } n \in \{3, ..., N_j^I + 1\} \\ 0, & \text{otherwise} \end{cases} \tag{4.11}$$

$$e_{j,n}^{WS} = \begin{cases} 1 - \sum_{i \in I} a_{ij}, & \text{for } n = 1 \\ \left(1 - d_{j,n-1}^{WS}\right) e_{j,n-1}^{WS}, & \text{for } n \in \{2, ..., N_j^{WS} + 1\} \\ 0, & \text{otherwise} \end{cases} \tag{4.12}$$

$$e_{j,n,m}^{WI} = \begin{cases} (1 - d_{j,1}) e_{j,m}^I d_{j,m}^I, & \text{for } m \in \{1, ..., N_j^I + 1\}, n = m, m+1 \\ \left(1 - d_{j,n-m}^{WI}\right) e_{j,n-1,m}^{WI}, & \text{for } m \in \{1, ..., N_j^I + 1\}, n \in \{m+2, ..., m + N_j^{WI} + 1\} \\ 0, & \text{otherwise} \end{cases} \tag{4.13}$$

We combine the probabilities $e^{WS}_{j,n}$ and $e^{WI}_{j,m,n}$ to calculate the probability that a patient of surgery group j is in the ward on day n, see Equation (4.14). We do this for all $m \leq n$. For $n = 1$ we just have $e^{WS}_{j,n}$, because we assume the minimum stay at the ICU is one day. For $n \in \{2, ..., N^W_j\}$ we have the probability that the patient from surgery group j is transferred to the ward and the probability that first the patient is transferred to the ICU. This second probability is the sum over all probabilities that the patient is in the ward on day n after a ward stay of m days, for all possible values of $m \leq n$.

$$e^W_{j,n} = \begin{cases} e^{WS}_{j,n}, & \text{for } n = 1 \\ e^{WS}_{j,n} + \sum_{m=1}^{n} e^{WI}_{j,n,m}, & \text{for } n \in \{2, ..., N^W_j\}, \\ 0, & \text{otherwise} \end{cases} \tag{4.14}$$

In this research, we assume that patients from the same surgery group do not always go to the same ward/ICU. Therefore, we need to determine the probability that patients from certain surgery groups are on specific wards and ICUs on day n. Thus, we need the probability that a patient from surgery group j is in ICU i, given they are in the ICU. $\sum_{i \in I} a_{i,j}$ is the probability a patient of surgery group j is in the ICU. So, for all $j \in \mathcal{J}$ and $i \in \mathcal{I}$ we have the probability:

$$\hat{a}_{i,j} = \frac{a_{i,j}}{\sum_{i \in I} a_{i,j}},$$

that a patient of surgery group j is in ICU i, given they are in the ICU. For the ward, this probability is given by $b_{w,j}$. We do not need to normalise this probability, because every patient in our model is transferred to the ward. Patients who do not stay at the ward, are modelled by a ward stay of zero days. $f^I_{i,j,n}$ and $f^W_{w,j,n}$ are the probability distributions of the number of patients from surgery group j in ICU i or ward w on day n. $\mathrm{f}^I_{i,j,n}$ and $\mathrm{f}^W_{w,j,n}$ are the discrete stochastic variables that are associated to the probability distributions $f^I_{i,j,n}$ and $f^W_{w,j,n}$ respectively. The probability that there is one patient on the ward or ICU is calculated by multiplying the probability that a patient from surgery group j goes to a certain ICU i (ward w), given they are on the ICU (ward), with the probability they are on the ICU (ward) on day n, see Equation (4.16) (Equation (4.18)). The probability that there are zero patients is equal to one minus the probability that there is one patient.

$$P(\mathrm{f}^I_{i,j,n} = 0) = 1 - \hat{a}_{i,j} e^I_{j,n}, \qquad i \in \mathcal{I}, j \in \mathcal{J}, n \in \{1, ..., N^I_j\} \tag{4.15}$$

$$P(\mathrm{f}^I_{i,j,n} = 1) = \hat{a}_{i,j} e^I_{j,n}, \qquad i \in \mathcal{I}, j \in \mathcal{J}, n \in \{1, ..., N^I_j\} \tag{4.16}$$

$$P(\mathrm{f}^W_{w,j,n} = 0) = 1 - b_{w,j} e^W_{j,n}, \qquad w \in \mathcal{W}, j \in \mathcal{J}, n \in \{1, ..., N^W_j\} \tag{4.17}$$

$$P(\mathrm{f}^W_{w,j,n} = 1) = b_{w,j} e^W_{j,n}, \qquad w \in \mathcal{W}, j \in \mathcal{J}, n \in \{1, ..., N^W_j\} \tag{4.18}$$

Notation	Description
$a_{i,j}$	Probability that a patient of surgery group j is transferred to ICU i after surgery
$b_{w,j}$	Probability that a patient of surgery group j is transferred to ward w after surgery or ICU
$c^{I}_{j,n}$	Probability that a patient of surgery group j stays n days in ICU
$c^{WS}_{j,n}$	Probability that a patient of surgery group j stays n days in ward after surgery
$c^{WI}_{j,n}$	Probability that a patient of surgery group j stays n days in ICU after surgery
$d^{I}_{j,n}$	Probability that a patient of surgery group j transferred to the ward on day n
$d^{WS}_{j,n}$	Probability that a patient of surgery group j discharged from the ward on day n
$d^{WI}_{j,n}$	Probability that a patient of surgery group j discharged from the ward n days after transfer from ICU
$e^{I}_{j,n}$	Probability that a patient of surgery group j who had surgery on day one is in ICU i on day n
$e^{WS}_{j,n}$	Probability that a patient of surgery group j who had surgery on day one and no ICU is in the ward on day n
$e^{WI}_{j,m,n}$	Probability that a patient of surgery group j who had surgery on day one and left the ICU on day m is in the ward on day n
$e^{W}_{j,n}$	Probability that a patient of surgery group j who had surgery on day one is in the ward on day n
$f^{I}_{i,j,n}$	Probability distribution of the number of patients of surgery group j who had surgery on day one in ICU i on day n
$f^{W}_{w,j,n}$	Probability distribution of the number of patients of surgery group j who had surgery on day one in ward w on day n

Table 4.6: Parameters.

4.4.2. Cyclical surgery group

In this step, we account for the fact that the schedule is cyclical. This means that the scheduled surgery groups are repeated every cycle. The maximum LoS of a patient can exceed the cycle time. Therefore, patients who had surgery in different cycles might be recovering at the same time. The number of overlapping cycles is dependent on the maximum number of days patients from surgery groups recover in the ICU N^I_j and the ward N^W_j. Furthermore, it depends on the cycle length L, which is the number of elements in $\mathcal{L}$. Depending on the day l in the cycle we have $\lfloor (N^I_j - l)/L \rfloor + 1$ the number of overlapping cycles for the ICU and $\lfloor (N^W_j - l)/L \rfloor + 1$ the number of overlapping cycles for the ward.

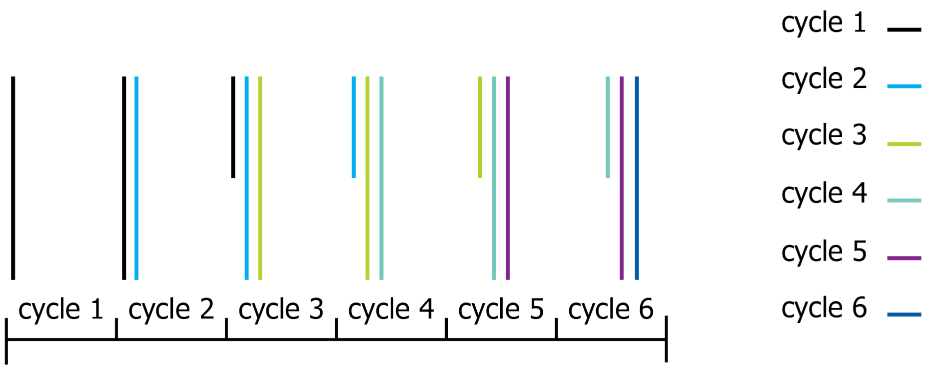

Figure 4.7: Six cycles shown for one surgery group. Patients stay a maximum of 2.5 times the cycle length.

In Figure 4.7, we see an example of a patient distribution from one surgery group throughout multiple cycles. Here, the maximum number of days a patient stays on the ward is two and a half times the cycle length. Therefore, a block only influences the next two cycles. From cycle three onward we see that the cycles all have the same distribution, caused not only by the patient from the surgery group in the current cycle, but also by patients from surgery groups from previous cycles. Therefore, we sum the patient distributions of all overlapping cycles. Because these random variables are independent the resulting probability distribution is the convolution of their individual distributions. We perform discrete convolutions on the patient distribution of all overlapping cycles, specified with symbol $*$, to obtain the distribution of patients on the days of one cycle. Suppose S equals the sum of two independent discrete random variables $S = X + Y$, then the distribution of S is given by the discrete convolution of the probability distributions of X and Y. Discrete convolution is defined as:

$$P(S = s) = \sum_{t=-\infty}^{\infty} P(X = t)P(Y = s - t).$$

So for each possible outcome s we determine the probability that it occurs. We do this by summing all probabilities where s is an outcome. The outcome is equal to s when we look at the probability that X equals k and Y equals $s - t$. We do this for all possible t.

Equations 4.19 and 4.20 give the distributions $F^I_{i,j,l}$ and $F^W_{j,l,w}$, of the number of recovering patients from surgery group j in ICU i and ward w, respectively, on the lth day of a cycle, when the cyclical surgery group is scheduled on day one of the cycle.

$$F^I_{i,j,l} = f^I_{i,j,l} * f^I_{i,j,l+L} * f^I_{i,j,l+2L} * \cdots * f^I_{i,j,l+\lfloor (N^W_j - l)/L \rfloor L}, \qquad i \in I, j \in \mathcal{J}, l \in \mathcal{L} \tag{4.19}$$

$$F^W_{w,j,l} = f^W_{w,j,l} * f^W_{w,j,l+L} * f^W_{w,j,l+2L} * \cdots * f^W_{w,j,l+\lfloor (N^W_j - l)/L \rfloor L}, \qquad w \in \mathcal{W}, j \in \mathcal{J}, l \in \mathcal{L} \tag{4.20}$$

This is the last step that can be done as preprocessing. Before generating a cyclic schedule, we can already determine the patient distribution over the days l in a cycle caused by one cyclical surgery group. For the next step, we require a given cyclic schedule.

4.4.3. Cyclic schedule

In this last step, we calculate the patient distributions on the wards and ICUs from a given cyclic schedule. The OR department does not handle elective cases during the weekends, so surgery groups can only be scheduled on weekdays. For a given cyclic schedule, we get a set of integer decision variables z. Each $z_{o,k,j}$ is the number of times surgery group j is assigned to OR o on day k. In this step we assume the cyclic schedule is given so these values are known. Let $\mathbf{1}_{z_{o,k,j}}$ be an indicator function, that is equal to one if $z_{o,k,j}$ is greater than zero and equal to zero if $z_{o,k,j}$ is zero. The patient distribution in ICU i and ward w on day l of the cyclic schedule coming from scheduling surgery group j once on OR-day (o,k) is given by $G^I_{i,o,k,j,l}$ and $G^W_{w,o,k,z,l}$ respectively, in Equations 4.21 and 4.22.

$$G^I_{i,o,k,j,l} = \begin{cases} F^I_{i,j,l-k+1}\mathbf{1}_{z_{o,k,j}}, & l \geq k \\ F^I_{i,j,l-k+1+L}\mathbf{1}_{z_{o,k,j}}, & \text{otherwise.} \end{cases} \qquad i \in \mathcal{I}, o \in \mathcal{O}, k \in \mathcal{K}, l \in \mathcal{L}, j \in \mathcal{J}_i \tag{4.21}$$

$$G^W_{w,o,k,j,l} = \begin{cases} F^W_{w,j,l-k+1}\mathbf{1}_{z_{o,k,j}}, & l \geq k \\ F^W_{w,j,l-k+1+L}\mathbf{1}_{z_{o,k,j}}, & \text{otherwise.} \end{cases} \qquad w \in \mathcal{W}, o \in \mathcal{O}, k \in \mathcal{K}, l \in \mathcal{L}, j \in \mathcal{J}_w \tag{4.22}$$

We make a distinction between days after day k on which a surgery group is assigned in the cycle and days that come before the day k on which a surgery group is assigned in the cycle. $F^I_{i,j,l}$ and $F^W_{w,j,l}$ are the distributions of the number of recovering patients when the cyclical surgery group was scheduled on day one of the cycle. Therefore, we need to shift these distributions to the day they actually take place. Here l is the day for which we are determining the patient distribution and k is the day the surgery group, of which we are determining the patient distribution, is scheduled in the cyclic schedule. If $l \geq k$, the number of days between day l and the day of the surgery k is $l-k$. For the patient distribution of surgery group j in a cyclic schedule, $F^I_{i,j,l}$, we assume that the surgery took place on day 1. To find the patient distribution on day l when the surgery group was scheduled on day k, we shift by $k-1$ days. If $l < k$, then the patient distribution on day l results only from surgery groups scheduled on day k of previous cycles. This means that the number of days between l and k is then $l-k+L$. Thus, we shift by $l-k+1+L$. We multiply these distributions by $\mathbf{1}_{z_{o,k,j}}$, which is only non-zero if the surgery group j is assigned to OR-day (o,k). We only need to sum over the surgery groups that can go to ward w or ICU i.
Next, we obtain the patient distribution in ICU i and ward w on day l of the cyclic schedule for a whole OR-day. In 4.21 and 4.22, we use the indicator function $\mathbf{1}_{z_{o,k,j}}$ to determine if a surgery group j is assigned to OR-day (o,k). However, a surgery group might be assigned multiple times to one OR-day. To obtain the distribution of patients from a whole OR-day, we need to convolve all distributions $G^I_{i,o,k,j,l}$ of the surgery groups scheduled on that OR-day. If a surgery group is assigned n times to one OR-day, then, we need to convolve the distribution n times with itself, before convolving it with the distributions of the other surgery groups assigned to that OR-day. Therefore, we use the convolution power, which is defined as the n-fold iteration of the convolution with itself. For h a function $\mathbb{Z} \to \mathbb{R}$ and $n \in \mathbb{N}_{>0}$ we have:

$$h^{*n} = \underbrace{h * h * \ldots * h * h}_{n}, \qquad h^{*0} = \delta_0 \tag{4.23}$$

where δ_0 is the delta dirac function. The delta dirac function focusses the mass of a function around zero, which in practice means that if we convolve a distribution zero times, the probability of being zero is equal to one. For each given cyclic schedule, the integer decision variables $z_{o,k,j}$ specify the number of times surgery group j is assigned to OR-day (o,k). The patient distribution in ICU i and ward w on day l of the cyclic schedule resulting from all surgeries from surgery group j scheduled on OR-day (o,k) is given by $\hat{G}^I_{i,o,k,j,l}$ and $\hat{G}^W_{w,o,k,j,l}$ in Equations 4.24 and 4.25 respectively.

$$\hat{G}^I_{i,o,k,j,l} = {G^I_{i,o,k,j,l}}^{*z_{o,k,j}}, \qquad i \in \mathcal{I}, o \in \mathcal{O}, k \in \mathcal{K}, j \in \mathcal{J}_i \tag{4.24}$$

$$\hat{G}^W_{w,o,k,j,l} = {G^I_{w,o,k,j,l}}^{*z_{o,k,j}}, \qquad w \in \mathcal{W}, o \in \mathcal{O}, k \in \mathcal{K}, j \in \mathcal{J}_w \tag{4.25}$$

where $z_{o,k,j}$ denotes the number of times surgery group j is scheduled on OR-day (o,k). We convolve the distribution as many times as it is assigned to the OR-day, to obtain the patient distribution on day l in the ICUs and wards coming from scheduling this surgery group in the cyclic schedule.

From these distributions, we can define the distributions $H^I_{i,o,k,l}$ and $H^W_{w,o,k,l}$, which represent the patient distributions on day l on ICU i and ward w, resulting from all surgeries on OR-day (o,k) (see Equations 4.26 and 4.27). We obtain these distributions by convolving all distributions for the surgery groups that can be scheduled on OR-day (o,k) and have a chance of going to ICU i or ward w after surgery, they are denoted by $j_1, j_2, ..., j_{\max} \in \mathcal{J}_i$.

$$H^I_{i,o,k,l} = \hat{G}^I_{i,o,k,j_1,l} * \hat{G}^I_{i,o,k,j_2,l} * ... * \hat{G}^I_{i,o,k,j_{\max},l}, \qquad i \in \mathcal{I}, o \in \mathcal{O}, k \in \mathcal{K}, j \in \mathcal{J}_i \tag{4.26}$$

$$H^W_{w,o,k,l} = \hat{G}^W_{w,o,k,j_1,l} * \hat{G}^W_{w,o,k,j_2,l} * ... * \hat{G}^W_{w,o,k,j_{\max},l}, \qquad w \in \mathcal{W}, o \in \mathcal{O}, k \in \mathcal{K}, j \in \mathcal{J}_w \tag{4.27}$$

To obtain the patient distributions resulting from the whole cyclic schedule, we need to convolve the distributions of all the days in the cyclic schedule (see Equations 4.28 and 4.29). $\hat{H}^I_{i,l}$ denotes the distribution of recovering patients in ICU i on day l of the cyclic schedule. $\hat{H}^W_{w,l}$ denotes the distribution of recovering patients in ward w on day l of the cyclic schedule. The last OR and the last day in the cyclic schedule on which surgeries take place are denoted by $\max\{\mathcal{O}\}$ and $\max\{\mathcal{K}\}$ respectively.

$$\hat{H}^I_{i,l} = H^I_{i,1,1,l} * H^I_{i,1,2,l} * ... * H^I_{i,1,\max\{\mathcal{K}\},l} * H^I_{i,2,1,l} * H^I_{i,2,2,l} * ... * H^I_{i,\max\{\mathcal{O}\},\max\{\mathcal{K}\},l} \qquad i \in \mathcal{I}, l \in \mathcal{L} \tag{4.28}$$

$$\hat{H}^W_{w,l} = H^W_{w,1,1,l} * H^W_{w,1,2,l} * ... * H^W_{w,1,\max\{\mathcal{K}\},l} * H^W_{w,2,1,l} * H^W_{w,2,2,l} * ... * H^W_{w,\max\{\mathcal{O}\},\max\{\mathcal{K}\},l} \qquad w \in \mathcal{W}, l \in \mathcal{L} \tag{4.29}$$

The functions $\hat{H}^I_{i,l}$ and $\hat{H}^W_{w,l}$ are used to model the probability distribution on day l of the number of patients in ICU i and ward w respectively. We define the probability of having n patients in ICU i or ward w on day l by $\hat{H}^I_{i,l}[n]$ and $\hat{H}^W_{w,l}[n]$. For a given solution for the cyclic schedule ψ, we want to minimise the variation in bed occupancy during the week. Therefore, we find the required number of beds for each day of the cyclic schedule. We calculate the probability that there are at most n patients, thus n required beds, by summing over the probabilities that there are less or equal than n patients in the ICU or ward. For each day l, we find the minimum number of patients n_l, for which the probability of there being n_l or less patients is greater or equal to the parameter p. So, for each day l, we can say with probability p that there are at most n_l patients. The required number of beds on day l on ICU i, $\gamma_{i,l}(\psi)$, for a given solution $\psi \in \Psi$ is given by:

$$\gamma_{i,l}(\psi) = \min\left\{n \,\middle|\, \sum_{m=0}^{n} \hat{H}^I_{i,l}[m] \geq p\right\}. \tag{4.30}$$

The required number of beds on day l on ward w for a given solution ψ is given by

$$\gamma_{w,l}(\psi) = \min\left\{n \,\middle|\, \sum_{m=0}^{n} \hat{H}^W_{w,l}[m] \geq p\right\}. \tag{4.31}$$

The variation in bed occupancy, denoted by $\gamma_i(\psi)$ and $\gamma_w(\psi)$, on ICU i and ward w is given by the difference between the maximum and minimum number of required beds during the week. Peaks in bed occupancy during weekdays can cause cancellation of scheduled surgical patients, because not enough beds are available. Therefore, we are interested in minimising the variation in bed occupancy during weekdays, because in weekends the bed occupancy is much lower than during the week due to no new patients being admitted. The variation in bed occupancy at ICU i and ward w are given by

$$\gamma_i(\psi) = \max_{l \in \mathcal{K}} \gamma_{i,l}(\psi) - \min_{l \in \mathcal{K}} \gamma_{i,l}(\psi) \tag{4.32}$$

$$\gamma_w(\psi) = \max_{l \in \mathcal{K}} \gamma_{w,l}(\psi) - \min_{l \in \mathcal{K}} \gamma_{w,l}(\psi) \tag{4.33}$$

where $\mathcal{K}$ is the set of all workdays as defined in Section 4.3.

Notation	Description
$F^{I}_{i,j,l}$	Distribution of number of recovering patients from surgery group j in ICU i on the lth day of a cycle, when the cyclical surgery group was scheduled on day one of all cycles.
$F^{W}_{w,j,l}$	Distribution of number of recovering patients from surgery group j in ward w on the lth day of a cycle, when the cyclical surgery group was scheduled on day one of all cycles.
$G^{I}_{i,o,k,j,l}$	Distribution of number of recovering patients in ICU i on day l of one surgery from surgery group j scheduled in OR o on day k
$G^{W}_{w,o,k,j,l}$	Distribution of number of recovering patients in ward w on day l of one surgery from surgery group j scheduled in OR o on day k
$\hat{G}^{I}_{i,o,k,j,l}$	Distribution of the number of recovering patients in ICU i on day l resulting from all surgeries of surgery group j on day k in OR o
$\hat{G}^{W}_{w,o,k,l}$	Distribution of the number of recovering patients in ward w on day l resulting from all surgeries of surgery group j on day k in OR o
$H^{I}_{i,o,k,l}$	Distribution of the number of recovering patients in ICU i on day l resulting from all surgeries on day k in OR o
$H^{W}_{w,o,k,l}$	Distribution of the number of recovering patients in ward w on day l resulting from all surgeries on day k in OR o
$\hat{H}^{I}_{i,l}$	Distribution of recovering patients in ICU i on day l of the cyclic schedule
$\hat{H}^{W}_{w,l}$	Distribution of recovering patients in ward w on day l of the cyclic schedule
$\gamma_i(\psi)$	Number of required beds on ICU i for given solution ψ
$\gamma_w(\psi)$	Number of required beds on ward w for given solution ψ

Table 4.7: Parameters.

4.4.4. Objective function

In accordance with our research goals, our model has two main goals. The first goal is to maximise the OR utilisation and the second goal is to minimise the variation in bed occupancy. Because the available OR time is determined beforehand, it is constant. Therefore, maximising the OR utilisation is equal to maximising the time needed for scheduled surgery groups. The utilised OR time is the sum of all session times (surgery durations), as defined in Section 2.1.1, of scheduled surgery groups. As in the research of Van Oostrum *et al.* [2], we assume that surgery durations are normally distributed with mean μ_j and variation σ_j^2. Furthermore, we want to minimise the variation in bed occupancy, γ_i

and γ_w, at the ICUs and wards. We include the weights θ_i and θ_w, so we can balance between the variation in bed occupancy and the OR utilisation.
The objective function is given by

$$\max \sum_{o \in O} \sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{J}} \mu_j \cdot z_{o,k,j} - \sum_{i \in I} \theta_i \cdot \gamma_i(\psi) + \sum_{w \in W} \theta_w \cdot \gamma_w(\psi), \tag{4.34}$$

where the objective function value for a given schedule ψ is denoted by $OB(\psi)$. The objective function, subject to constraints 4.1-4.4, form our basic model.

The calculations in Sections 4.4.1 and 4.4.2 can be performed beforehand. However, the calculations in Section 4.4.3 still involve the convolution of several probability distributions. Therefore, calculating the objective function takes a lot of computation time and it is hard to predict the effect in the objective function for different OR schedules. Furthermore, the minimum and maximum operators in Equations 4.32 and 4.33 are not linear operators. Moreover, constraint 4.4 is a nonlinear constraint so the resulting model is a nonlinear model. To reduce the computation time, we can either (1) approximate the objective function and the nonlinear constraint and use these approximations in an MILP or (2) we use a local search heuristic based on the given constraints and objective function. The approaches are discussed in more detail in the following chapters.

5
Global approach

In our model, we have a complex objective function and a nonlinear constraint. Part of the objective function minimises the number of required beds on the wards and ICU. It is hard to predict the effect in this part of the objective function for different OR schedules. We use two different approaches to solve the model described in the previous chapter. We can either solve the original problem with a heuristic procedure or use a global approach to a simplified version of the problem. We want to investigate which of these two methods leads to better solutions, because there is no guarantee that either finds the optimal solution. In this chapter, we simplify the original problem so we can use a global approach. This approach uses an approximation of the objective function and a linearised version of the nonlinear constraint 4.4 and searches the entire solution space.

In Section 5.1, we linearise the overtime constraint 4.4. Because there is no direct relation between a given OR-schedule and the number of required beds, we also linearise the objective function in Section 5.2. This results in an mixed integer linear programme (MILP), which is proven to be NP-hard in Section 5.3.

5.1. Linearising the surgery duration constraint

In the problem formulation introduced in Section 4.3, we have a nonlinear constraint. This constraint makes the surgery schedule more robust against overtime. It ensures that in addition to the planned surgeries we also assign planned slack to each OR-day. To use a MILP solver we need to linearise the overtime constraint. Our approach is based on Bosch [31]. The overtime constraint is given by

$$P(g_{o,k} \leq \tau_{o,k}) \geq 1 - \alpha, \qquad \forall o \in \mathcal{O}, k \in \mathcal{K} \tag{5.1}$$

where $g_{o,k}$ is the probability distribution of the total session time of all surgery types scheduled on OR day (o,k) and $\tau_{o,k}$ is the total available time to perform surgeries on OR-day (o,k).

Stepaniak *et al.* [32], compare the fit of the normal distribution with that of the 2- and 3- parameter log-normal distributions for surgery durations. They conclude that the 3-parameter log-normal distribution provides the best results. However, Van Oostrum *et al.* [2] and Hans *et al.* [33] assume that surgery durations are normally distributed. Assuming a normal distribution for surgery durations has the advantage that the sum of surgery durations is also normally distributed. This means that the probability of overtime on an OR-day is easy to calculate. When we assume the 3-parameter log-normal distribution for surgery durations, there is no exact expression for the sum of surgery durations. Therefore, we assume that surgery durations are normally distributed. The total duration of OR-day (o,k) is normally distributed with mean $\mu_{o,k}$ and variance $\sigma_{o,k}^2$. Thus, $g_{o,k}(x) \sim \mathcal{N}(\mu_{o,k}, \sigma_{o,k})$. Rewriting the probability that no overtime occurs to the standard normal form gives

$$P(g_{o,k} \leq \tau_{o,k}) = \Phi\left(\frac{\tau_{o,k} - \mu_{o,k}}{\sigma_{o,k}}\right). \tag{5.2}$$

The overtime constraint can then be written as

$$\Phi\left(\frac{\tau_{o,k}-\mu_{o,k}}{\sigma_{o,k}}\right) \geq 1-\alpha. \tag{5.3}$$

Rewriting Equation (5.3) gives us a new formulation of the overtime constraint

$$\mu_{o,k} + \Phi^{-1}(1-\alpha)\sigma_{o,k} \leq \tau_{o,k}. \tag{5.4}$$

We assume the surgery duration of surgery group j is normally distributed with mean μ_j and variance σ_j^2 for all surgery groups j. The number of times a surgery from surgery group j is scheduled on day k in OR o is given by $z_{o,k,j}$. Therefore, the mean and variance of the total surgery duration $g_{o,k}$ on OR-day (o,k) can be written as

$$\mu_{o,k} = \sum_{j\in\mathcal{J}} z_{o,k,j}\mu_j \quad \text{and} \quad \sigma_{o,k}^2 = \sum_{j\in\mathcal{J}} z_{o,k,j}\sigma_j^2. \tag{5.5}$$

Substituting this into the overtime constraint 5.4 gives

$$\sum_{j\in\mathcal{J}} z_{o,k,j}\mu_j + \Phi^{-1}(1-\alpha)\sqrt{\sum_{j\in\mathcal{J}} z_{o,k,j}\sigma_j^2} \leq \tau_{o,k}, \qquad \forall o\in\mathcal{O}, k\in\mathcal{K}. \tag{5.6}$$

However, this constraint is still nonlinear in decision variable $z_{o,k,j}$ due to the square root. Therefore, we approximate the square root function $f(x)=\sqrt{x}$ by a piecewise linear function to reduce the nonlinear problem to a linear one. We only look at the optimisation in a given interval $[x_{\min}, x_{\max}]$. We do not want to underestimate the function $f(x)$, therefore, the approximation function must be greater or equal to $f(x)$ in all $x\in[x_{\min}, x_{\max}]$. The intervals of the piecewise linear functions are determined by breakpoints $n\in N$, where $N=\{0,1,...,m\}$. Here, x_n is the value on the x-axis of breakpoint $n\in N$. We define x_0 as the first x-value and x_m as the last x-value, for which we approximate the square root function. The other x-values, x_n for $n=\{1,...,m-1\}$, are intersection point of the linear approximations. In each interval we define the points t_n. Each linear function is given by the tangent line of the square root function in the point t_n for $n\in N\backslash\{0\}$. We describe the linear approximation functions for each interval $n\in N\backslash\{0\}$ by $h_n(x)=a_n+b_nx$ where b_n is the derivative of the square root function in point t_n.

$$b_n = \left(\sqrt{t_n}\right)' = \frac{1}{2\sqrt{t_n}} \tag{5.7}$$

We determine a_n by setting the linear approximation function equal to the square root function in the point t_n.

$$\begin{aligned}\sqrt{t_n} &= h_n(t_n)\\ &= a_n + b_n t_n\end{aligned}$$

Substituting the definition of b_n into this equation gives us

$$\begin{aligned}a_n &= \sqrt{t_n} - \frac{1}{2\sqrt{t_n}}t_n\\ &= \sqrt{t_n} - \frac{1}{2}\sqrt{t_n}\\ &= \frac{1}{2}\sqrt{t_n}.\end{aligned} \tag{5.8}$$

So for each interval n, the linear approximation function on that interval is given by

$$h_n(x) = \frac{1}{2}\sqrt{t_n} + \frac{1}{2\sqrt{t_n}}x \tag{5.9}$$

Let y_n be the function value of the linear approximation function at breakpoint n, so $y_n = h_n(x_n)$. How to determine the breakpoints is explained in Appendix A. Once the breakpoints are known we can use the λ-formulation, a common method to model piecewise linear functions together as described in Bisschop [34]. The function value of any point between two breakpoints is the weighted sum of the function values of these two breakpoints. Let λ_n denote n nonnegative weights such that their sum equals one. The piecewise linear approximation of the square root function can be written as:

$$\sum_{n=0}^{N} \lambda_n h_n(x_n) = h_n(x) \tag{5.10}$$

$$\sum_{n=0}^{N} \lambda_n x_n = x \tag{5.11}$$

$$\sum_{n=0}^{N} \lambda_n = 1 \tag{5.12}$$

The last constraint and a requirement for the set of λ's, form a special ordered set type two (SOS2) constraint, see Bisschop [34]. The SOS2 constraint has the common restriction that out of a set of nonnegative variables, at most two variables can be nonzero. Furthermore, in a fixed order list the two variables that are nonnegative must be adjacent to each other.
The overtime constraint is given for every OR-day (o, k). Therefore, we need a set of λ_n's for every OR-day. The overtime constraint, with additional constraints, can now be written as:

$$\sum_{j \in \mathcal{J}} z_{o,k,j}\mu_j + \Phi^{-1}(1-\alpha) \sum_{n \in N} \lambda_{o,k,n} y_n \leq \tau_{o,k} \qquad \forall o \in \mathcal{O}, \forall k \in \mathcal{K} \tag{5.13}$$

$$\sum_{n \in N} \lambda_{o,k,n} x_n = \sum_{j \in \mathcal{J}} z_{o,k,j}\sigma_j^2 \qquad \forall o \in \mathcal{O}, \forall k \in \mathcal{K} \tag{5.14}$$

$$\sum_{n \in N} \lambda_{o,k,n} = 1 \qquad \forall o \in \mathcal{O}, \forall k \in \mathcal{K} \tag{5.15}$$

where y_n is equal to $\sqrt{x_n}$.

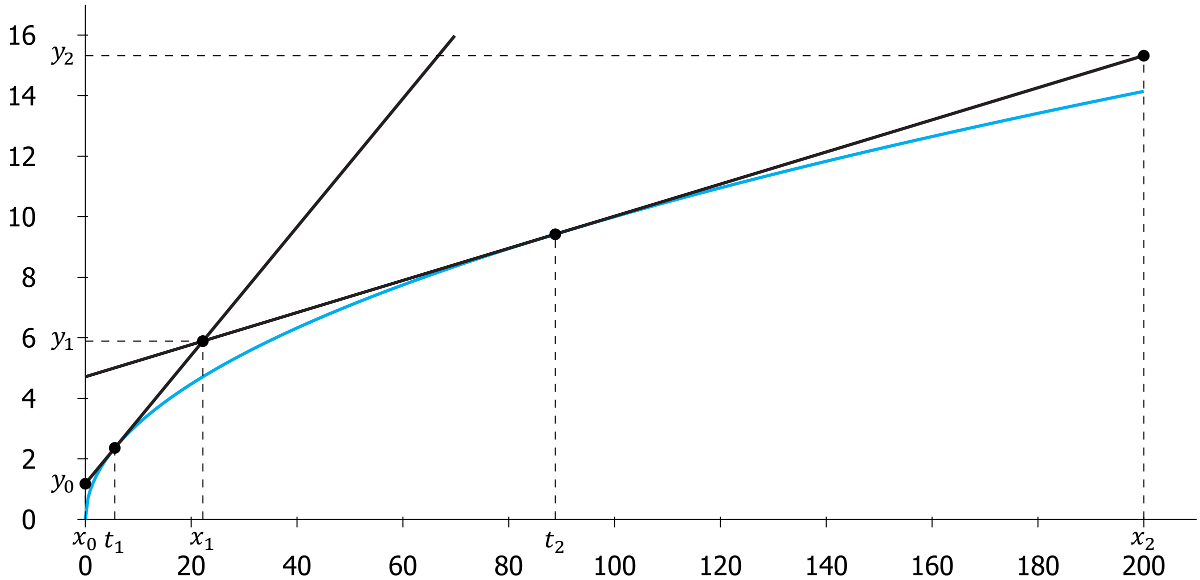

Figure 5.1: Linear approximation of the square root function on the interval $[0, 200]$ with two approximation functions.

Following the approach explained in Appendix A, we can determine the breakpoints. In Figure 5.1, an example with two linear approximation functions on the domain $[0, 200]$ is shown. To determine the breakpoints we need the maximum interval and the maximum allowed approximation error. The maximum interval on which we want to calculate the square root function is determined by calculating the largest standard deviation possible on one OR-day. The maximum approximation error is set to five minutes. This means we will overestimate the standard deviation by at most five minutes. To obtain this maximum approximation error on the calculated interval we need six linear approximation functions. However, at breakpoint x_0, the breakpoint value y_0 is greater than zero, because all linear approximation functions overestimate the square root function in the breakpoints. Suppose there is an OR-day, $(\bar{o},\bar{k})$, where an OR is closed and therefore the available time, $\tau_{\bar{o},\bar{k}}$, equals zero. On these days, no surgeries are scheduled. However, for α less than 0.5, the overtime constraint will be violated.

$$\begin{aligned}
&\sum_{j\in\mathcal{J}} z_{\bar{o},\bar{k},j}\mu_j + \Phi^{-1}(1-\alpha)\sum_{n\in N}\lambda_{\bar{o},\bar{k},n}y_n \\
&= \Phi^{-1}(1-\alpha)\lambda_{\bar{o},\bar{k},0}y_0 \\
&= \Phi^{-1}(1-\alpha)\cdot y_0 \\
&\geq \tau_{\bar{o},\bar{k}} = 0
\end{aligned}$$

To solve this problem we create an extra breakpoint between breakpoints x_0 and x_1, denoted by x_+. This breakpoint x_+, will be very close to zero. We then give the corresponding breakpoint value y_+ the old value of y_0 and let $y_0 = 0$. Now, the overtime constraint will not be violated when scheduling no surgeries on an OR-day. Moreover, it does not influence the other approximations. The smallest standard deviation of all surgery groups is larger than x_1, and therefore, we never use the approximation functions between x_0 and x_1.

5.2. Linearisation of the objective function

In this section, we linearise the objective function. Our approach is based on the approach of Beliën and Demeulemeester [9], but applied to a more extended model. Instead of using γ_i and γ_w, the required number of beds in the ICU and ward we work with the expected values $\bar{\gamma}_{w,l}(\psi)$ and $\bar{\gamma}_{i,l}(\psi)$, the expected number of required beds at ward w and ICU i, on day l of the cycle, for solution ψ. We first show that these variables are linear in the integer decision variable $z_{o,k,j}$. We use the expected value of the distribution functions $\hat{H}^I_{i,l}$ and $\hat{H}^W_{w,l}$. They are defined as the probability distributions of the number of recovering patients in the ICU and ward respectively.

Let $\hat{\mathcal{H}}^I_{i,l}$ and $\hat{\mathcal{H}}^W_{w,l}$ be the discrete random variables that are associated to the probability distributions $\hat{H}^I_{i,l}$ and $\hat{H}^W_{w,l}$. The expected number of required beds at ICU i on day l is equal to the expected value of $\hat{\mathcal{H}}^I_{i,l}$, the discrete random variable corresponding with the probability distribution of recovering patients in ICU i on day l of the cyclic schedule, see Equation 5.16. This random variable is the sum of multiple discrete random variables $\hat{\mathcal{H}}^I_{i,o,k,l}$, the random variables corresponding to the probability distributions described in Equation (4.28). Suppose random variable S is the sum of two random variables $S = X + Y$ then $\mathbb{E}(S) = \mathbb{E}(X+Y) = \mathbb{E}(X) + \mathbb{E}(Y)$. The random variable $\hat{\mathcal{H}}^I_{i,l}$ is the sum of the random variables $\mathcal{H}^I_{i,o,k,l}$ that are associated to the probability distributions $H^I_{i,o,k,l}$. The expected value of sums is equal to the sum of expected values, which gives Equation (5.18). This is the sum over all OR-days (o,k) of the expected number of patients recovering in ICU i on day l who had surgery on OR-day (o,k). The expected number of patients in ICU i on day l from OR-day (o,k) is equal to the sum over all surgery groups $j \in \mathcal{J}_i$ of the expected number of patients in ICU i on day l resulting from surgeries of surgery group j OR-day (o,k), see Equation (5.20). The expected number of patients from surgery group j on OR-day (o,k) is equal to the expected value resulting from scheduling surgery group j once on OR-day (o,k) times $z_{o,k,j}$, the number of times surgery group j is scheduled on OR-day (o,k), see Equation (5.21). In Equation (4.21), we define the distribution function $G^I_{i,o,k,j,l}$. The definition depends on l in relation to k. Therefore, the expected value of the random variable corresponding to the distribution function $F^I_{i,j,n}$ is split into two parts, one for $l \geq k$ and one for $l < k$. This gives Equation (5.21), where $z_{o,k,j}$ equals zero if group j is not scheduled on OR-day (o,k). The expected value of $\mathcal{F}^I_{i,j,n}$ is the expected number of patients on day n in ICU i resulting from scheduling surgery group j on day one. Because we want to know the expected number of patients on day l we shift with the number of days between day l and the day of the surgery day k. For $l \geq k$ this is equal to $l-k+1$, for $l < k$ this is equal to $l-k+1+L$. Again using that the expected value of the sum of random variables is equal to the sum of the expected values we obtain Equation (5.23). We sum from $n = 0$ to $n = \lfloor (N^I_j - (l-k+1))/L \rfloor + 1$, i.e., the number of overlapping cycles on day $l \in \mathcal{L}$ when $l \geq k$. When $l < k$ we sum from $n = 1$ to $n = \lfloor (N^I_j - (l-k+1+L))/L \rfloor + 1$ the number of overlapping cycles on day $l \in \mathcal{L}$. The expected value of $\mathrm{f}^I_{i,j,n}$ is given by the probability that there is a patient of group j in ICU i on day n, see Equation (4.15).

$$\bar{\gamma}_{i,l} = \mathbb{E}\left(\hat{\mathcal{H}}^I_{i,l}\right) \tag{5.16}$$

$$= \mathbb{E}\left(\sum_{o \in \mathcal{O}} \sum_{k \in \mathcal{K}} \mathcal{H}^I_{i,o,k,l}\right) \tag{5.17}$$

$$= \sum_{o \in \mathcal{O}} \sum_{k \in \mathcal{K}} \mathbb{E}\left(\mathcal{H}^I_{i,o,k,l}\right) \tag{5.18}$$

$$= \sum_{o \in \mathcal{O}} \sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{J}_i} \mathbb{E}\left(\hat{\mathcal{G}}^I_{i,o,k,j,l}\right) \tag{5.19}$$

$$= \sum_{o \in \mathcal{O}} \sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{J}_i} \mathbb{E}\left(\mathcal{G}^I_{i,o,k,j,l}\right) z_{o,k,j} \tag{5.20}$$

$$= \sum_{o \in \mathcal{O}} \sum_{\substack{k \in \mathcal{K} \\ l \geq k}} \sum_{j \in \mathcal{J}_i} \mathbb{E}\left(\mathcal{F}^I_{i,j,l-k+1}\right) z_{o,k,j} + \sum_{o \in \mathcal{O}} \sum_{\substack{k \in \mathcal{K} \\ l < k}} \sum_{j \in \mathcal{J}_i} \mathbb{E}\left(\mathcal{F}^I_{i,j,l-k+1+L}\right) z_{o,k,j} \tag{5.21}$$

$$\begin{aligned} &= \sum_{o \in \mathcal{O}} \sum_{\substack{k \in \mathcal{K} \\ l \geq k}} \sum_{j \in \mathcal{J}_i} \mathbb{E}\left(\mathrm{f}^I_{i,j,l-k+1} + \mathrm{f}^I_{i,j,l-k+1+L} + \dots + \mathrm{f}^I_{i,j,l-k+1+\lfloor (N^I_j-(l-k+1))/L \rfloor L}\right) z_{o,k,j} \\ &\quad + \sum_{o \in \mathcal{O}} \sum_{\substack{k \in \mathcal{K} \\ l < k}} \sum_{j \in \mathcal{J}_i} \mathbb{E}\left(\mathrm{f}^I_{i,j,l-k+1+L} + \mathrm{f}^I_{i,j,l-k+1+2L} + \dots + \mathrm{f}^I_{i,j,l-k+1+(\lfloor (N^I_j-(l-k+1+L))/L \rfloor+1)L}\right) z_{o,k,j} \end{aligned} \tag{5.22}$$

$$\begin{aligned} &= \sum_{o \in \mathcal{O}} \sum_{\substack{k \in \mathcal{K} \\ k \geq l}} \sum_{j \in \mathcal{J}_i} \sum_{n=0}^{\lfloor (N^I_j-(l-k+1))/L \rfloor} \mathbb{E}\left(\mathrm{f}^I_{i,j,l-k+1+nL}\right) z_{o,k,j} \\ &\quad + \sum_{o \in \mathcal{O}} \sum_{\substack{k \in \mathcal{K} \\ l < k}} \sum_{j \in \mathcal{J}_i} \sum_{n=1}^{\lfloor (N^I_j-(l-k+1+L))/L \rfloor+1} \mathbb{E}\left(\mathrm{f}^I_{i,j,l-k+1+nL}\right) z_{o,k,j} \end{aligned} \tag{5.23}$$

$$\begin{aligned} &= \sum_{o \in \mathcal{O}} \sum_{\substack{k \in \mathcal{K} \\ l \geq k}} \sum_{j \in \mathcal{J}_i} \sum_{n=0}^{\lfloor (N^I_j-(l-k+1))/L \rfloor} \hat{a}_{i,j} e^I_{j,l-k+1+nL} \cdot z_{o,k,j} \\ &\quad + \sum_{o \in \mathcal{O}} \sum_{\substack{k \in \mathcal{K} \\ l < k}} \sum_{j \in \mathcal{J}_i} \sum_{n=1}^{\lfloor (N^I_j-(l-k+1+L))/L \rfloor+1} \hat{a}_{i,j} e^I_{j,l-k+1+nL} \cdot z_{o,k,j} \end{aligned} \tag{5.24}$$

Similarly, we obtain

$$\bar{\gamma}_{w,l} = \mathbb{E}\left(\hat{H}^W_{w,l}\right) \tag{5.25}$$

$$\begin{aligned} \bar{\gamma}_{w,l} &= \sum_{o \in \mathcal{O}} \sum_{\substack{k \in \mathcal{K} \\ l \geq k}} \sum_{j \in \mathcal{J}_w} \sum_{n=0}^{\lfloor (N^W_j-(l-k+1))/L \rfloor} b_{w,j} e^W_{j,l-k+1+nL} \cdot z_{o,k,j} \\ &\quad + \sum_{o \in \mathcal{O}} \sum_{\substack{k \in \mathcal{K} \\ l < k}} \sum_{j \in \mathcal{J}_w} \sum_{n=1}^{\lfloor (N^W_j-(l-k+1+L))/L \rfloor+1} b_{w,j} e^W_{j,l-k+1+nL} \cdot z_{o,k,j} \end{aligned} \tag{5.26}$$

with $\lfloor (N_j^W - (l-k+1))/L \rfloor + 1$ the number of overlapping cycles on day $l \in \mathcal{L}$ when the surgery is scheduled on day k and $l \geq k$ and $\lfloor (N_j^W - (l-k+1+L))/L \rfloor + 1$ the number of overlapping cycles on day $l \in \mathcal{L}$ when $l < k$. The expected number of required beds on day l is given by the sum over all surgery groups of the probability that a patient from surgery group j is still in the hospital on day l, accounting for all cycles, multiplied with the number of times this surgery group is scheduled on all OR-days (o,k). Since $\sum_{n=1}^{\lfloor (N_j^I - l)/L \rfloor + 1} \hat{a}_{i,j} e^I_{j,l-k+1+nL}$ is constant, the new objective is linear in the decision variables. Again, we want the maximum and minimum of $\bar{\gamma}_{i,l}(\psi)$ and the maximum and minimum $\bar{\gamma}_{w,l}(\psi)$, because the variation in bed occupancy is given by the difference between the maximum and minimum number of beds needed during the week. The maximum and minimum operator are not linear. Therefore, we add constraints 5.30-5.33 and let

$$\hat{\gamma}_i = \bar{\gamma}_i^{\max} + \bar{\gamma}_i^{\min}, \qquad \forall i \in \mathcal{I}, \tag{5.27}$$

$$\hat{\gamma}_w = \bar{\gamma}_w^{\max} + \bar{\gamma}_w^{\min}, \qquad \forall w \in \mathcal{W}. \tag{5.28}$$

The resulting MILP model is given by

$$\max \sum_{o \in \mathcal{O}} \sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{J}} \mu_j \cdot z_{o,k,j} - \sum_{i \in \mathcal{I}} \theta_i \hat{\gamma}_i - \sum_{w \in \mathcal{W}} \theta_w \hat{\gamma}_w \tag{5.29}$$

s.t. (4.1) – (4.3), (5.13) – (5.15)

$$\bar{\gamma}_i^{\max} \geq \bar{\gamma}_{i,l} \qquad \forall i \in \mathcal{I}, l \in \mathcal{L} \tag{5.30}$$

$$\bar{\gamma}_w^{\max} \geq \bar{\gamma}_{w,l} \qquad \forall w \in \mathcal{W}, l \in \mathcal{L} \tag{5.31}$$

$$\bar{\gamma}_i^{\min} \geq -\bar{\gamma}_{i,l} \qquad \forall i \in \mathcal{I}, l \in \mathcal{L} \tag{5.32}$$

$$\bar{\gamma}_w^{\min} \geq -\bar{\gamma}_{w,l} \qquad \forall w \in \mathcal{W}, l \in \mathcal{L}. \tag{5.33}$$

We refere to this problem as the linear OR schedule problem.

5.3. Complexity of the problem

In this section, the computational complexity of the linear OR schedule problem, as described in Section 5.2, is discussed. In Section 5.3.1, we give an introduction to computational complexity. In Section 5.3.2, it is proven that the problem given in Section 5.2 is NP-hard.

5.3.1. Computational complexity

In this section, the theory of computational complexity is introduced based on Papadimitriou and Steiglitz [35] and Haemers [36]. With information given in this section, statements regarding the computational complexity of linear programming are made in Section 5.3.2.

In complexity theory, two different types of problems can be distinguished: decision problems and optimisation problems. A decision problem can be answered by 'yes' or 'no'. Given an optimisation problem, we can define a closely related decision problem. An example of an optimisation problem is the shortest path problem: *Given an undirected unweighted graph $G = (V,E)$ and two vertices $v,u \in V$, find the shortest path between v and u.* The related decision problem is given by the question: *Given an undirected unweighted graph $G = (V,E)$, two vertices $v,u \in V$ and a non-negative integer k, is there a path between u and v in G that has a length of at most k?* The decision problem is not harder to solve than the original optimisation problem. Therefore, any results proven about the complexity of the decision problem will also apply to the related optimisation problem.

The computation time or running time of an algorithm is defined by counting the number of elementary operations performed by the algorithm. The algorithm's performance varies for different input sizes, but can also vary for different inputs of the same size. Therefore, it is common to use the worst-case time complexity, $T(n)$, of an algorithm. The worst-case time complexity, $T(n)$, is the maximum amount of time taken on each input of size n. If the running time of an algorithm is upper bounded by a polynomial expression in the size of the input n of the algorithm, then it is said to be of polynomial time. Thus, the running time of an algorithm is of polynomial time when $T(n) = \mathcal{O}(n^p)$ for some non-negative integer p. This notion of polynomial running time gives rise to the classes P and NP, which are defined in the following two definitions.

Definition 1. *(Class P) The class P contains all decision problems for which the correct answer, i.e., a 'yes' or 'no'-instance, can be found in polynomial time.*

Definition 2. *(Class NP) The class NP contains all decision problems for which the correct answer, i.e., a 'yes'-instance, can be verified in polynomial time given a certificate.*

It is clear that P is a subset of NP. Suppose there is a decision problem Π that is in the class P, so there exists a polynomial-time algorithm for problem Π. Given any 'yes'-instance of Π, the polynomial-time algorithm that operates on this instance is a valid certificate for the class NP. To show that a decision problem belongs to a certain class, we use polynomial-time reductions, as defined in Definition 3.

Definition 3. *(Polynomial-time reduction) Let Π_1 and Π_2 be two decision problems. We say that Π_1 polynomially reduces to Π_2 (notation: $\Pi_1 \propto \Pi_2$), whenever any instance I_1 of Π_1 can be transformed in polynomial time to an instance I_2 of Π_2 such that I_2 is a 'yes'-instance for Π_1 if and only if I_2 is a 'yes'-instance for Π_2.*

When decision problem Π_1 polynomially reduces to decision problem Π_2 and it is known that decision problem Π_1 cannot be solved in polynomial time, then decision problem Π_2 cannot be solved in polynomial time either.
There is another subclass of NP next to the class P, namely the class NPC, see Definition 4.

Definition 4. *(Class NPC) The class NPC contains all decision problems $\Pi_1 \in NP$ for which all decision problems $\Pi_2 \in NP$ can be polynomially reduced to Π_1, i.e., $\Pi_2 \propto \Pi_1$, $\forall \Pi_2 \in NP$.*

A decision problem that in in the class NPC is called NP-complete. No polynomial-time algorithms are known for decision problems in the class NP-complete. It is expected that there exists no polynomial-time algorithm for this class.

Definition 5. *A problem Π_1 is said to be NP-hard, if $\Pi_2 \propto \Pi_1$ for every problem Π_2 in NP.*

In practice this means that a problem Π_1 is NP-hard if and only if there exists an NP-complete problem Π_2 such that $\Pi_2 \propto \Pi_2$. When a related decision problem of an optimisation problem Π_2 is NP-complete then Π_1 is NP-hard. Problems that are NP-hard are at least as hard as the hardest problem in NP. If there exists a polynomial-time algorithm to solve any NP-hard problem, then there would exist polynomial-time algorithms for all problems in NP, and P would not be a strict subset of NP, but $P = NP$.

5.3.2. Complexity of the linear problem

In this section, we prove that the linear OR schedule problem is NP-hard by formulating the related decision problem and reducing the unbounded knapsack problem to it. First we introduce the unbounded knapsack problem. Then we make a polynomial-time reduction from the unbounded knapsack problem to the decision problem related to the OR schedule problem.

Unbounded knapsack problem

The Unbounded Knapsack Problem (UKP) is an NP-hard combinatorial optimisation problem [35]. It is formulated as follows:

Definition 6. *(Unbounded knapsack problem) Given a knapsack of capacity c and a set of n items, with item j having weight w_j and profit of p_j the problem is as follows:*

$$
\begin{aligned}
\max \quad & \sum_{j=1}^{n} p_j x_j \\
s.\ t. \quad & \sum_{j=1}^{n} w_j x_j \le c, \\
& x_j \ge 0 \text{ and integer for } j = 1, \ldots, n
\end{aligned}
$$

where the variable x_j is equal to the number of times item j is put into the knapsack. The unbounded knapsack problem describes the problem of selecting items such that the weight of the combined items is less than or equal to the capacity of the knapsack, while maximising the profit. Items can be selected multiple times. The related decision problem is given in Definition 7.

Definition 7. *(Decision problem unbounded knapsack) Given a knapsack of capacity c and a set of n items, with item j having weight w_j and a profit of p_j. Can a profit of at least K be achieved without exceeding the capacity c of the knapsack?*

A proof that the decision problem of unbounded knapsack is NP-complete can be found in Papadimitriou and Steiglitz [35].

Reduction to the linear OR schedule problem

In this section, we show that there exists a polynomial-time reduction of the unbounded knapsack problem to the linear OR schedule problem.

Let I_1 be an instance for the unbounded knapsack decision problem with n items with profit p_j and weight w_j for item j. The capacity of the knapsack is c. Binary variable x_j is equal to the number of times item j is placed in the knapsack. This is a 'yes'-instance when the profit is at least K.
Now, we construct instance I_2 for the decision problem related to the OR schedule problem with n surgery groups, where there is one OR, a planning horizon of one day, one specialty and one ward. This means we have one OR-day with available time τ. All surgery groups can be scheduled on this day, because they all belong to the same specialty. Suppose $\alpha = 0.5$, then the standard deviation of the surgery groups are not taken into account. Therefore, the surgery duration for surgery group j is given by μ_j. Because it is a cyclic schedule and there is only one OR-day, the expected number of patients on the ward, e_j^W, resulting from scheduling surgery group j is equal to the expected number of days a patient from surgery group j stays at the ward. We minimise the total number of expected patients. The surgery durations are defined as

$$\mu_j = w_j, \text{ for } j = 1, ..., n$$

and the expected number of patients resulting from scheduling group j is defined as

$$e_j^W = -p_j + \mu_j, \text{ for } j = 1, ..., n.$$

The available amount of time on the OR-day, τ, is given by $\tau = c$ and the objective function value should be at least $L = K$.

Claim 1. *The decision problem related to the OR schedule problem is in NP.*

Given which surgery groups are scheduled, it is not difficult to check whether the sum of surgery durations is less than the available time. Therefore, the problem is in NP.

Claim 2. *I_1 is a 'yes'-instance for the unbounded knapsack problem if and only if I_2 is a 'yes'-instance for the decision problem related to the OR schedule problem. Thus, $\sum_{j=1}^n (\mu_j - e_j^W) z_j \geq L$.*

Proof

⇒ Assume I_1 is a 'yes'-instance for the unbounded knapsack decision problem. Let integer variable x_j be equal to the number of times item j is assigned to the knapsack. Then $\sum_{j=1}^n p_j x_j \geq K$ and $\sum_{j=1}^n w_j x_j \leq c$. Now, $z_j = x_j$, where z_j is the integer variable that denotes how many times surgery group j is assigned to the OR-day. Then

$$\sum_{j=1}^{n} \mu_j z_j = \sum_{j=1}^{n} w_j x_j \leq c = \tau,$$

so the overtime constraint is not violated. Furthermore, $\gamma = \sum_{j=1}^n e_j^W z_j$ and the objective function value is given by

$$\sum_{j=1}^{n} \mu_j z_j - \gamma = \sum_{j=1}^{n} (\mu_j - e_j^W) z_j.$$

We have that $\sum_{j=1}^{n} p_j z_j \geq K$, and therefore, by definition of e_j^W and μ_j we get

$$\sum_{j=1}^{n} (\mu_j - e_j^W) z_j \geq K = L.$$

This means that I_2 is a 'yes'-instance for the decision problem of the OR schedule problem.

$\Leftarrow$ Assume I_2 is a 'yes'-instance for the decision problem of the OR schedule problem. Let z_j be the integer decision variable that is equal to the number of times surgery group j is scheduled on the OR-day. Then $\sum_{j=1}^{n} (\mu_j - e_j^W) z_j \geq L$. Now, $x_j = z_j$, where integer decision variable x_j denotes the number of times item j is assigned to the knapsack. Then $\sum_{j=1}^{n} p_j x_j = \sum_{j=1}^{n} (e_j^W - \mu_j) z_j \leq L = K$ and the capacity of the knapsack is not exceeded, because $\sum_{j=1}^{n} w_j x_j = \sum_{j=1}^{n} \mu_j z_j \leq \tau = c$. So, we also have a 'yes'-instance for the unbounded knapsack problem.

We can conclude that the decision problem related to the OR schedule problem is NP-complete. Therefore, the linear OR schedule problem is NP-hard.

6

Local search approach

As described in Chapter 5, we use two approaches to solve our problem. In Chapter 5, a simplified version of our problem was introduced which can be solved with a global approach. In this chapter, we use a local approach, simulated annealing (SA), which was first introduced by Kirkpatrick *et al.* [37]. The advantage of SA is that it can immediately evaluate the true objective function instead of a linearised version as in the global approach. In Section 6.1, the approach is introduced. In Section 6.2, we describe how the parameters are set.

6.1. Introduction to simulated annealing

Simulated annealing (SA) is a local search approach. Starting from a current solution, ψ_c, it randomly selects a neighbour solution ψ_n. If the neighbour solution has a *better* objective function value than the current solution, then the neighbour solution is accepted as the new current solution. Here, with *better* objective function value we mean lower when minimising the problem and higher when maximising the problem. Because we maximise the objective function, a neighbour solution has a better objective function value when it is higher than the objective function value of the current solution.

Simple heuristics only move to better neighbours and stop when they have reached a solution where no neighbour has a better objective function value. Instead of only accepting neighbour solutions with better objective function values, it is also possible to accept a neighbour solution that has a worse objective function value than the current objective function value in order to avoid staying in a local optimum. A failure to find a better state among neighbours does not mean that there is no better solution. If the objective function value of the neighbour solution is worse than the current objective function value, we accept it as the new solution with a certain probability. This method allows for searching a larger part of the solution space. The acceptance probability depends on the current objective function value, the objective function value of the neighbour solution and on a parameter that denotes the temperature. During the search process, this temperature parameter, denoted by T, gradually decreases. When the temperature decreases, the probability of accepting worse solutions also decreases. Cooling of the temperature happens after one of the following events: ω iterations where performed or ω_{new} new solutions where accepted, where $\omega_{\text{new}} \leq \omega$. Because the next solution only depends on the current solution, the iterations form a Markov chain. The temperature is cooled by multiplying the current temperature with reduction factor ρ, where $0 < \rho < 1$, which gives the cooling scheme. The procedure stops when the final temperature, T_f is achieved. During the procedure we save the solution with the best objective function value, ψ_b. The objective function value of a solution ψ is denoted by $OB(\psi)$.

A summary of the SA procedure is given below:

Step 1 Start with an initial solution ψ_c. We use the one given by running our MILP model for one minute. Let the best overall solution $\psi_b := \psi_c$. Then, determine the objective function of this solution and set the initial temperature, $T := T_{in}$. Furthermore, set the reduction factor ρ and the final temperature T_f.

Step 2 Repeat ω times or until ω_{new} new solutions are accepted:

1. Select a random neighbour ψ_n of our current solution ψ_c and determine the objective function value.
2. If the objective function value of the neighbour solution is better than the objective function value of the current solution, $OB(\psi_n) > OB(\psi_t)$, set $\psi_c := \psi_n$. Otherwise, set $\psi_c := \psi_n$ with probability $e^{-\frac{\Delta}{T}}$, where Δ is given by the difference between the objective function value of the neighbour solution and the objective function value of the current solution, i.e. $\Delta = OB(\psi_t) - OB(\psi_n)$. If $OB(\psi_c) > OB(\psi_b)$, let $\psi_b := \psi_c$.

Step 3 Set $T := \rho T$. If $T < T_f$ then stop. Otherwise, go to **Step 2**.

The best overall solution is given by ψ_b.

6.2. Selecting parameters and neighbour solutions

In Section 6.1, we described the parameters that need to be specified: the initial temperature, the final temperature, the reduction factor and the maximum number of new solutions accepted. The effectiveness of the method greatly depends on the choices made for these parameters. Furthermore, different ways of defining neighbour states can also influence the method. First, we explain how we define neighbour solutions, and then, we describe how we set the parameters.

Neighbour solutions

To obtain feasible neighbour solutions, we set a generator function that uses the current solution to produce a new solution. We consider four different ways to generate a neighbour solution.

Removing a surgery group The first way to find a possible neighbour solution is removing one surgery group from all scheduled surgery groups in the current solution. To find a feasible new solution, it is important to only remove a surgery group if it is scheduled more often than the required minimum amount.

Adding a surgery group The second way to find a neighbour solution is taking the current solution and adding a surgery group to a certain OR-day. Again, it is important to check if adding a certain surgery group does not violate any constraint. Therefore, we check which groups can be added to which OR-days based on the overtime constraint and the specialties assigned to the OR-days. From this list with valid combinations of surgery groups and OR-days, we randomly choose one combination and add that surgery group to the specified OR-day.

Swap two OR-days We also define neighbour solutions by swapping all surgery groups between two OR-days. This can only be done if they have the same available time for surgeries and the same specialty can operate in the ORs. We do not swap two OR-days that take place on the same day, because this would not lead to a decrease in the objective function value.

Swap two groups The last way to define neighbour solutions is to swapping two different surgery groups that have been scheduled in the current solution. They can only be swapped if either the OR or the day on which they are scheduled is different. Furthermore, the new solution is only feasible if the swap happens between surgery groups from the same specialty and the overtime constraint is not violated.

These four interventions in the current solution define feasible neighbour solutions. They have an equal probability of getting chosen.

Initial temperature and final temperature

In the beginning of the procedure, we want the probability of accepting a neighbour with a worse objective function value to be relatively high. Near the end of the procedure, the probability of accepting a neighbour solution with even a slightly worse objective function value should be very low. Therefore, we choose the initial temperature T_{in} in such a way that a decrease in the objective function value is accepted with a relatively high probability. Furthermore, we choose the final temperature T_f such that a small decrease in the objective function value is only accepted with a very low probability. We observe that the maximum decrease can be caused by either removing a surgery group or by swapping two groups or two OR-days, which can lead to an increase in the variation in bed occupancy. Adding a surgery group can also lead to an increase in the variation in bed occupancy, however, the utilised OR time improves. Therefore, we focus on the other possibilities.

Removing a surgery group This decreases the objective function value with the mean surgery duration of the removed surgery group. Furthermore, this can lead to a increase of one bed in the variation of bed occupancy.

Swapping two surgery groups The OR utilisation stays the same, but the variation in bed occupancy can increase. During preliminary testing, we found that this leads to at most an increase in variation of two beds.

Swapping two OR-days Again the OR utilisation stays the same, but the variation in bed occupancy can increase. However, the increase can be much higher than two beds. During preliminary tests, the highest increase we registered was ten beds.

We do not want to accept every decrease in the objective function value with a high probability. A small decrease, such as when removing a surgery group leads to an increase in variation in bed occupancy, can lead to a better solution later on. For example, if the removal is accepted, it is possible that another group is scheduled in place of the removed group, with a similar LoS and a higher surgery duration. This would result in the original variation in bed occupancy and a higher OR utilisation, thus, a higher objective function value. However, we do not want to accept very large decreases with a high probability, not even in the beginning of our procedure. For example, we do not want to accept swapping two OR-days that lead to a very high increase in the variation in beds, because then it can take a long time to come back to the original objective function value.
In our objective function, the number of required beds at the ward and ICUs are multiplied with the factors θ_w and θ_i respectively. In the preliminary tests, we used the parameter $\theta = \theta_w = \theta_i$, so each ward and ICU was given the same weight. Therefore, the maximum decrease that we permit, is given by $\max_{j\in\mathcal{J}} \mu_j + \theta$, which depends on the parameters θ and the surgery groups $\mathcal{J}$. At the start of the procedure, we want to accept the maximum decrease permitted with a probability 0.5. Thus, the initial temperature is given by

$$T_{in} = \frac{\text{max decrease permitted}}{\ln(0.5)} \tag{6.1}$$

We determine the final temperature using the same approach. Near the end of the procedure we want to accept small negative changes in the objective function with a very low probability. This way, no worse solution is accepted and the procedure converges to a local minimum. Our minimum negative change is given by removing the surgery group with the shortest surgery duration, while not influencing the variation in bed occupancy. We set the probability of accepting this change to 0.001, i.e.,

$$T_f = \frac{-\min_{j\in\mathcal{J}} \mu_j}{\ln(0.001)} \tag{6.2}$$

Other parameters

Next to the initial and final temperature values, we also need to set the reduction factor, the number of iterations per temperature and the maximum of accepted solutions per temperature. We set the reduction factor ρ to 0.95, so the system will cool slowly. Furthermore, the number of iterations per temperature value is based on the number of neighbour solutions that can be reached in one step. This is dependent on which way a neighbour solution is chosen.

- The number of feasible neighbour solutions when removing a surgery group, depends on how many surgery groups are scheduled more often than necessary in the current solution.
- When adding a surgery group the number of feasible neighbour solutions depends on how many OR-days in the current solution still have available time and which groups can be scheduled there.
- Swapping OR-days can only occur when the ORs have the same specialty assigned to them.
- Swapping two surgery groups is only permitted if they are of the same specialty and the overtime in not violated after the swap.

For the data of the LUMC, all of these interventions have a comparable amount of neighbours at the start of the procedure. Because there is no guarantee these are the best choices for the parameters, we tested the effect of varying the different parameters. Results of these tests can be found in Chapter 7. With these results we determined the best combination of parameters to make a consideration between solution time and the quality of the solution found.

7

Computational results

In Chapter 4, we introduced the method we use to form surgery groups based on LoS and surgery duration. We discuss the collected data and the results of the clustering method in Section 7.1. Then, we analyse the parameter settings for both the MILP and the SA approach, which were introduced in Chapter 5 and Chapter 6. We perform multiple numerical experiments to determine the best settings for both methods. In the global approach, we calculate the objective function value differently from the SA approach. Therefore, we determine the original objective function value for the resulting OR-schedule afterwards, using the 90-percentile. In Section 7.2, results of the MILP for different parameter settings are given. The results of the SA approach for different initial temperatures, a different cooling schedule and other parameters are given in Section 7.3. We compare the two methods in Section 7.4. In Section 7.5, we try to improve our best solution from the MILP model with the SA procedure. We compare the performance indicators of the best solution with the performance of the LUMC in 2015 in Section 7.6.

The performance indicators are based on the two objectives of this research: minimising the variation in the required number of beds and maximising the OR utilisation. The required number of beds at the wards and ICUs are expected to be lower in the weekends, since no surgeries are preformed on Saturdays and Sundays. Patients can be discharged in the weekend, but no new patients are admitted. Therefore, there is a difference between the variation in bed occupancy during the planning horizon with Saturdays and Sundays and without Saturdays and Sundays. The main problem with the variation in bed occupancy is that elective patients are cancelled, because there are not enough available beds at the ward due to a high bed occupancy. This mainly happens during the week days. Our approach calculates the required number of beds for all days of the planning horizon. We look at the variation in required number of beds in the planning horizon without Saturdays and Sundays. This is the difference between the maximum and minimum number of beds needed during the week. To compare solutions, we sum the difference in required number of beds over all wards and ICUs and call this the *difference in beds*, which refers to the variation in required number of beds.
Another performance indicator is the computation time. We record the computation time for the different parameter settings. This model is used to create a tactical schedule, which is a two week cyclic schedule for a whole year. The model will be used a few times a year to incorporate possible changes. Therefore, the computation time does not need to be very short. However, because the created schedule needs to be checked by physicians and other personnel, it would be preferable if the computation time is less than a day, so small changes can be easily applied.
A less important performance indicator is the total number of required beds. This is the sum of the maximum number of beds needed at each ward and ICU. This indicator is not used to determine how good a solution is. It is only used to draw conclusions about required number of beds and other performance indicators.

Solving the MILP model is done by using version 4.2.3 of AIMMS. AIMMS is a software system that is meant for modelling and solving optimisation and scheduling-type problems. Multiple solvers are linked to AIMMS. For our MILP model, we use CPLEX version 12.6.3 that uses a branch and cut method. The clustering method and the SA procedure are implemented in MATLAB R2016b. All computational experiments are performed on a PC with an Intel Core i7 6700K 4.20 GHz with 16.00 GB RAM.

7.1. Surgery groups

As mentioned in the first two chapters, this research was performed at the LUMC. The goal of the research is to give the LUMC more insight in the way that the OR schedule influences the bed occupancy and provide them with a different way of scheduling surgeries. Therefore, the data used is based on the data of the LUMC. We have a master surgery schedule where each OR-day is assigned to a specialty. The planning horizon is 14 days, with 20 ORs where 11 different surgical specialties operate. We have 13 different wards and one ICU.

Data was gathered from interviews with surgeons, OR personnel, ward nurses, managers and the hospital data warehouse (DWH). From DWH, we obtained the OR, ward and ICU data of all the patients that were operated on in 2015. The surgery groups are determined based on this data. We linked the data of these units to get a dataset with information about the surgical procedure, surgery duration, length of stay at the ICU and the length of stay at the ward. If certain data was missing, which lead to no information about one of these variables, the data was excluded. No registration of the end time of the surgery and a wrong entry for surgical procedure caused the most deleted data. 75% of the total dataset was used.

Patients can have multiple surgeries during a hospital stay. However, we do not take this into account in our model. We take the main surgery of every patient and link this to the information from the ward.

In practice, patients are sometimes transferred to another ward, because of capacity issues or a change in the patients medical condition. Since we develop a tactical schedule and assume that the system operates as it should, patient transfers between wards are not taken into account in our model. Therefore, we use the ward they spend the most time on to link to the other available data. The LoS on the ICU is, however, taken into account. It is assumed that after surgery, patients go to the ICU and then to the ward or directly to the ward after surgery. However, there are a few instances when patients go back to the ICU when they were already at the ward. These instances are adapted to fit the assumed path. This gives us a dataset with the variables: specialty, procedure, ICU LoS, ward LoS and surgery duration.

For each specialty, we use the clustering method to determine the threshold between the short stay group and the long stay group. For most specialties, this threshold is between 0.5 and 2.5 days. However, for the specialties NS and CTS, the threshold is 4.4 days and 7.7 days, respectively. The procedures with a median LoS of less than the threshold are denoted as short stay, while the procedures with a median LoS higher than the threshold are in the long stay group. In Figure 7.1, the precision of the 22 LoS groups is shown. The precision is defined as the number of realised instances in the group that were indeed lower or higher than the threshold divided by the total realised instances in the group. The procedures of each LoS group are then divided over three surgery groups based on the surgery duration of the procedures. Two thresholds are determined and procedures are put into a short, medium or long surgery duration groups dependent on their mean surgery duration. However, some LoS groups did not contain enough procedures to be split into three significantly different surgery groups. In these cases, only two surgery groups are defined. This approach leads to a total of 62 different surgery groups. In Figure 7.2, the precision of all these surgery groups is shown. Four surgery groups have a precision of less than 0.6. These groups all belong to different specialties and are the groups that contain the procedures with medium surgery duration. For these specific groups, the two thresholds defining the three surgery groups are very close together, namely less than 30 minutes. Therefore, the mean surgery duration of certain procedures may fall into the interval between the two thresholds, but many realised instances are outside these bounds, which leads to the low precision. However, the three groups have significantly different means, and therefore, the method does define three groups instead of two. We could adjust the method to define only two groups instead of three groups whenever the precision is lower than a certain parameter. However, this may lead to surgery groups with a higher variance.

For each group, the mean surgery duration and variance in surgery duration are determined. Furthermore, we determine the probability of patients from a surgery group going to ICU i and ward w. With the model described in Section 4.4, we determine the patient distribution resulting from scheduling the surgery groups. The changeover time between surgeries is set to 15 minutes.

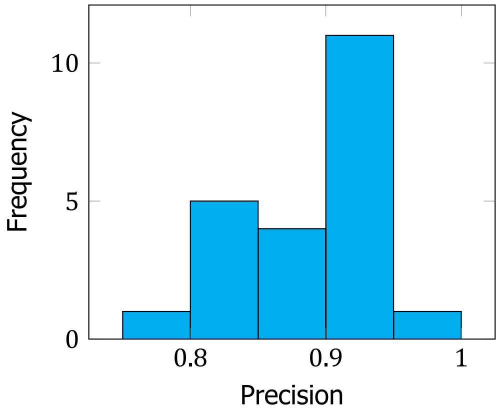

Figure 7.1: Precision of the length of stay groups.

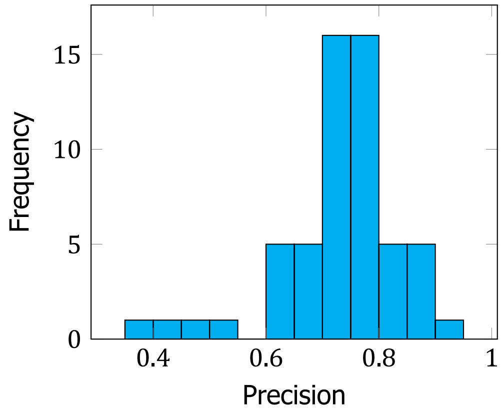

Figure 7.2: Precision of the surgery groups.

7.2. Global approach

In this section, results are given for the global approach. Using all data proved to be difficult, because of the computation time needed. After three hours, no solution was found when using the whole dataset. Fortunately, two specialties, NS and CTS, have designated ORs and wards. Therefore, we separate the data for those specialties from our main data. The specialties NS and CTS are implemented separately in AIMMS. For the computational results given in the following sections, we used the data of all other specialties. Thus, we use the data of 9 surgical specialties, that can operate in 13 ORs with a planning horizon of 14 days, which results in a total of 1107 OR-days.

In our MILP model, only a few parameters are not recovered from data, because they can be influenced by management. We have the parameter α that denotes the probability of overtime occurring, and in the objective function, we have the parameters θ_w and θ_i that denote the balance between the OR utilisation and the variation in bed occupancy at the wards and ICU. These parameters can be set by the management. In our basic model, every ward and ICU is equally important. Therefore, we let $\theta = \theta_w = \theta_i$ for all $w \in W$ and $i \in I$. First, we analyse the influence of different values of θ on the solution. The OR utilisation is given in minutes of utilised OR time. This means that when we set the factor $\theta = 200$, we would remove scheduled surgery groups with a total OR time of 200 minutes if this would lead to a decrease of one bed in the variation of number of required beds. Thus, it is a factor that determines how many OR minutes a decrease of one bed in variation is worth. Because we have a high number of total OR minutes in our schedule, we denote the OR utilisation by the percentage of the total available OR time in our tables. After analysing the influence of different values of θ, we analyse the influence of the parameter α. In our basic model, we used $\alpha = 0.3$, so the probability that overtime will occur is 30%. We interrupt the CPLEX solver after a set computation time, because proving optimality of a solution can take up quite some time. When solving, AIMMS also calculates the linear programming (LP) bound of the problem. This is an upper bound for our solution, and from this, we can calculate the optimality gap, which is the gap between the current best solution and the upper bound.

In Table 7.1, we provide some performance indicators for different values of θ with a computation time of 10 minutes. We cannot compare the objective function values of the solutions, because θ influences the objective function value. When $\theta = 0$, we are only optimising the OR utilisation, which results in a high OR utilisation, but also in a high variation in number of required beds at the wards and ICU. With $\theta = 1000$, the variation in number of required beds is low, but the OR utilisation has also decreased a lot. The total sum of all available hours in the OR for the specialties we are scheduling is 950 hours. Thus, a decrease of 10% in OR utilisation means that the utilisation decreased with 95 hours. Because this is a lot, we prefer a value of θ that leads to a higher OR utilisation. Therefore, we analyse the solutions for θ in between 0 and 1000. In Table 7.1, we see that the optimality gap increases when θ increases. The difficulty of the problem increases when the number of required beds gets more important.

θ	OR utilisation	Required beds	Difference in beds	Optimality gap
$\theta = 0$	87.59%	177	54	4.68%
$\theta = 200$	84.82%	158	20	15.17%
$\theta = 300$	84.50%	155	18	19.72%
$\theta = 400$	83.97%	153	19	20.21%
$\theta = 500$	82.87%	156	18	23.64%
$\theta = 800$	81.40%	155	17	29.51%
$\theta = 1000$	77.38%	141	16	45.98%

Table 7.1: Results for different θ's used in AIMMS with 10 minutes computation time with $\alpha = 0.3$

In Table 7.1, we see that for $\theta = 300$, $\theta = 400$ and $\theta = 500$ the variation in number of required beds is almost equal. However, the OR utilisation is higher for $\theta = 300$. The other values presented in the table lead to either a lower OR utilisation with a small decrease in variation of number of required beds or to more variation in the number of required beds, with only a small increase in OR utilisation. Therefore, we look more closely at solutions for $\theta = 300$, $\theta = 400$ and $\theta = 500$. In Table 7.2, we present solutions for $\theta = 300$, $\theta = 400$ and $\theta = 500$ for different computation times, namely 10, 30 or 90 minutes. We see that for $\theta = 300$, the OR utilisation increases slightly with a longer computation time. The variation in number of required beds goes down by one bed for a computation time of 90 minutes compared to the solution after 10 minutes. The OR utilisation also increases for longer computation times, when $\theta = 400$, but after 90 minutes the number of required beds goes down by three beds compared to the solution after 10 minutes. For $\theta = 500$ the OR utilisation also slightly increases for longer computations times. It is still lower than the OR utilisation for $\theta = 300$, however, after 90 minutes computation time a solution is found where the difference between the maximum and minimum number of beds needed is only 12, which is much lower than seen in other solutions. Therefore, we use $\theta = 500$ with further experiments, unless specified otherwise.

θ	Computation time	OR utilisation	Required beds	Difference in beds
$\theta = 300$	10 min	84.50%	155	18
$\theta = 300$	30 min	85.05%	157	19
$\theta = 300$	90 min	85.41%	156	17
$\theta = 400$	10 min	83.97%	153	19
$\theta = 400$	30 min	83.98%	152	17
$\theta = 400$	60 min	84.24%	155	16
$\theta = 500$	10 min	82.87%	156	18
$\theta = 500$	30 min	83.09%	151	16
$\theta = 500$	90 min	83.91%	152	12

Table 7.2: Results for different θ's and computation times with $\alpha = 0.3$

In the overtime constraint, we have parameter α which denotes the probability of overtime occurring. We set the computation time to 10 minutes and consider different values for α, see Table 7.3. For $\alpha = 0.5$, the standard deviation of the surgery groups is not taken into account at all and no additional slack is scheduled. The schedule is made with just the mean surgery duration. This means there is a probability of 50% that overtime will occur. The OR utilisation decreases when α increases, which is to be expected, because OR schedules with less surgeries will have a lower probability of overtime. We did not include lower values of α, because after 10 minutes of computation time these are still infeasible. In Table 7.3, the optimality gap for different values of α is also given. When α takes on higher values, the optimality gap decreases for the same computation time. The objective function value is higher for solutions with higher α Using a low value of α decreases the probability of overtime occurring, but increases the probability of not using all available OR time. We use $\alpha = 0.3$ in further experiments, because we want the probability of overtime to be relatively low, while the OR utilisation is relatively high.

In Figure 7.3, the LP bound and current best solution are shown over time. We can see that setting longer computation times allows the solution to improve, thus, the optimality gap decreases over time. However, the speed of improvement also decreases a lot over time. Because we have a tactical schedule, which in theory should only be calculated a couple of times a year, a computation time of 90 minutes is acceptable.

Probability overtime	Objective function value	OR utilisation	Required beds	Difference in Beds	Optimality gap
$\alpha = 0.5$	45576	94.97%	174	17	7.27%
$\alpha = 0.4$	40660	88.97%	165	20	17.77%
$\alpha = 0.3$	38186	82.87%	156	18	23.64%
$\alpha = 0.25$	37667	81.08%	148	17	24.52%

Table 7.3: Results for different values of α, the probability that overtime will occur, with computation time set to 10 minutes.

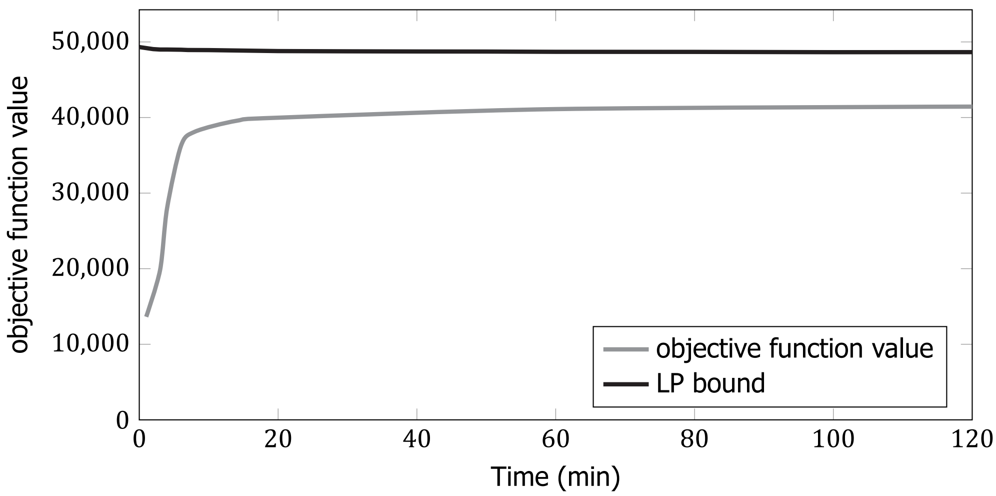

Figure 7.3: The upper LP bound and objective function value of the best solution over time.

7.3. Local search heuristic

In this section, we discuss results from our local search heuristic SA. Our initial solution comes from running AIMMS for 60 seconds. First, we run preliminary tests to find the right parameter settings. We discuss the initial temperature, reduction factor, final temperature and the maximum number of iterations within one temperature. As in Section 7.2, we use $\alpha = 0.3$ and $\theta = 500$. Furthermore, the same dataset is used. In Table 7.4, the standard settings of the parameters in our SA procedure are given. We want to compare the results for different parameter settings. Therefore, we use a seed to initialise the random number generator that generates neighbour solutions and the acceptance probabilities of these solutions. For the SA procedure, we do not have an upper bound, because we do not know how close we are to the optimal solution. The upper bound of the MILP method does not work, because it does not take the 90-percentile for the number of required beds into account.

Symbol	Standard value	Description
T_{in}	1000	Initial temperature
T_f	5	Final temperature
ρ	0.95	Reduction factor
ω	300	Number of iterations in one temperature
ω_{new}	20	Maximum number of new solutions accepted in one temperature

Table 7.4: Standard settings of the parameters used in the SA procedure.

Initial temperature

In Chapter 6, we showed that theoretically the initial temperature should be defined by

$$T_{in} = \frac{\text{max decrease } \textit{permitted}}{\ln(0.5)} \tag{7.1}$$

Using our data, this results in $T_{in} \approx 1000$. However, we could also set the initial temperature in such a way that the maximum *possible* decrease is accepted with probability 0.5. The maximum *possible* decrease is a lot higher than the maximum *permitted* decrease that we used to determine $T_{in} = 1000$. Accepting the maximum possible decrease with probability 0.5 gives an initial temperature of $T_{in} \approx 10000$. Furthermore, we compare these results with the results for setting the initial temperature to $T_{in} = 100$, which would be the initial temperature when we would only want to accept very small changes in the objective function value. In Figure 7.4, we see that for $T_{in} = 10000$ during the first 300 iterations the objective function value goes down. It takes almost 300 iterations to start improving on the original objective function value. This can be easily explained. By setting the initial temperature in such a way that it accepts the maximum decrease of the objective function value at a high probability, all relatively small decreases are accepted with a very high probability. A high amount of these smaller decreases results in a large decrease of the objective function value. For $T_{in} = 1000$, the objective function value starts improving immediately, however, small negative changes do occur. For $T_{in} = 100$, we can conclude that the procedure almost never excepts worse neighbour solutions as the objective function value is almost always increasing. In Table 7.5, we see the results of the best solutions for different initial temperatures. Because we used the same final temperature, the computation time of lower initial temperatures is shorter. The solution found with initial temperature $T_{in} = 10000$ has the best objective function value. However, it also has the longest computation time and most iterations. The initial temperature $T_{in} = 1000$ is preferred, because it starts improving the solution immediately.

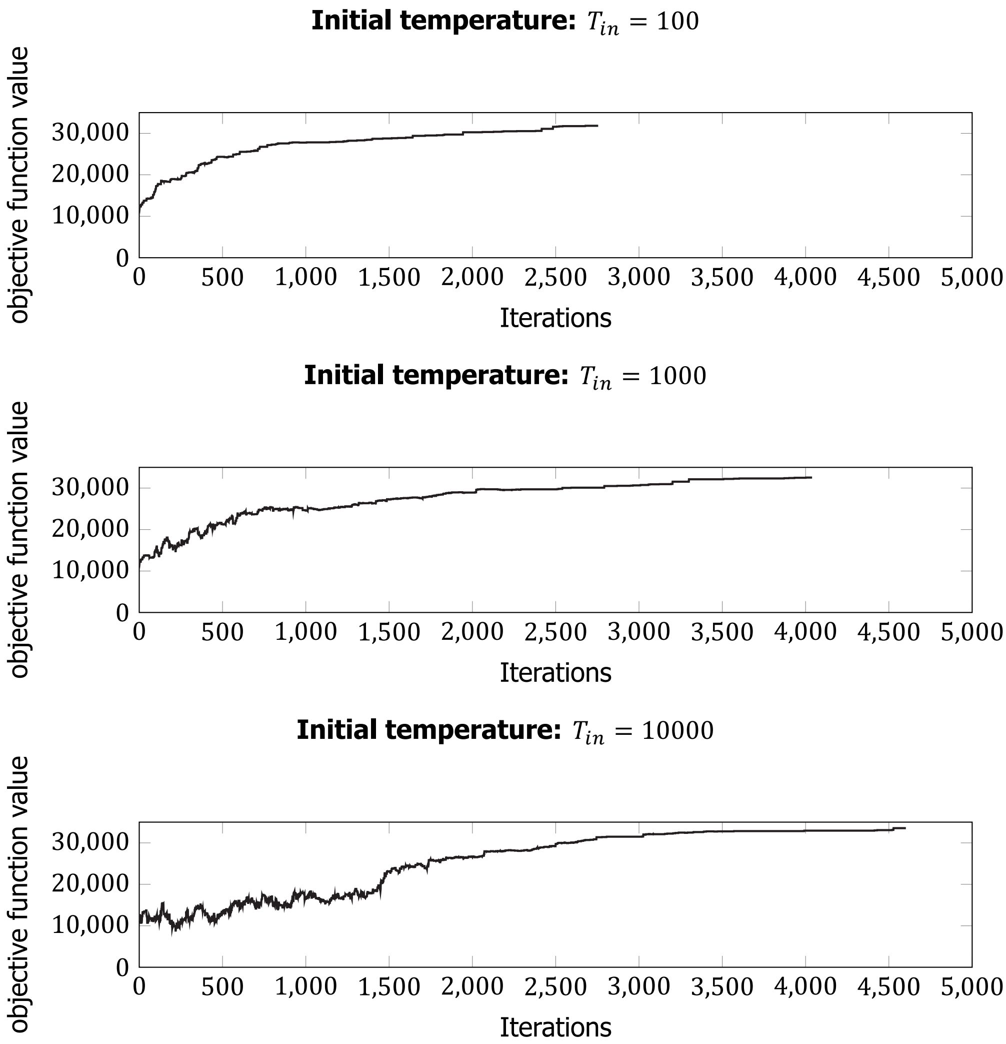

Figure 7.4: The objective function value during the SA procedure for different initial temperatures.

Initial temperature	Objective function value	OR utilisation	Required beds	Difference in beds	Computation time (min)
100	31742	74.19%	136	21	38
1000	32490	74.62%	138	20	63
10000	33132	82.77%	141	28	83

Table 7.5: Results for different initial temperatures.

Cooling schedule

We also tested different cooling schedules. Using a lower reduction factor ρ, leads to a quicker cooling schedule. In total, this leads to fewer iterations, so shorter computation times. Moreover, the acceptance probabilities of decreases in the objective function value change a lot faster than with a higher reduction factor. In Table 7.6, results are shown for decreasing the reduction factor to 0.6 and 0.8 and increasing it to 0.97. The computation time decreases a lot for lower reduction factors. However, the objective function values of the solutions are much lower than for the reduction factor $\rho = 0.95$. Increasing the reduction factor leads to a much better result, however, the computation time increases.

Cooling factor	Objective function value	OR utilisation	Required beds	Difference in beds	Computation time (min)
ρ =0.6	20240	68.91%	138	38	4
ρ =0.8	27214	71.50%	135	27	12
ρ = 0.95	32490	74.62%	138	20	63
ρ = 0.97	35595	76.56%	136	16	100

Table 7.6: Results for different cooling factors.

Final temperature

In Chapter 6, we showed that theoretically the final temperature should be defined by

$$T_f = \frac{-\min_{j\in\mathcal{J}} \mu_j}{\ln(0.001)} \tag{7.2}$$

Using our data, this results in $T_f \approx 5$. In Table 7.7, results are shown for different final temperatures. If we were to use a lower final temperature, we would perform more iterations so the Markov chain would increase in length. Using a higher temperature results in a shorter Markov chain. Increasing the final temperature to 10, results in a shorter computation time, however, in Figure 7.5 we can see that the solution has not yet converged. For $T_f = 1$, it looks like the procedure converged to a solution. No changes are made the last 1000 iterations. Therefore, we prefer $T_f = 1$, even though the computation time increases.

Final temperature	Objective function value	OR utilisation	Required beds	Difference in beds	Computation time (min)
$T_f = 1$	33476	76.35%	139	20	83
$T_f = 3$	33086	75.67%	138	20	67
$T_f = 5$	32490	74.62%	138	20	63
$T_f = 10$	31548	73.85%	137	21	48

Table 7.7: Results for different final temperatures.

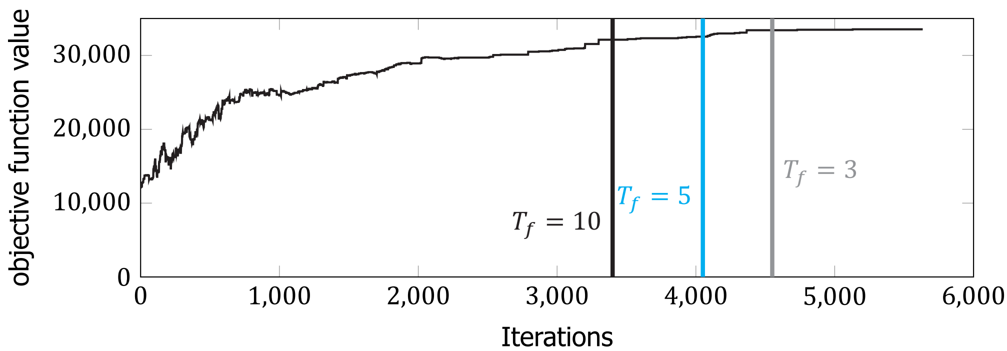

Figure 7.5: The objective function value during the SA procedure for different final temperatures.

Maximum number of success

We can also increase the length of the Markov chain by increasing ω and ω_{new}, the maximum number of successes in one temperature. First, we only increase ω_{new} while using the same value of ω. In Table

7.8, we see that increasing the number of iterations improves the objective function value, however, the computation times also increases. Because the increase in computation time is so large, we used $\omega_{\text{new}} = 20$ for other preliminary tests.

Maximum number of success	Objective function value	OR utilisation	Required beds	Difference in beds	Computation time (min)	Iterations
$\omega_{\text{new}} = 20$	33397	79.37%	148	24	63	3959
$\omega_{\text{new}} = 50$	34161	81.07%	147	24	155	10114
$\omega_{\text{new}} = 100$	37552	83.51%	153	20	250	15374

Table 7.8: Results for different values of ω_{new}.

We also tried to increase the Markov chain, by increasing both ω and ω_{new}. In Table 7.9, we see results for different values of ω_{new} and ω. We set $\omega = 3 \cdot \omega_{\text{new}}$ and vary the values of ω_{new}. The Markov chain grows for higher values of ω_{new}. However, a higher number of iterations does not necessarily leads to a better solution, but it does mean a longer computation time. When using a different number of iterations per temperature, the cooling to a new temperature happens at a different solution. Therefore, it is possible that the best objective function value obtained when $\omega_{new} = 200$ is less high than the best objective function value for $\omega_{new} = 150$, even though the same choices are made. The best solution is obtained with $\omega_{\text{new}} = 150$ and $\omega = 450$. The computation time is just over six hours.

Maximum number of success	Objective function value	OR utilisation	Required beds	Difference in beds	Computation time (min)	Iterations
$\omega_{\text{new}} = 100$	35963	85.11%	160	25	307	17807
$\omega_{\text{new}} = 150$	38699	85.53%	159	20	412	24137
$\omega_{\text{new}} = 200$	38031	86.11%	162	22	513	29556

Table 7.9: Results for different values of ω_{new} and ω.

Best solution

In conclusion, we choose to use initial temperature $T_{in} = 1000$, because this temperature allows acceptable negative changes in the objective function value, while a higher temperature allows too much negative change and a lower temperature allows no negative changes. Furthermore, we have stopping temperature $T_f = 1$, which should make sure that the procedure converges to an optimal solution. To obtain a good solution, we let $\rho = 0.97$, $\omega = 450$ and $\omega_{\text{new}} = 150$.

7.4. Comparing the global and local approach

In this section, we compare the local and the global approach. To determine which of the two approaches performs better, we compare the results for the best solutions of both approaches. Both objective function values are calculated using the 90-percentile of the probability distribution of the number of required beds.

The SA procedure is very slow compared to the MILP approach. The best obtained solution is shown in Table 7.10 and took over seven hours to compute. This can be explained by the high amount of convolutions needed to calculate the objective function value. The MILP performs better in less time. Within 1.5 hours, we have a solution with a good OR utilisation and very little variation in number of required beds. Running AIMMS for seven hours, does not improve the objective function value.

Solution method	Objective function value	OR utilisation	Required beds	Difference in beds	Computation time (hours)
SA	38699	85.53%	159	20	7
MILP	41778	83.91%	152	12	1.5

Table 7.10: Results for the best solution of MILP and SA procedure.

In Figure 7.6, the 90-percentile number of required beds for the ward VBS3 is shown for both approaches. We see big dips during the weekend, but during the week the MILP approach gives a very steady number of required beds. The SA approach performs worse with a difference of two beds during the week. However, for both solutions the same amount of beds are needed at this ward, namely 14 beds, while the OR utilisation is slightly higher for the solution of the SA approach.

In Figure 7.7, we see the number of required beds at the wards KVVL and VCH1, for both methods. The solution for the KVVL obtained with the MILP results in a very steady number of required beds during the week. The SA method has more variation during the week. For the ward VCH1, it seems more difficult to decrease the variation in number of required beds. However, the MILP again outperforms the SA method, because the variation in beds is slightly less.

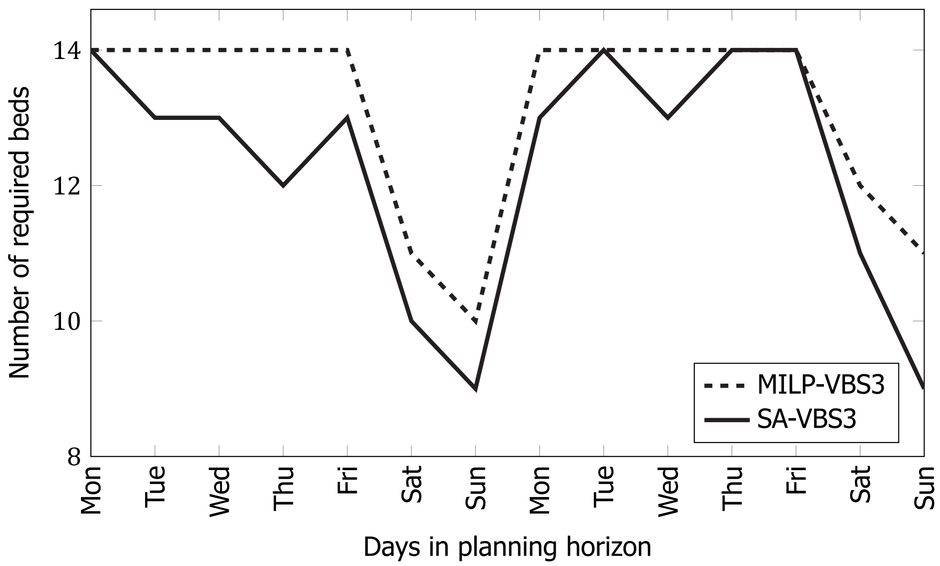

Figure 7.6: The number of required beds at the ward VBS3, from the best AIMMS and the best SA solution.

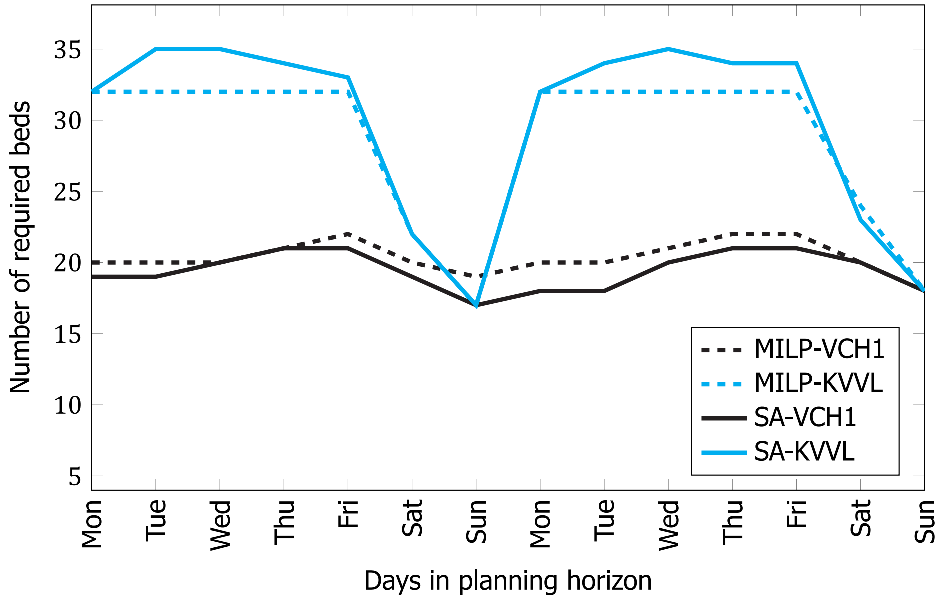

Figure 7.7: The number of required beds at the wards VCH1 and KVVL, from the best AIMMS and the best SA solution.

The MILP also outperforms the SA approach for the 85-percentile, see Table 7.11. The objective function of the MILP only depends on the expected number of beds needed, while the SA approach considers the chosen percentile during the procedure. This means that we do not have to solve the MILP again, but only determine the 85-percentile for the solution found by the MILP. We do have to run SA again to determine new solutions for the 85-percentile.

Solution method	Objective function value	OR utilisation	Required beds	Difference in beds	Computation time (hours)
SA	36518	84.33%	149	23	6
MILP	41278	83.91%	146	13	1.5

Table 7.11: Results for the 85-percentile.

The MILP performs really well, even though it uses a linearised version of the objective function. In the MILP, we use the expected number of required beds and use this to minimise the variation in number of required beds. It seems that the expected number of required beds corresponds well with the true number of required beds, which depends on the chosen percentile. In Figure 7.8 and Figure 7.9, we see for ward VCH1 and KVVL the number of required beds for different percentiles. We see that the expected value of the number of required beds is lower than for higher percentiles, however, the trend is very comparable. Therefore, the expected number of required beds is a good predictor of the difference between the maximum number of required beds and the minimum number of required beds during weekdays. On ward KVVL, we see that for the 75-percentile the difference is one bed, while for the 90-percentile and the expected value the difference is zero beds. On ward VCH1, the difference is two beds for the expected value, which is also true for the 75-percentile and the 90-percentile. This explains why the global approach performs well: the difference in beds is approximated well by the expected value of the number of required beds.

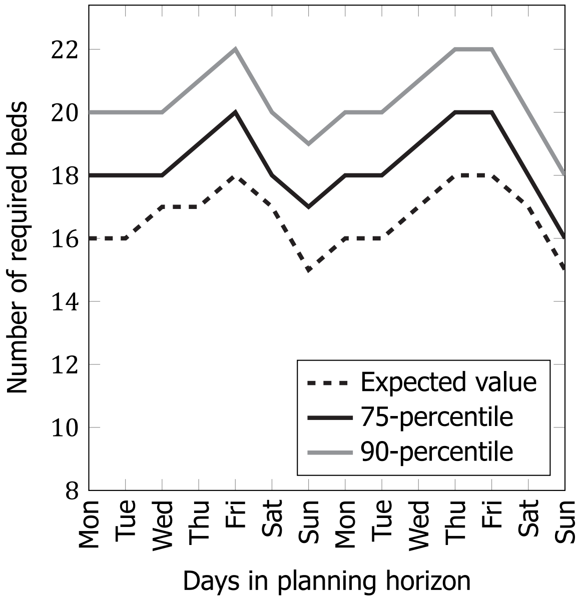

Figure 7.8: The number of required beds at the ward VCH1 for different percentiles.

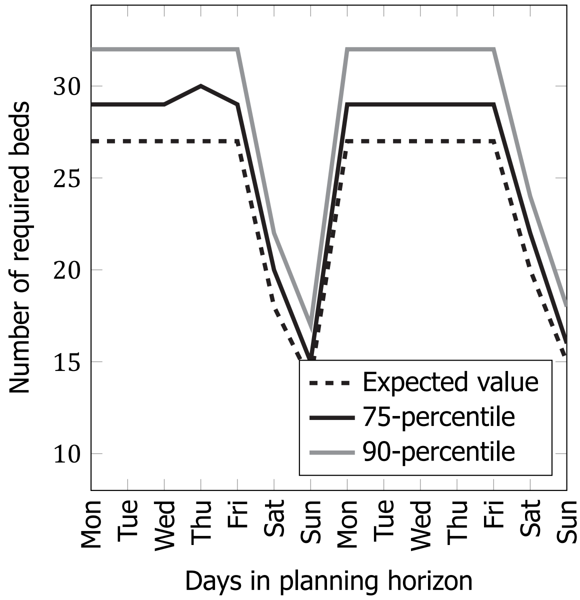

Figure 7.9: The number of required beds at the ward KVVL for different percentiles.

In Beliën and Demeulemeester [9], a similar comparison is made between a MILP approach and using SA. In their case, SA was also slower than the global approach, however, on all other aspects the SA procedure outperformed the global approach. In Van Essen *et al.* [14], the SA procedure is much faster than the global approach. However, the global approach gives better solutions. In this research, the SA procedure is much slower. We can give multiple reasons why this might be the case. First, in [14] and [9] OR blocks of whole days are scheduled instead of surgery groups. Because there are on average four surgery groups scheduled each OR-day, a lot more convolutions have to be calculated. Secondly, in their SA methods, neighbour solutions are defined by swapping OR blocks that are assigned to different days, while we define neighbour solutions by adding, removing and swapping surgery groups and swapping OR-days. To fill an OR schedule completely with surgery groups is computationally more intensive than having a filled OR schedule and only swapping OR blocks. This explains why in our case the global approach performs better than the local search heuristic.

Together with stakeholders we set the goal for the OR utilisation at 85%. Our best solution does not achieve this goal. We can add a constraint that ensures that the OR utilisation is higher than 85%. However, the computational complexity increases and after three hours no solutions were found.

7.5. Improving MILP solution with SA

The SA procedure needs an initial solution. In our experiments, we used a solution that our MILP approach produced after one minute. However, improving the initial solution might improve the outcome of the SA procedure. Therefore, we test if the solution generated by AIMMS after 90 minutes can be improved by the SA procedure. In Figure 7.10, we see the objective function value during the iterations of the SA procedure with $T_{in} = 1000$, the default setting. After 3534 iterations the procedure stopped due to reaching the maximum number of consecutive rejections. During the first 500 iterations, the objective function value goes down. Apparently, no neighbour solutions with better objective function values are found. After this, the solution starts to improve again. However, it never reaches the original objective function value. The returned solution is just the initial solution. With the initial temperature at $T_{in} = 1000$, we did not obtain better solutions. Therefore, we analysed different initial temperatures. Instead of treating the initial solution like we normally would in the SA procedure, so accepting a decrease in objective function value, we treat it like it is at the end of the SA procedure, so only accepting a decrease in objective function value with a very small probability. Therefore, we use initial temperatures that are much lower, namely $T_{in} = 100$ and $T_{in} = 10$. We also decrease the stopping temperature with the same factor, so we use $T_f = 10$ and $T_f = 1$ respectively. For $T_{in} = 100$, we get a slightly better solution in the first 200 iterations. It reaches the maximum number of consecutive rejections after 8500 iterations. For $T_{in} = 10$, we find a better solution after 100 iterations. It keeps improving slightly until it reaches the maximum consecutive rejections after 30 minutes of computation time. In Table 7.12, we see the performance indicators for all solutions. The computation time is given by the sum of the computation time of the MILP and the SA procedure. For $T_{in} = 10$, the OR utilisation improves with 0.66% and the variation in required number of beds goes down by one bed. With just an extra half hour of computation time, this is a good result. We therefore recommend to try and improve the best solution given by the MILP method with SA.

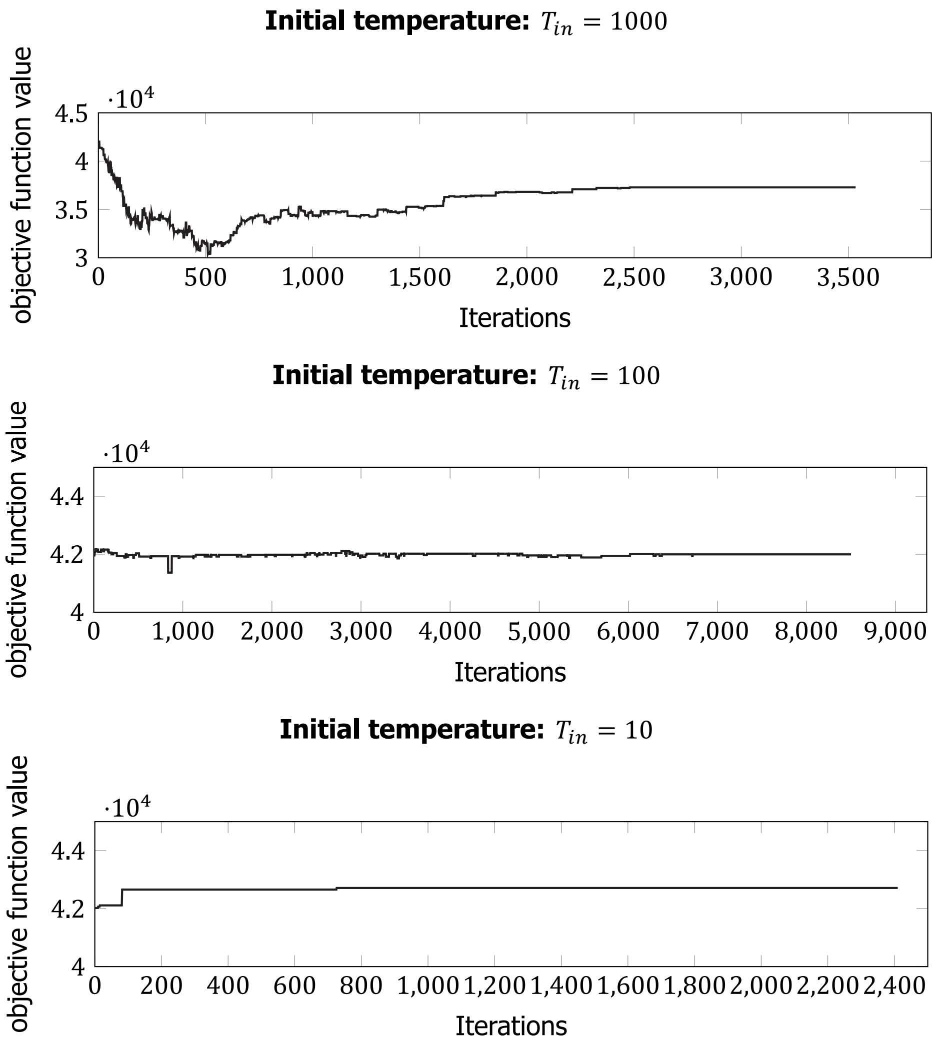

Figure 7.10: The objective function value during the SA procedure with different initial temperatures and the best solution from the global approach as initial solution.

Solution method	Objective function value	OR utilisation	Required beds	Difference in beds	Computation time (hours)
MILP	41778	83.91%	152	12	1.5
SA, $T_{in} = 1000$	41778	83.91%	152	12	3
SA, $T_{in} = 100$	42084	84.45%	152	12	4
SA, $T_{in} = 10$	42658	84.57%	152	11	2

Table 7.12: Results for the MILP, and running the SA procedure with the best MILP solution as initial solution.

7.6. Comparing results to situational analysis

In this section, we compare the performance of the model with the performance of the LUMC in 2015. In 2015, the average OR utilisation of the surgical specialties we took into account was 71%. The variation in bed occupancy during the week over all relevant wards and ICUs was 53 beds. When we compare this with our best solution, the variation in bed occupancy can be brought down to 11 beds, while the OR utilisation can be improved to 84.57%. In Table 7.13, we see the mean variation in bed occupancy for 2015 and the variation in required number of beds for each ward and the ICU. We see that for each ward the variation in bed occupancy can decrease, or in case of the VRVK, stay the same.

Ward	Historical mean variation in bed occupancy	Variation in required number of beds
DBVL	9	0
KVVL	13	0
VBS3	5	0
VCH1	4	2
VCH2	4	1
VCH3	5	2
VERL	2	1
VIG1	1	0
VPL2	2	0
VRVK	4	4
ICU	4	1

Table 7.13: The variation in bed occupancy from historical data compared to the variation in required number of beds according to our model.

8
Scenarios

We want to give more insight into the factors that influence the bed occupancy and the OR utilisation. Therefore, we use the global approach to determine solutions for different scenarios. In Section 8.1, we try to avoid having patients in the short stay ward during the weekend. The wards: VCH1, VCH2 and VCH3 are seen as important. Therefore, we perform a scenario where we increase the importance of these wards in Section 8.2. In Section 8.3, we discuss a scenario in which we minimise the total number of required beds instead of the variation. In Section 8.4, we analyse a way to include the sum of standard deviations of the surgery duration in the objective function. The influence of changeover times on the OR utilisation is shown in Section 8.5. In Section 8.6, we analyse how much there is to gain when the constraints regarding the current master surgery schedule are relaxed. To analyse if our MILP model also performs well with other datasets, we use data from the HagaZiekenhuis in Section 8.7.

8.1. Short stay ward during weekends

We would like to avoid having patients in the short stay ward during the weekend. The short stay ward closes during the weekend and patients that are still there need to be transferred to other wards. Unfortunately, this cannot be added to the model described in Section 4.4. There, we first determine the probability distribution for each ward and each surgery group. Therefore, the distribution of the number of patients on each ward is already determined before the surgery groups are assigned to days in the schedule. We can only determine the probability that patients are in the short stay ward during the weekend after the surgery groups are assigned to days in the cyclic schedule. But at that time, we can no longer influence the probability distributions of these groups, because they are already set. Therefore, we use a penalty Q when patients are on the short stay ward during the weekend. We define r as the variable that denotes how many patients there are on the short stay ward during the weekend. $r(\psi)$ is equal to the maximum number of required beds on Saturdays and Sundays. We denote the short stay ward with SSW and Saturdays and Sundays in the planning horizon are given by the set $\mathcal{L}_r$, a subset of all days in the planning horizon $\mathcal{L}$. The resulting MILP model is given by

$$\max \sum_{o\in O}\sum_{k\in\mathcal{K}}\sum_{j\in\mathcal{J}} \mu_j \cdot z_{o,k,j} - \sum_{i\in I}\theta_i\hat{\gamma}_i - \sum_{w\in\mathcal{W}}\theta_w\hat{\gamma}_w - Q\cdot r(\psi). \tag{8.1}$$

s.t. (4.1) – (4.3), (5.13) – (5.15), (5.30) – (5.33)

$$r \geq \bar{\gamma}_{SSW,l} \qquad \forall l \in \mathcal{L}_r \tag{8.2}$$

In Figure 8.1, we see the results of the model with $Q = 10000$ and a computation time of 90 minutes compared to the best solution produced by the regular MILP model. We see that the number of required beds during the weekend goes down, but does not come close to zero. It also affects the OR utilisation, which decreases by 7.5% and the difference in beds, which increases by 12 beds. For higher values of Q, the results for the KVVL stay the same. All surgery groups have a minimum demand, the minimum amount of times they need to be scheduled. In this solution, the surgery groups that include patients that can go to the KVVL are already scheduled the minimum amount of times. We can explain why it is difficult to require no beds on the short stay ward during the weekend by how we formed the surgery groups. We cluster surgical procedures into surgery groups. However, surgical procedures are not the best predictors for the ward a patient goes after surgery. Therefore, patients from certain surgery groups may have a probability of going to the short stay ward, even though the LoS of these patients can exceed the LoS of a typical short stay ward patient, which is less than five days. By changing the clustering procedure, i.e., defining surgery groups not only by surgical procedure but also by, for example, age and general health, we improve the probability of finding a schedule for which the short stay ward requires no beds during the weekend.

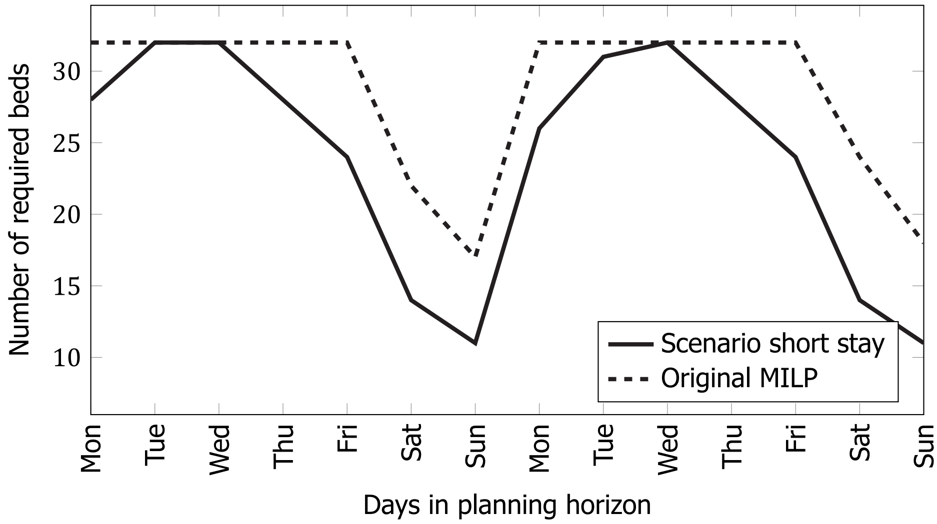

Figure 8.1: The required number of beds at the ward KVVL.

8.2. VCH1, VCH2, VCH3

According to personnel, the wards VCH1, VCH2 and VCH3 are the long stay wards with the most variation in bed occupancy. This is confirmed by data from these wards. A lot of cancellations of elective surgical patients happened, because there were no beds available on one of these wards. Our model also cannot remove all variation in required number of beds for these wards. All solutions still contain some variation in the number of required beds during the week for these three wards. Therefore, we want to run a scenario, where we increase the importance of these wards. We use θ_w to make the wards VCH1, VCH2 and VCH3 five times as important as the other wards and ICU. In Figure 8.2, we see the outcome of this scenario. For the ward VCH2, the variation in required number of beds has improved. For the other wards, the difference in required beds during the week stays the same. It does have a slightly negative influence on the variation in required number of beds of other wards as this goes up with three beds. Furthermore, the OR utilisation goes down with 3% in comparison with the solution for the basic model.

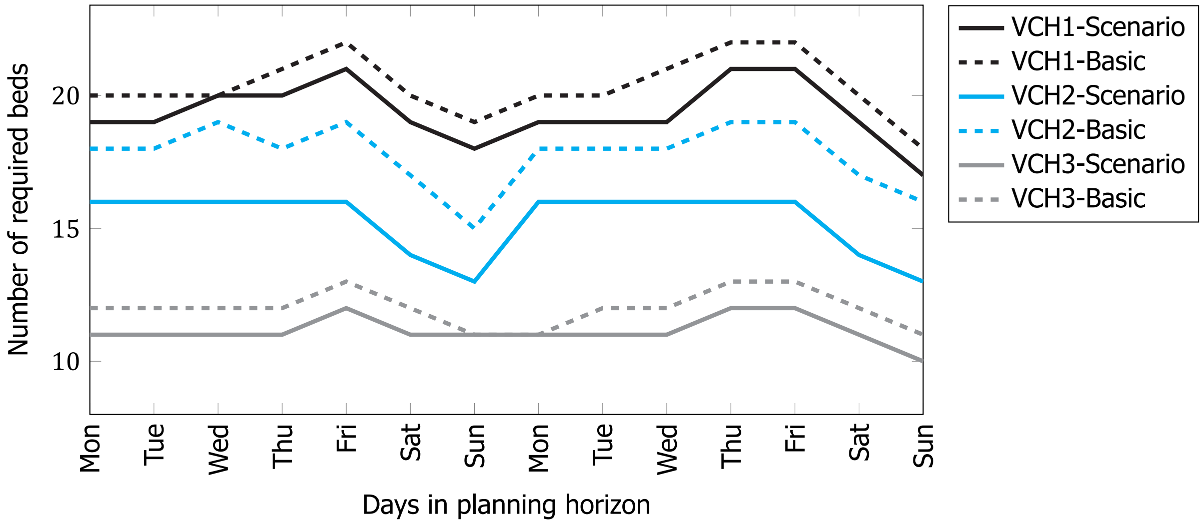

Figure 8.2: The required number of beds at the VCH.

8.3. Minimising bed occupancy

In our default model, we minimise the variation in bed occupancy. However, personnel to keep the beds open is expensive. Therefore, instead of minimising the variation in bed occupancy, we can also minimise the maximum number of required beds in the planning horizon. Even though there might be more variation, the maximum number of required beds can decrease, which can lead to needing less beds at the wards.

This is a variation of our basic linear model described in Section 5.2. Here, we use the maximum values of $\bar{\gamma}_{i,l}(\psi)$ and $\bar{\gamma}$ instead of using the difference between the maximum and minimum values of $\bar{\gamma}_{i,l}(\psi)$ and $\bar{\gamma}_{w,l}(\psi)$, because the required amount of beds during the week is equal to the maximum amount of beds needed during the week. The resulting MILP is:

$$\max \sum_{o\in\mathcal{O}}\sum_{k\in\mathcal{K}}\sum_{j\in\mathcal{J}} \left(\Theta_e \cdot \mu_j + \Theta_v \cdot \sigma_j\right) z_{o,k,j} - \sum_{i\in\mathcal{I}} \theta_i \hat{\gamma}_i - \sum_{w\in\mathcal{W}} \theta_w \hat{\gamma}_w \tag{8.3}$$

$$\begin{aligned} \text{s.t. } & (4.1)-(4.3), (5.13)-(5.15) \\ & \hat{\gamma}_i \geq \bar{\gamma}_{i,l} \qquad \forall i \in \mathcal{I}, l \in \mathcal{L} \\ & \hat{\gamma}_w \geq \bar{\gamma}_{w,l} \qquad \forall w \in \mathcal{W}, l \in \mathcal{L}. \end{aligned}$$

The results for running this scenario for 90 minutes are shown in Table 8.1. We cannot compare objective function values, because these are different for both methods. The difference in number of required beds is a lot higher, as was to be expected. However, the OR utilisation is also higher while less beds are needed in total. In Figure 8.3, we have the number of required beds for ward VCH1 and VBS3 for the scenario where we minimise the number of required beds and the basic model. Even though the basic model has less variation during the week, more beds are needed for this solution.

Solution method	OR utilisation	Required beds	Difference in beds	Computation time (hours)
min Variation	83.91%	152	12	1.5
min Beds	85.23%	147	26	1.5

Table 8.1: Results for the best solution of our basic model and the scenario in which we minimise the required number of beds.

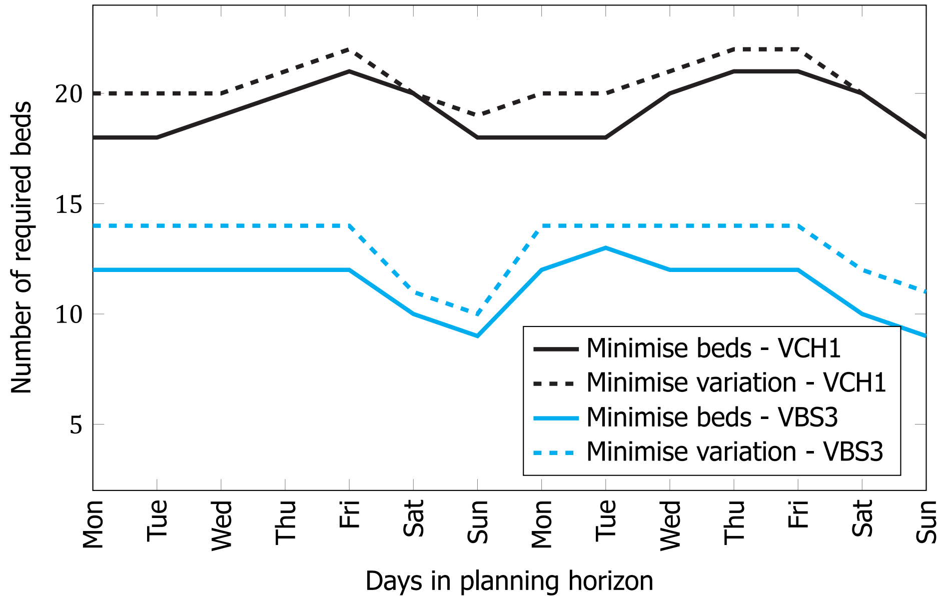

Figure 8.3: The required number of beds at the wards VBS3 and VCH1.

8.4. Standard deviation

As in the research of Van Oostrum *et al.* [2], we assume that surgery durations are normally distributed with mean μ_j and variation σ_j^2. In our basic model, we only maximise the sum of mean surgery durations. This, in combination with constraint 4.4 will most likely yield solutions that prefer surgery groups with a low variation in surgery duration. In this scenario, we include the sum of standard deviations in the objective function, so surgery groups with a high variation are more likely to be scheduled. Our resulting objective function is given by

$$\max \sum_{o \in O} \sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{J}} \left(\Theta_m \cdot \mu_j + \Theta_v \cdot \sigma_j\right) z_{o,k,j} - \sum_{i \in I} \theta_i \hat{\gamma}_i + \sum_{w \in W} \theta_w \hat{\gamma}_w(\psi). \tag{8.4}$$

where Θ_m and Θ_v are parameters that denote the importance of mean surgery duration versus the standard deviation of the surgery duration. In Figure 8.4, we see how many times surgery groups with low, middle and high variation in surgery duration are scheduled more than their minimum demand. In comparison with the basic model, slightly less surgery groups with low and middle standard deviations of the surgery duration are scheduled, while slightly more surgery groups with a high standard deviation of the surgery duration are scheduled. The OR utilisation for this solution is only 0.2% less than the OR utilisation of the basic model. The method seems to work, however, the difference in required number of beds does go up by 7 beds.

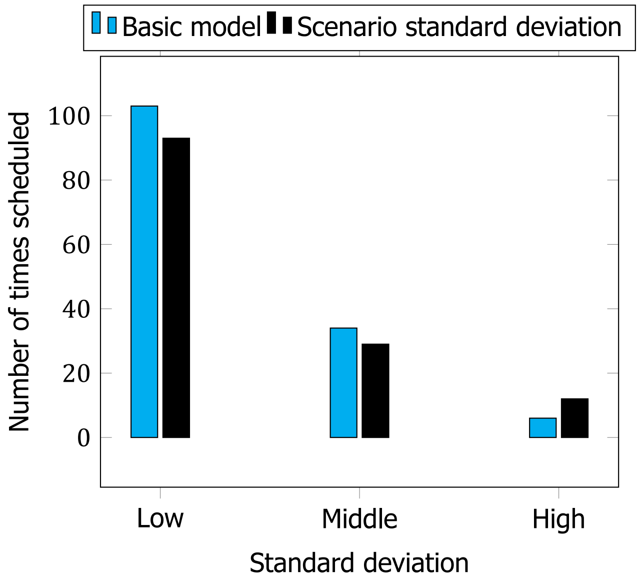

Figure 8.4: How many times groups are scheduled per standard deviation group.

8.5. Changeover times

In our default scenario, we use a changeover time of 15 minutes. The changeover time is defined as the time between one patient leaving the OR and the next patient entering the OR. This changeover time is chosen, because we want to assume the theoretically achievable changeover time. In practice, the changeover time is often higher than 15 minutes. Therefore, we compare the results for a solution with a changeover time of 15 minutes and a solution with a changeover time of 30 minutes in Table 8.2. It is clear that the changeover time heavily influences the OR utilisation. With less time between surgeries, more time can be spend on surgeries, which improves the OR utilisation. If we only take the minimum cleaning time into account, 5 minutes, the OR utilisation can improve by 5%. This is a very optimistic value for the changeover time, but is shows that in order to improve the OR utilisation, the LUMC should work towards shorter changeover times.

Changeover time	OR utilisa-tion	Required beds	Difference in beds	Computation time (hours)
5 min	89.54%	164	17	1.5
15 min	83.91%	152	12	1.5
30 min	78.36%	146	16	1.5

Table 8.2: Results for the best solution of our basic model and the scenario in which we set the changeover times.

8.6. Different master surgery schedule

In our default scenario, we use the master surgery schedule of LUMC. This means that, for example, the surgery groups that belong to the plastic surgery specialty can never be scheduled on Thursdays. We analyse whether the MSS has a large influence on the variation in number of required beds. In Table 8.3, we see that the problem becomes more complex when the MSS is not used. After 10 minutes of computation time, the optimality gap is very large and no good solution has been found. After 90 minutes of computation time, the optimality gap is still 45.81%, however, the solution improved greatly on the solution after 10 minutes. The solution has a slightly lower objective function value than the solution of the default scenario, which has an objective function value of 41778, an OR utilisation of 83.91% and the difference in beds is 12 beds. After six hours, the objective function value is higher than the objective function value of the solution of the default scenario. The OR utilisation has improved, but the difference in beds has increased compared to the solution of the MILP method for our default scenario.

Computation time (min)	Objective function value	OR utilisa-tion	Required beds	Difference in beds	Optimality gap
10 min	2106	54.63 %	127	58	885.14 %
90 min	40561	87.04%	163	18	45.81 %
180 min	41595	87.10%	162	16	42.70%
360 min	41847	87.54%	162	16	39.25%

Table 8.3: Scenario in which the MSS is disregarded.

8.7. Different data

We use a dataset from HagaZiekenhuis to analyse if our MILP model also performs well for other datasets. It is a dataset from the orthopedic surgery department. They identify 43 surgery groups, where the means, variation and minimum demand are given. Instead of different surgical specialties, here, we have surgeons who all perform a subset of the surgery groups. As with the surgical specialties in our default model, a whole OR-day is assigned to one surgeon. The LoS probability distribution is given for each surgery group. Only one ward and no ICUs are taken into account. We use the 95-percentile to calculate the required number of beds. The results can be found in Table 8.4, for different computation times. Bosch [31] uses the same dataset and probability of overtime occuring and aims to maximise the OR utilisation while minimising the required number of beds. The solution where instruments are not taken into account, which is comparable to our solution, yields a OR utilisation of 91% that needs 45 beds in total. Our solution has 1.4% less OR utilisation, however, the required number of beds goes down by six beds. In [31], first OR-days are formed based on the OR utilisation, then these OR-days are used to find a schedule which minimises the required number of beds. The method they use is much faster. Their method generates a solution in under a minute. However, the OR-days are formed beforehand, so there is no flexibility in removing or adding a surgery group to an OR-day, which could be the reason we end up with a solution that requires a lot less beds.
In [31], the aim was an OR utilisation of 90%, which they reached for most scenarios. In order to compare the solutions even better, we added a constraint to ensure the OR utilisation is at least 90%. However, after three hours no solution is found.

Computation time (min)	OR utilisa-tion	Required beds	Difference in beds	Optimality gap
10 min	86.69 %	40	3	15.35%
90 min	89.59%	39	3	12.21%

Table 8.4: Results for the data from HagaZiekenhuis.

9

Conclusions and recommendations

In Chapter 1, we have stated four research goals:

- *Minimise the variation in bed occupancy on the wards during the week.*
- *Minimise the variation in bed occupancy on the intensive care unit (ICU) during the week.*
- *Maximise the operating room (OR) utilisation.*
- *Minimise overtime in the ORs.*

Section 9.1 discusses the conclusions from this research. In Section 9.2, we give managerial recommendations and further research recommendations.

9.1. Conclusions

In Chapter 2, a situational analysis is given. Possible causes for poor performance were identified:

- Scheduling with intervention times
- Scheduling without considering the capacity of the wards
- Scheduling without considering downstream capacity needs of other surgical specialties (post anaesthesia care unit, ICU)
- Not taking the stochastic nature of surgery duration into account

In Chapter 3, we conducted a literature review into scheduling approaches to overcome the causes mentioned in Chapter 2. We developed a surgery scheduling approach based on Van Oostrum *et al.* [2]. To model the stochastic length of stay (LoS), we use a model based on Fügener [3], which is based on Vanberkel *et al.* [13].

In Chapter 4, we propose a method to cluster surgical procedures into surgery groups based on LoS and surgery duration. We contribute to literature by extending the model of Fügener [3] to include multiple ICUs, account for scheduling surgery groups instead of whole OR-days and include the possiblity that patients from the same group are going to different wards after surgery.

We use two different solution methods to solve the model: the global approach and a local search heuristic, i.e., simulated annealing. In Chapter 5, we linearise part of the objective function that is based on the analytical model to determine the required number of beds, so it can be used in a global approach. Only a simpler version of this model, as presented in Beliën and Demeulemeester [9], has ever been used in a global approach. The resulting MILP is proven to be $\mathcal{NP}$-hard. The other solution method, simulated annealing, is introduced in Chapter 6. Four possible ways to identify neighbour solutions are given.

In Chapter 7, we conducted numerical experiments for both these approaches. First, results from the clustering method are shown. Dividing surgical procedures into groups based on their average LoS resulted in well defined groups. However, dividing these groups again into groups based on the average surgery duration resulted in a few less well defined groups. This can have multiple reasons: too few surgical procedures in a group, surgical procedure is not always a good predictor of surgery duration or too much variation within surgical procedures. In Section 9.2.3, we discuss recommendations regarding the clustering procedure.

The mixed integer linear programming (MILP) method performs well within two hours. The best solution with respect to the objective function value gives an OR utilisation of 83.91%, which is 1.1% less than preferred. The variation in number of required beds is very low. To obtain the best solution for the SA procedure we had to run the procedure for seven hours. This solution does have an OR utilisation above 85%, however, the variation in number of required beds is more than 1.5 times the variation in number of required beds of the solution from the MILP.

In the MILP, the simplification of the part of the objective function that denotes the variation in required number of beds, predicts the outcomes for other percentiles well. Furthermore, the linearisation of the overtime constraint uses only five piecewise linear functions to obtain a maximum error of five minutes on the total surgery duration of the whole OR-day. This means we overestimate the total surgery duration of an OR-day with at most five minutes. It is possible to use more piecewise linear functions and obtain a lower maximum error. Overall, the global approach performs well, and is therefore, preferred over the local search heuristic. As shown in Section 7.5, we can improve the best solution of the MILP method by using it as an initial solution for the SA procedure. We recommend to use a low initial temperature in that case, to prevent a decrease in objective function value. This results in a solution with an OR utilisation of 84.57% and a total of 11 beds in variation in number of required beds.

With our global approach, we found that changing the probability that overtime will occur, highly influences the OR utilisation. Using longer or shorter changeover times also highly influences the OR utilisation. Furthermore, with this model and data, it is not possible to find a solution where the short stay ward does not require any beds during the weekend. To achieve this, surgery groups should not only be defined by surgical procedure, but also take other factors into account like age and medical condition. However, this does not guarantee a solution were the short stay ward can close during the weekend.

Our model allows us to have influence on which wards and ICUs we find more important. Furthermore, we can ensure that surgery groups with a high variation in surgery duration are scheduled more often. By changing the constraints slightly, we can minimise the required number of beds instead of minimising the variation in required number of beds. This can be useful when personnel for beds becomes more important than other factors.

In Section 7.5, we describe the improvements that can be made with our surgery scheduling approach. The OR utilisation can be improved with 13.6%, while the variation in bed occupancy can be brought down from 53 beds to 11 beds.

9.2. Recommendations

In this section, we discuss recommendations for the LUMC and further research. We propose that the LUMC applies the surgery scheduling approach described in this research. First, we discuss some general recommendations regarding available data in Section 9.2.1. Then we discuss the proposed managerial implementation. Lastly, we give recommendations for further research in Section 9.2.3.

9.2.1. General recommendations

In this section, we discuss general recommendations for the LUMC.

Define the start of OR-days Anaesthetists are responsible for two ORs at the same time. This means that during the first session of the day, an anaesthetist should perform the induction of two patients almost simultaneously. Therefore, the first patient can arrive on the OR before the anaesthetist has time to perform the induction. This can reflect in the registered surgery duration, which starts when the patient arrives on the OR. When the starting time of the two ORs differentiates, this may be prevented. Furthermore, the start of the OR-day is not clearly defined. Either the incision should be made at 8.00 or the patient should be there at 8.00. By defining the start of the OR-day clearly, late starts because ambiguity of starting times should decrease.

Schedule with session times Currently, the OR planners schedule surgeries based on intervention time (time from incision until closing the wound) and can schedule 360 minutes of intervention time on a 480 minute OR-day. The total amount of changeover time, induction time and time after closing the wound until the patient leaves the OR, can vary a lot per OR-day. It depends on the amount of surgeries and type surgeries that are scheduled on that OR-day. Therefore, we propose that planners of the LUMC start scheduling with session times, which is the time between a patient arriving at the OR until the patient leaving the OR. This session time should be estimated based on historical data. Furthermore, the planners should take the needed amount of changeover time and slack needed to prevent overtime into account.

Clearly define the case mix The case mix of the LUMC is not clearly defined for every specialty. The production agreements should be clearly stated and known to planners. Then, they can be taken into account in the model by setting the minimum demand of surgery groups in the cyclic schedule. Now, the minimum demand is based on historical data, however, this may not be accurate for the production agreements of the coming years.

Improve quality of data We recommend that the LUMC invests in improving the recording of data. A lot of data is already collected, however, the quality of the data can be improved. With more reliable data, better performance evaluations can be made. In relation to our scheduling approach we name a few important points for improvement:

- Accurate time registration of surgeries
- Time registration, registration of specialty and registration on the wards of emergency surgeries
- Registration of multidisciplinary surgeries

With accurate time registration, the surgery duration of surgical procedures can be better estimated. There could even be made a distinction between surgeons, so a more accurate estimation of the total duration of an OR-day can be made to reduce overtime.

Emergency patients are not always registered. Also, the data of the surgery of emergency patients was considerably worse than the data of surgeries of elective patients. Furthermore, the priority rating of patients had discrepancies between data from the OR department and data from the wards. From interviews with personnel we found that elective patients sometimes get a higher priority status so they can be operated on, even though they are not real emergency patients. OR management needs to set strict standards on when a patient gets an emergency patient status. With better registration of emergency patients, they could be taken into account in the model.

Another instance that was not well documented were multidisciplinary surgeries. Because the LUMC is an academic hospital, there are multidisciplinary surgeries where surgeons of multiple specialties are present. However, this was not recorded well in the available data. If better data becomes available about these type of surgeries it could be implemented in the model.

Decreasing variation in surgery duration When surgeries have less variation in surgery duration, the OR utilisation can go up with a same probability of overtime. The data collected from the LUMC showed that the variance in surgery duration for a surgical procedure can be hours.
A simple way to obtain better defined surgery durations is to clearly register when a physician in training is going to perform (part of) the surgery and when the more practised physician is performing the surgery. Then, you can make a distinction between the surgery duration when the physician in training is operating or when the physician is operating. When this is known beforehand, the variance of the surgery duration decreases, which can decrease the probability of overtime.

9.2.2. Managerial implementation

Every year, the surgery groups should be evaluated, to see if they are still up to date. The tactical schedule should be generated every 3-6 months with relevant data and surgery groups, dependent on how well the cyclic schedule is performing. The tactical schedule should be communicated to the OR department, planners, wards and the ICU. The wards and ICU can then anticipate on the resource demand of the surgery schedule. To achieve this, the LUMC board should have clear production agreements with the surgical specialties and a division of OR capacity for the cyclic schedule.

Gather data automatically. In this research, we processed the data manually to determine the surgery duration and length of stay. Two different datasets were used, one from the wards and one from the OR department. In linking the two datasets, some data was lost. Furthermore, this process was time consuming and there were discrepancies in the data from both sets. Therefore, these systems should be linked and process the historical data automatically. The surgery types should be integrated into the information system by making a new list of codes. If the information system is adjusted, it will be easier for OR planners to start using historical surgery durations to schedule surgeries.

Use production agreements. As stated before, the OR management should make clear production agreements per two week period, where up to date capacity availability is taken into account. The minimum demand of a surgery group in the planning horizon can then be based on these production agreements. Furthermore, knowledge about waiting lists should be used to determine the maximum demand of surgery groups, to prevent that they are scheduled too often regarding the supply of patients. If production agreements are met throughout the year, it should also reduce the peak of overtime at the end of the year.

Decisions regarding the model. The model requires the input of certain parameters that denote the probability of overtime and the balance between OR utilisation and variation in number of required beds. The OR management should make decisions regarding these parameters. Setting a higher probability of overtime leads to a higher OR utilisation, but will also have an effect on staff and budget. We have a parameter that denotes the balance between beds and OR utilisation. It denotes how many OR minutes are worth the decrease of one bed. We recommend that the OR management determines the costs of having an occupied bed, the costs of not using the OR for one hour and the costs of overtime. Then, a consideration can be made for the parameters. Another possibility is that the capacity of certain resources, due to a shortage of OR personnel or nurses on the wards, are the determining factor. If there is only so much available personnel, then this should be taken into account when setting the parameters.

Evaluate the performance of the model. We recommend that before putting the model into practice, a performance evaluation is done. This could help gain commitment of surgeons, OR planners and other personnel. Given a surgery schedule, the model can determine the OR utilisation, probability of overtime and the bed occupancy. This can be compared to the actual performance.

Pilot with cardiothoracic surgery (CTS) or neurosurgery (NS). We propose a pilot with one surgical specialty. We recommend to start with either CTS or NS, as both specialties have separate resources, such as their own ORs and ward. Planners from these specialties are also interested in a new way of planning. The model will be used by the patient logistics team to find a suitable cyclic schedule. The OR planners can start using the cyclic schedule to assign patients to their surgery group slot. The patient logistics team can work together with the OR planner and surgeons to implement changes in the model. During this time, the surgery group definitions can be sharpened and the surgeons availability can be taken into account. The performance of the cyclic schedule can then be evaluated.

Develop a tool that planners can use. After the model is evaluated, it is important to develop a tool, so all OR planners can easily start scheduling according to the cyclic schedule. The tool can be made in house or outsourced to a software development company. The planner of the specialty where the pilot is performed can test the tool.

Let all OR planners use the tool. When the tool is running smoothly for the pilot specialty, OR planners of other specialties can start using it too.

9.2.3. Research recommendations

In this section, we describe recommendations regarding further research. We split this into three parts: the clustering method, the model and the solution methods.

Clustering method

In this research, we propose a clustering method that takes the surgery duration and LoS into account. We create surgery groups to which we assign surgical procedures. However, surgical procedures are not always the best predictor regarding surgery duration and LoS. Many other important factors like diagnosis, age and medical condition also play an important role. Therefore, we propose to research which factors, or combination of factors, are the most important and usable. For instance, medical condition is harder to quantify than age and can therefore be difficult to use. The next step would be to adjust the clustering method to incorporate these factors. This should lead to better defined surgery groups.

Redefining the surgery groups may also help with the problem of the short stay ward. With our current model and defined surgery groups, we cannot find a schedule in which the short stay ward can completely close during the weekend. If the surgery groups are redefined, there should be surgery groups that solely go to the short stay ward. This would mean defining the surgery groups based on specialty, surgical procedure, age and medical condition. Only the surgery groups that have an average LoS of less than five days can be send to the short stay ward. It would be interesting to research if with differently defined groups it would be possible to close the short stay ward during the weekend.

Model

In this report, we created a tactical schedule where we assume that the patients all follow one of two paths. From surgery to ICU to ward or directly from surgery to ward. However, in practice patients are often transferred between wards. It would be interesting to incorporate this into the model. Furthermore, we assume that on the day of admission and the day of discharge patients occupy the bed for the whole day. In practice, this is not the case. Two patients can occupy the same bed on a day, because the first patients is discharged and the second admitted on that day. To better relate the bed occupancy from our model to the actual bed occupancy at the wards, we should determine the bed occupancy for shorter time intervals.

Currently, our model incorporates the OR department and some of the downstream resources, i.e., ICU and wards. Before arriving at the OR department, patients arrive at the hospital through the outpatient clinic. The way this is organised influences the admission rate of surgical patients. It would be interesting to incorporate this into the model, because it highly influences the waiting lists.

We recommend that in further research, an operational surgery scheduling approach based on this model is made. Then, actual patients are scheduled, so it is possible to better predict the surgery duration and LoS. In an operational model, we could also incorporate the order in which to perform the surgeries assigned to an OR day. This could be used minimise the variation on the holding area, recovery area and PACU. To achieve this, the bed occupancy needs to be determined more than once a day. By using shorter time intervals, the model can represent the actual bed occupancy.

In the operational model, we should also include emergency patients. In theory, only the emergency patients are operated on in the emergency OR. However, elective patients are sometimes urgent, such that the emergency OR can be used for their surgery. Therefore, it would be interesting to analyse if emergency ORs are the best choice for the LUMC. Without emergency ORs, the elective patients need to be scheduled in such a way that there are multiple break-in times during the day. These break-in times can be used to operate on emergency patients. The emergency ORs can then be used for elective patients.

In this report, we assumed that surgery durations were normally distributed. However, from research we know that the 3-parameter log-normal distribution can be a better fit for surgery durations. In further research, it would be interesting to use the 3-parameter log-normal distribution and compare the resulting schedule with a schedule when a normal distribution is assumed. Then, we can determine if it is worthwhile to use the 3-parameter log-normal distribution.

Solution methods

We recommend using the MILP and, if necessary, improve the solution with the SA procedure. Within two hours a good solution is found. However, our method is currently too slow to be used in an operational model. It could be rewarding to analyse if the computation time of the MILP decreases when the overtime constraint if formulated in a different way or the LoS is assumed to be deterministic. By comparing the approaches, we can determine if using a stochastic LoS is worthwhile.

A

Determing breakpoints

A good approximation comes from having a high amount of breakpoints. However, having a high amount of breakpoints also leads to a more complex ILP due to the increase in variables. Therefore, we determine the minimum amount of breakpoints needed to obtain the maximum approximate error $\Delta^{\max}$. Every linear approximation function has the highest error at the breakpoints, see Figure A.1. The approximate error δ_n at breakpoint $n \in N$ is given by

$$\delta_n = h_n(x_n) - \sqrt{x_n} \qquad \forall n > 0 \tag{A.1}$$

$$\delta_0 = h_1(x_0) - \sqrt{x_0} \tag{A.2}$$

The maximum error $\delta^{\max}$ is minimised when $\delta_0 = \delta_1 = ... = \delta_m$. The solution can be found by solving

$$h_n(x_n) = h_{n+1}(x_n) \qquad n = 1, ..., m-1 \tag{A.3}$$

$$\delta_n = \delta_{n+1} \qquad n = 0, 1, ..., m-1 \tag{A.4}$$

$$x_n < t_{n+1} < x_{n+1} \qquad n = 0, 1, ..., m-1 \tag{A.5}$$

Equation (A.3), follows from the intersections of two linear approximations. The x_n are defined as the begin- and endpoints of the linear approximation for each interval. Therefore, two successive linear approximations h_n and h_{n+1} should have the same value in their shared point x_n. Equation (A.4) ensures that all approximation errors are equal. The last Equation (A.5) ensures that the interval of the linear function is positive and that the tangent point is in between two subsequent breakpoints. Equation (A.3) can be expressed in terms of z and t. This gives Equation (A.6), which can be rewritten with the definitions of a_n and b_n from (5.8) and (5.7).

$$\begin{aligned}
a_n + b_n x_n &= a_{n+1} + b_{n+1} x_n \\
x_n &= \frac{a_{n+1} - a_n}{b_n - b_{n+1}} \\
x_n &= \frac{\sqrt{t_{n+1}} - \sqrt{t_n}}{\sqrt{\frac{1}{t_n}} - \sqrt{\frac{1}{t_{n+1}}}} = \sqrt{t_{n+1} t_n}.
\end{aligned} \tag{A.6}$$

From the second equation, Equation (A.4), it follows that

$$\begin{aligned}
\delta_n &= \delta_{n+1} \\
h_{n+1}(x_n) - \sqrt{x_n} &= h_{n+1}(x_{n+1}) - \sqrt{x_{n+1}} \\
x_{n+1} &= \left(2\sqrt{t_{n+1}} - \sqrt{x_n}\right)^2
\end{aligned} \tag{A.7}$$

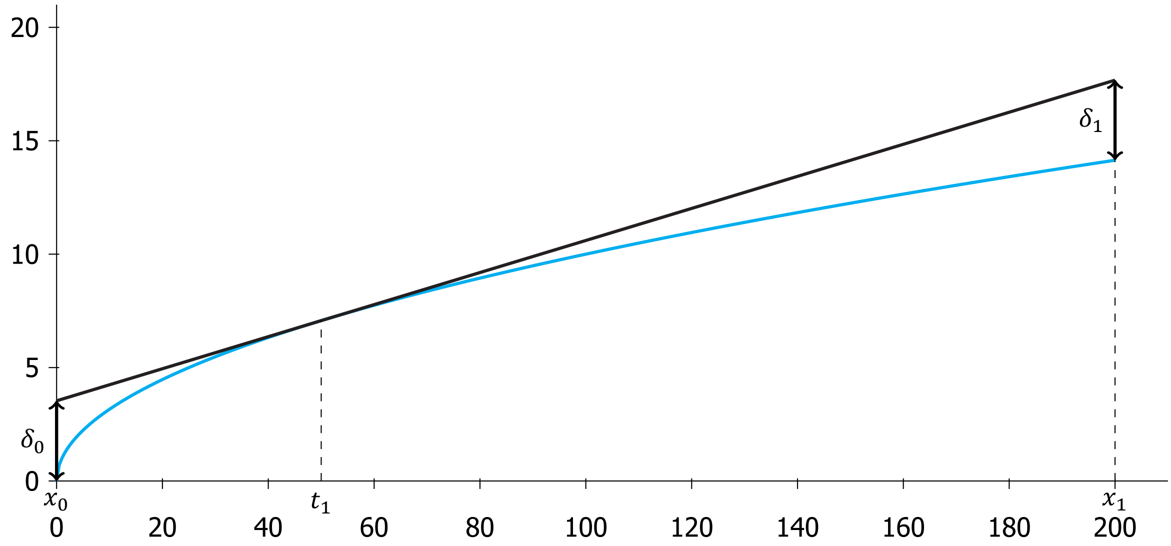

Figure A.1: Approximation of $\sqrt{x}$ on interval $[0, 200]$ by one linear function.

When no surgeries are scheduled on an OR-day, the variance is zero. Therefore, we assume that $x_0 = 0$. Using (A.6) and (A.7) gives

$$x_1 = \left(2\sqrt{t_1} - \sqrt{0}\right)^2 = 4t_1 \tag{A.8}$$

$$t_2 = 4^2 t_1 \tag{A.9}$$

$$x_2 = (2 \cdot 4\sqrt{t_1} - \sqrt{4}\sqrt{t_1})^2 \tag{A.10}$$

$$= \frac{(2 \cdot 4 - \sqrt{4})^2}{4^2} t_2 \tag{A.11}$$

The variable c_n is introduced to denote the ratio between x_n and t_n. When $x_0 = 0$ this ratio is constant. We get the following set of equations:

$$c_1 = 4 \tag{A.12}$$

$$x_n = c_n t_n \qquad \text{for } n = 1, ..., m \tag{A.13}$$

$$t_{n+1} = c_n^2 t_n \qquad \text{for } n = 1, ..., m \tag{A.14}$$

$$c_{n+1} = \frac{(2c_n - \sqrt{c_n})^2}{c_n^2} \qquad \text{for } n = 1, ..., m \tag{A.15}$$

When $x_0 = 0$, $c_1 = 4$. With Equation (A.15) we can compute c_n for all n, see Table A.1.

Number of intervals n	1	2	3	4	5	6	7	8	9	10
c_n	4	2.25	1.78	1.56	1.44	1.36	1.31	1.27	1.23	1.21

Table A.1: c-values per number of intervals n for $x_0 = 0$.

The approximation error for the last breakpoint δ_m can be expressed in terms of x_m and c_m. All approximation errors are equal at the breakpoints, see Equation (A.4). Therefore, we have that $\delta^{\max} = \delta_m$

$$\begin{aligned}\delta^{\max} &= \delta_m \\ &= h_m(x_m) - \sqrt{x_m} \\ &= \sqrt{x_m}\left(\frac{1}{2\sqrt{c_m},} + \frac{\sqrt{c_m}}{2} - 1\right)\end{aligned}$$

This means that the approximation error depends on the maximum value of the interval (x_m) and the number of linear functions (m). Given an interval, we can calculate the approximation error for different numbers of linear functions. For a given maximum approximation error $\Delta^{\max}$ we determine the minimal number of linear functions needed to achieve at most this error by finding the smallest m for which $\delta^{\max} \leq \Delta^{\max}$.

Number of intervals n	1	2	3	4	5	6	7	8	9	10
$\delta^{\max}$	25.00	8.33	4.17	2.50	1.67	1.19	0.89	0.69	0.56	0.45

Table A.2: Approximation errors per number of intervals n with $x_m = 10000$.

For example, let $[0, 10000]$ be the domain of our square root function. In Table A.2, we see the approximation errors for $x_m = 10000$ for different numbers of intervals. Suppose we want at most an overestimation of five minutes, so $\Delta^{\max} = 5$, then we need at least three linear functions. Because $m = 3$ is the smallest value of m for which $\delta^{\max} \leq \Delta^{\max}$.

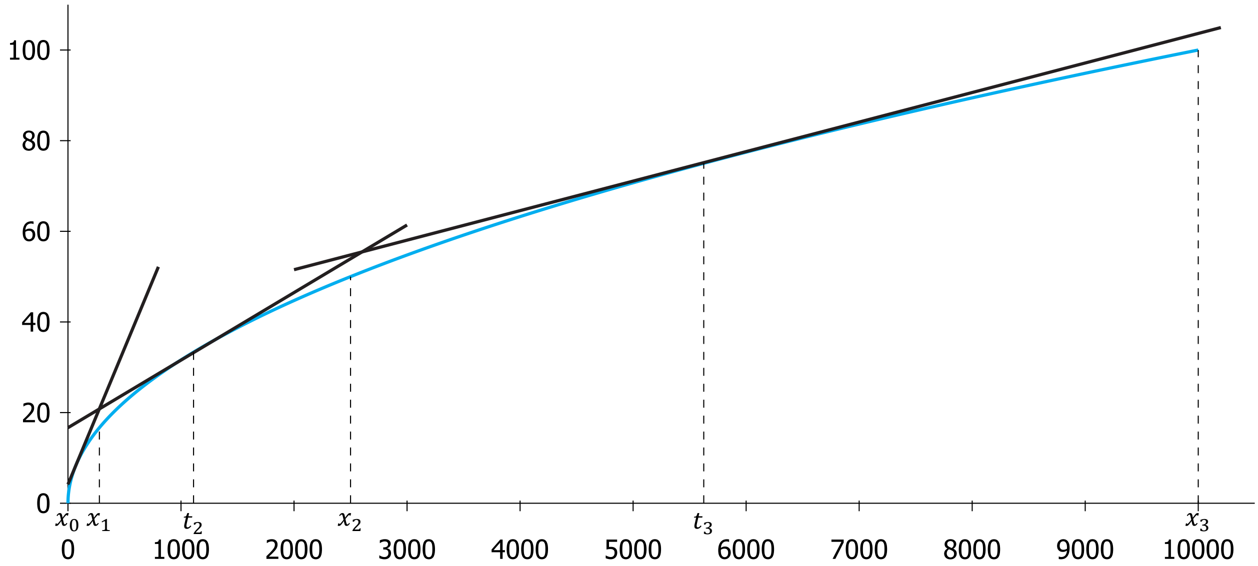

Figure A.2: Linear approximation of the square root function on the interval $[0, 10000]$ with three linear approximation functions.

This approach results in the following plan of attack.

1. Define a maximum value of the interval $[x_{\min}, x_{\max}]$ for which to approximate the square root function.
2. Set a maximum approximation error $\Delta^{\max}$ as the maximum overestimation of the square root function.
3. Compute the c_n-values
4. Calculate the maximum approximation errors δ^{max} for each of the linear functions m.
5. Choose the lowest m-value for which $\delta^{max} \leq \Delta^{max}$
6. Determine t_m
7. Determine the other tangent points t_n for all n.
8. Determine the breakpoints x_n for all n.
9. Determine the breakpoint function values $y_n = h_n(x_n)$

Bibliography

[1] E. W. Hans, M. Houdenhoven, and P. J. H. Hulshof, *Handbook of healthcare system scheduling,* (Springer US, Boston, MA, 2012) Chap. A Framework for Healthcare Planning and Control, pp. 303–320.

[2] J. van Oostrum, M. van Houdenhoven, J. Hurink, E. Hans, G. Wullink, and G. Kazemier, *A master surgical scheduling approach for cyclic scheduling in operating room departments,* Or Spectrum **30**, 355 (2008).

[3] A. Fügener, *An integrated strategic and tactical master surgery scheduling approach with stochastic resource demand,* Journal of Business Logistics **36**, 374 (2015).

[4] OECD, *Health at a glance 2015,* (2015), http://dx.doi.org/10.1787/health_glance-2015-en.

[5] F. Guerriero and R. Guido, *Operational research in the management of the operating theatre: a survey,* Health Care Management Science **14**, 89 (2011).

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

[7] P. Vanberkel, R. Boucherie, E. Hans, J. Hurink, W. van Lent, and W. van Harten, *An exact approach for relating recovering surgical patient workload to the master surgical schedule,* Journal of the Operational Research Society **62**, 1851 (2011).

[8] A. Fügener, E. Hans, R. Kolisch, N. Kortbeek, and P. Vanberkel, *Master surgery scheduling with consideration of multiple downstream units,* European Journal of Operational Research **239**, 227 (2014).

[9] J. Beliën and E. Demeulemeester, *Building cyclic master surgery schedules with leveled resulting bed occupancy,* European Journal of Operational Research **176**, 1185 (2007).

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

[11] J. Beliën, E. Demeulemeester, and B. Cardoen, *A decision support system for cyclic master surgery scheduling with multiple objectives,* Journal of Scheduling **12**, 147 (2009).

[12] Z. Yahia, A. Eltawil, and N. Harraz, *The operating room case-mix problem under uncertainty and nurses capacity constraints,* Health Care Management Science (2015).

[13] P. Vanberkel, R. Boucherie, E. Hans, J. Hurink, W. van Lent, and W. van Harten, *Accounting for inpatient wards when developing master surgical schedules,* Anesthesia and Analgesia **112**, 1472 (2011).

[14] J. van Essen, J. Bosch, E. Hans, M. van Houdenhoven, and J. Hurink, *Reducing the number of required beds by rearranging the or-schedule,* Or Spectrum **36**, 585 (2014).

[15] A. Fügener, G. Edenharter, P. Kiefer, U. Mayr, J. Schiele, F. Steiner, R. Kolisch, and M. Blobner, *Improving intensive care unit and ward utilization by adapting master surgery schedules,* A A Case Rep **6**, 172 (2016).

[16] R. Bekker and P. M. Koeleman, *Scheduling admissions and reducing variability in bed demand,* Health Care Management Science **14**, 237 (2011).

[17] I. Adan, J. Bekkers, N. Dellaert, J. Vissers, and X. Yu, *Patient mix optimisation and stochastic resource requirements: A case study in cardiothoracic surgery planning,* Health Care Management Science **12**, 129 (2009).

[18] I. Adan, J. Bekkers, N. Dellaert, J. Jeunet, and J. 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**, 290 (2011).

[19] D. Min and Y. Yih, *Scheduling elective surgery under uncertainty and downstream capacity constraints,* European Journal of Operational Research **206**, 642 (2010).

[20] C. Banditori, P. Cappanera, and F. Visintin, *A combined optimization-simulation approach to the master surgical scheduling problem,* Ima Journal of Management Mathematics **24**, 155 (2013).

[21] V. Chow, M. Puterman, N. Salehirad, W. Huang, and D. Atkins, *Reducing surgical ward congestion through improved surgical scheduling and uncapacitated simulation,* Production and Operations Management **20**, 418 (2011).

[22] D. Astaraky and J. Patrick, *A simulation based approximate dynamic programming approach to multi-class, multi-resource surgical scheduling,* European Journal of Operational Research **245**, 309 (2015).

[23] B. Cardoen, E. Demeulemeester, and J. Beliën, *Sequencing surgical cases in a day-care environment: An exact branch-and-price approach,* Computers & Operations Research **36**, 2660 (2009).

[24] 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**, 221 (2010).

[25] R. Aringhieri, P. Landa, and E. Tànfani, *Assigning surgery cases to operating rooms: A vns approach for leveling ward beds occupancies,* Electronic Notes in Discrete Mathematics **47**, 173 (2015).

[26] A. Jebali and A. Diabat, *A stochastic model for operating room planning under capacity constraints,* International Journal of Production Research **53**, 7252 (2015).

[27] J. M. van Oostrum, T. Parlevliet, A. P. M. Wagelmans, and G. Kazemier, *A method for clustering surgical cases to allow master surgical scheduling,* Infor **49**, 254 (2011).

[28] J. Ward, *Hierarchical grouping to optimize an objective function,* Journal of the American Statistical Association **58**, 236 (1963).

[29] W. M. van der Aalst, *Process Mining* (Springer, 2011).

[30] E. M. Carter and H. W. Potts, *Predicting length of stay from an electronic patient record system: a primary total knee replacement example,* BMC Medical Informatics and Decision Making **26** (2014).

[31] J. Bosch, *Better utilisation of the OR with less beds*, Master's thesis, University of Twente (2011).

[32] P. Stepaniak, C. Heij, G. Mannaerts, M. de Quelerij, and G. de Vries, *Modeling procedure and surgical times for current procedural terminology-anesthesia-surgeon combinations and evaluation in terms of case-duration prediction and operating room efficiency: a multicenter study.* Anesthesia and Analgesia **109**, 1232 (2009).

[33] E. Hans, G. Wullink, M. van Houdenhoven, and G. Kazemier, *Robust surgery loading,* European Journal of Operational Research **185**, 1038 (2008).

[34] J. Bisschop, *AIMMS optimizating modelling.* (2016).

[35] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity* (Englewood Cliffs, N.J.: Prentice Hall, 1998).

[36] W. Haemers, *Discrete optimization. lecture notes.* (2007).

[37] S. Kirkpatrick, C. Gelatt, and M. Vecchi, *Optimization by simulated annealing,* Science, New Series **220**, 671 (1983).