

THESE DE DOCTORAT DE

L'UNIVERSITE D'ANGERS
COMUE UNIVERSITE BRETAGNE LOIRE

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

Par

Hasan AL HASAN

Surgical case scheduling with medical instruments sterilizing activities constraints

Thèse présentée et soutenue à Learning lab, UCO Angers, le 28/05/2019

Unité de recherche : Laboratoire Angevin de Recherche en Ingénierie des Systèmes (LARIS)

Rapporteurs avant soutenance :

Nathalie SAUER Professeure des universités, Université de Lorraine
Antoine JOUGLET Professeur des universités, Université de technologie de Compiègne

Composition du Jury :

Sylvie NORRE Professeure des universités, IUT d'Allier
Nathalie SAUER Professeure des universités, Université de Lorraine
Antoine JOUGLET Professeur des universités, Université de technologie de Compiègne

Directeur de thèse
David RIVREAU Professeur, Institut de Mathématiques Appliquées d'Angers, UCO

Co-directeur de thèse
Christelle GUÉRET Professeure des universités, Université d'Angers

Co-encadrant de thèse
David LEMOINE Maître-assistant, IMT Atlantique

*This thesis is dedicated to my mother,
my father and my brother.
Without their endless love, support
and encouragement, I would never have
been able to complete my graduate studies.*

*This thesis is also dedicated to my beautiful wife,
who was always there for me through out
many sleepless nights.*

ACKNOWLEDGEMENT

To my life-coaches, my mother and father: because I owe it all to you. Many Thanks!

A very special gratitude goes out to my advisors. Without their support, none of this work would have been possible. I am sincerely grateful to them for putting their trust in me especially during the difficult periods of this research. I thank Christelle Guéret for all her encouragement and invaluable advices on both my research and my career. I also thank her for being available for me whenever I needed some advice and guidance. I am also thankful to David Rivreau for his precious comments on my work and all the valuable input on the research direction. I also thank David Lemoine for the many discussions and debates we had as it has been very instructive to me.

I also want to thank professors Sylvie Norre, Nathalie Sauer and Antoine Jouglet for accepting to be part of the jury.

I would like to express my sincere gratitude to Laurent Hubert, Anne-Valérie Lebel and Antoine Robelet from the CHU of Angers for this enriching collaboration. I am very thankful for them for proposing this challenging problem. I am also grateful to them for taking the time to explain to me the work process at the CHU and the context of the problem. This collaboration and research has been possible due to the financial support from Angers Loire Metropole and the IMT Atlantique. This support is gratefully acknowledged.

Finally, I thank Claudia, Nassib and all my fellow labmates in LARIS for the enriching discussions and the many stress relieving coffee breaks we had.

TABLE OF CONTENTS

| | |
|--|-----------|
| Introduction | 9 |
| I Context, problem description and literature review | 13 |
| 1 The surgical case scheduling problem of the CHU of Angers | 15 |
| 1.1 Introduction | 15 |
| 1.2 The orthopaedic surgery unit | 16 |
| 1.3 The surgical kits | 17 |
| 1.4 The sterilizing unit | 18 |
| 1.5 The work process at the OSU | 19 |
| 1.6 The work process dysfunctions at the OSU | 21 |
| 1.7 CHU objectives | 24 |
| 1.8 The provided data | 25 |
| 1.8.1 OSU's planned schedule analysis | 26 |
| 1.8.2 OSU's achieved schedule analysis | 28 |
| 1.8.3 Difference between planned and achieved schedules | 30 |
| 1.9 Conclusion | 33 |
| 2 Literature review | 35 |
| 2.1 Introduction | 35 |
| 2.2 Strategic level | 37 |
| 2.2.1 Capacity planning | 37 |
| 2.2.2 Capacity allocation | 37 |
| 2.3 Tactical level | 38 |
| 2.4 Operational level | 39 |
| 2.4.1 Advanced scheduling problem | 40 |
| 2.4.2 Allocation scheduling problem | 43 |
| 2.4.3 Advanced and allocation scheduling problem | 45 |
| 2.5 Synthesis and analysis | 49 |
| 2.6 Conclusion | 55 |
| II The Deterministic Surgical case scheduling problem | 57 |
| 3 Deterministic Static Surgical Case Scheduling problem | 59 |
| 3.1 Introduction | 59 |
| 3.2 Formal problem presentation | 60 |
| 3.3 Proof of complexity | 62 |
| 3.4 A MILP formulation | 63 |
| 3.5 Heuristic approach | 67 |
| 3.6 Experimental results | 68 |

| | |
|---|------------|
| 3.6.1 Test Data | 68 |
| 3.6.2 Comparison of planned schedules | 69 |
| 3.7 Conclusion | 73 |
| 4 Deterministic Dynamic Surgical Case Scheduling problem | 75 |
| 4.1 Introduction | 75 |
| 4.2 Technical background | 76 |
| 4.2.1 Dynamic scheduling problem | 76 |
| 4.2.2 Solution approaches | 76 |
| 4.2.3 The rolling horizon approach | 78 |
| 4.3 Formal problem presentation | 80 |
| 4.4 Mathematical formulation | 81 |
| 4.5 Experimental results | 85 |
| 4.6 Conclusion | 90 |
| 5 From planned to achieved schedules | 91 |
| 5.1 Introduction | 91 |
| 5.2 Achieved schedules generating algorithm | 92 |
| 5.3 Deterministic Static SCS problem | 93 |
| 5.3.1 Achieved schedules | 93 |
| 5.3.2 Degradation | 94 |
| 5.4 Deterministic Dynamic SCS | 96 |
| 5.4.1 Achieved schedules | 96 |
| 5.4.2 Degradation | 97 |
| 5.5 Conclusion | 99 |
| III The Non-Deterministic Surgical case scheduling problem | 101 |
| 6 Non-Deterministic Static Surgical Case Scheduling problem | 103 |
| 6.1 Introduction | 103 |
| 6.2 Technical review | 104 |
| 6.2.1 Non-Deterministic Optimisation | 104 |
| 6.2.2 Robust Optimisation | 105 |
| 6.3 First formulation | 108 |
| 6.4 Second formulation | 111 |
| 6.5 Experimental results | 112 |
| 6.5.1 Comparison of planned schedules | 113 |
| 6.5.2 Comparison of achieved schedules | 116 |
| 6.5.3 Results degradation | 118 |
| 6.6 Conclusion | 120 |
| 7 Non-Deterministic Dynamic Surgical Case Scheduling problem | 121 |
| 7.1 Introduction | 121 |
| 7.2 Technical review | 122 |
| 7.3 First formulation | 123 |
| 7.4 Second formulation | 124 |

| | |
|---|-----|
| 7.5 Experimental results | 125 |
| 7.5.1 First robust model results | 125 |
| 7.5.2 Second robust model results | 125 |
| 7.6 Models performance analysis | 132 |
| 7.7 Conclusion | 134 |

General conclusion and perspectives **135**

Bibliography **139**

A Notations **147**

INTRODUCTION

Introduction and problem addressed

We always attribute happiness and wealth to health care as Mahatma Gandhi once said: "It is health that is real wealth and not pieces of gold and silver". But we tend to forget that a good health care system comes with a big price tag. This can be seen clearly in the recent study of the Direction of research, study, and evaluation of statistics in France (DREES 2016) which states that the expenses in the French health care system are evaluated to be equal to 194.6 billion euros, which is equal to 8.9% of the national Gross Domestic Product (GDP). In addition, these costs are expected to increase in the future due to the population ageing and population explosion, the scarcity of resources and the increasing demand for a better health care. Therefore, hospitals are rapidly paying more and more attention to the importance of costs saving methods.

One of the most expensive and complex functional units within the hospital is the operating theatre [Denton et al., 2007, Association et al., 2003]. It has been recognized as a main source of income as it generates around two thirds of hospital revenues [Jackson, 2002], while also counting for around 40% of hospital costs [Macario et al., 1995] throughout the use of facilities (operating rooms, intensive care beds, etc.) and the personnel costs (surgeons, nurses, anaesthetists, etc.). In addition, the operating theatre provides services for many specialities which makes it a possible bottleneck. Thus, it introduces a need for careful resource planning. In this context, hospitals face an increasing pressure for high quality care and cost effectiveness.

Recent studies have shown that most of the costs of surgical procedures consist of personnel, infrastructure, equipment, logistics and administrative support expenses [Roland et al., 2006] which have driven the need for efficient resource usage. At the same time, Operating Room (OR) planning and scheduling is challenging. Firstly, multiple stakeholders with conflicting interests are involved [Glouberman and Mintzberg, 2001] such as hospital managers, OR personnel, surgeons of various specialities, and patients. Secondly, OR surgical scheduling is complicated due to the big uncertainty regarding the occurrence and duration of surgeries. For example, the arrival of semi-elective patients may disrupt the planned scheduling throughout the day. In addition, the complications that can happen in surgeries because of their unpredictable nature also cause modifications to fixed schedules. Lastly, the OR departments are facing conflicting performance criteria: allocating less time for surgeries to evade the staff overtime may lead to excessive patient waiting, while allocating more time for surgeries to decrease the patients waiting could give rise to staff overtime. This problem has thus attracted the attention of many researchers (e.g. [Cardoen et al., 2010, Guerriero and Guido, 2011]).

Another big factor in OR planning is the limited nature of the resources that are shared among the specialities. These resources include the ORs and the medical instruments. The fact that these limited numbers of medical instruments must be sterilized after each surgery in order to be ready for the next usage, creates a possible bottleneck. In addition, the sterilizing

unit is often a shared unit in hospitals among all the units, thus a good resource management system is highly needed as poor implementation in one unit can lead to high utilization of the sterilizing unit which affects the performance of the whole hospital.

In this thesis, we will be working with the Centre Hospitalier Universitaire d'Angers (CHU) of Angers in France. We focus on the surgery scheduling problem at the orthopaedic surgery unit, a problem formally called: Surgical Case Scheduling (SCS). In our problem, we will consider the activities of the sterilizing unit with the objectives of minimizing the operating costs and maximize the resource utilization while keeping a good level of service quality.

Research questions

The SCS problem proved to be a challenging one with a big interest to the hospital management. The main research question of this thesis is:

How can we use operations research techniques to improve the resources utilizations and minimize the costs at the surgery unit of the CHU?

In order to tackle this problem, we chose to break it down to sub-problems where we address a new aspect of the problem in each question:

RQ1: How can we use computer-based modelling techniques to fully represent and understand the problem?

RQ2: Can we achieve better results by minimizing the operational costs and improving the resources utilization when applying the operations research techniques in comparison to the real results achieved at the CHU, when only considering the simple (static) version of the surgical case scheduling problem?

RQ3: Can we improve our results by generating more robust solutions that are less distorted by the disturbances and the uncertainties in the data?

RQ4: Can we implement our model and method in a real-life work flow?

Organization of the dissertation

This thesis is focused on the surgical case scheduling problem in order to support the decision makers at the CHU. It is organized as follows:

The first part is devoted to the presentation of the SCS problem of the CHU and its methodological details.

In Chapter 1, we describe the concerned elements and work process at the CHU. We explain the day to day operations at the two involved units namely the orthopaedic surgery unit and the sterilisation unit. Later, we describe the test data that were given to us by the CHU and analyse it in depth.

In Chapter 2, we review what other researchers have already done and give an intensive literature review for the operating room management and scheduling problem.

The second part of this thesis is devoted to solving the deterministic version of the SCS problem, where we disregard the uncertainties presented in surgeries durations.

In Chapter 3, we tackle the deterministic static version of the SCS, where we consider that the duration of the surgeries are known in advance. First, we prove that the problem is NP-hard. Next, we propose a Mixed Integer Linear Programming (MILP). Experimental results are shown next and evaluated using the operational costs metric. Finally, we provide a detailed comparison of our results with the results of the CHU.

In Chapter 4, we consider the deterministic dynamic version of the problem. We start by presenting a technical background for deterministic dynamic scheduling problems. Next, we adapt the MILP presented in Chapter 3 to solve the dynamic (online) version of the problem, where day to day consulted surgeries are considered. In addition, we consider the patients delay factor to evaluate our method in addition to the cost factors. We then implement a rolling horizon method to solve the problem. Finally, experimental results are shown and compared to the ones of the CHU to evaluate our approach.

In Chapter 5, we discuss the uncertainties found in surgeries durations and show their effect on our results from Chapters 3 and 4. We evaluate the degradation in our results in both chapters and compare it to the ones of the CHU.

The third part of this thesis is devoted to solving the non-deterministic version of the surgical case scheduling problem, where we consider the stochastic nature of surgeries durations.

In Chapter 6, we tackle the static non-deterministic SCS problem. We start by providing a technical background for the two common approaches to deal with uncertainties namely stochastic programming and robust optimization. Next, we present two robust formulation for the non-deterministic static SCS problem. Finally, we present our experimental results and compare it to the ones of the CHU and compare the degradation in solution qualities found in the obtained results with the ones from the deterministic version and the ones of the CHU.

In Chapter 7, we consider the dynamic non-deterministic version of the problem. We start by adapting the rolling horizon method presented in Chapter 4 to the robust formulations presented in Chapter 6. Next, we present our experimental results for both models and compare it to the ones of the CHU. Finally, we analyse the degradation in solution qualities found in the obtained results and compare it with the ones from the deterministic version and the ones of the CHU.

Finally, we conclude and provide various perspectives.

PART I

**Context, problem description and
literature review**

THE SURGICAL CASE SCHEDULING PROBLEM OF THE CHU OF ANGERS

1.1 Introduction

The Centre Hospitalier Universitaire d'Angers (CHU) is a French university hospital centre employing approximately 6300 hospital staff including 1133 medical and pharmaceutical staff. Each year, the CHU receives around 90,000 emergency room visits, 430,000 outpatient consultations and 100,000 hospitalizations.

The CHU consists of several surgery units and a sterilizing unit. Following the requirements of the CHU, we will focus in this thesis on the Orthopaedic Surgery Unit (OSU) and the Sterilization Unit (SU). In this chapter, we will start by describing the problem components and the work process at the OSU and SU. Since the SU is providing its services for all the other departments including the OSU, we will focus only on the important aspects that are linked directly to our problem. Next, we will show the problems faced by the CHU with their current work flow. Finally, we will describe and analyse the real data that we received from the CHU and conclude with this analysis.

1.2 The orthopaedic surgery unit

Surgical practices can be categorized in two main categories, namely elective and non-elective surgeries. Elective surgeries group urgent surgeries that cannot be scheduled in advance, while non-elective surgeries group surgeries that can be scheduled in advance. The OSU deals exclusively with non-elective surgeries. They perform 211 types of surgeries with a total of approximately 2500 surgeries a year.

Two major patient classes are considered at the OSU, namely outpatient and inpatient (see Figure 1.1). The first class groups patients who are hospitalized for less than 24 hours with no overnight stay (ambulatory care), whereas the second class represents the patients that receive longer care. All outpatient surgeries must finish before 15:00 as outpatients need to spend at least 2 hours in the recovery room after the surgery before being discharged and the recovery room closes at 17:00. Contrarily, inpatient surgeries can finish up to when the operating room closes because inpatients are transferred to the Intensive Care Unit (ICU) after the surgery and the ICU is open 24 hours a day.

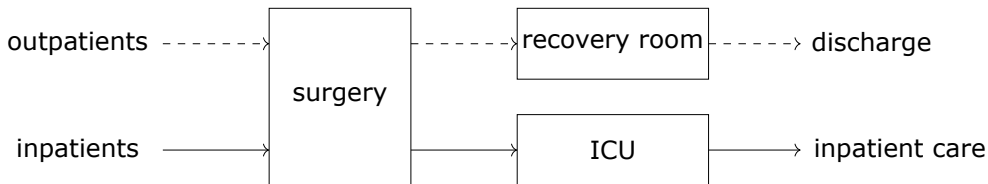


Figure 1.1: The 2 patient classes

There are 3 operating rooms and 15 surgeons at the OSU. Each surgeon is assigned to a room for a full day where he can arrange his surgical cases. This assignment is indicated in the Work Shift Schedule (WSS) which is generated 6 months in advance. The WSS schedule also indicates the ambulatory shifts which are reserved for the outpatients only and any surgeon can operate in these shifts as long as the case is outpatient. When the WSS is approved by the surgeons, it can only be changed under certain circumstances such as the absence of a surgeon or when a surgeon quits the job. The reservation time is a full working day which means that the surgeon has a room just for his/her operations for the full working hours of that room. The working hours of the 3 operating rooms are given in Table 1.1. Note that some rooms can be closed on some days for technical maintenance reasons

| Operating room | Days per week | Opening time | Surgeries starting time | Closing time |
|----------------|----------------|--------------|-------------------------|--------------|
| 1 & 2 | M, T, W, Th, F | 8:00 | 8:15 | 17:00 |
| 3 | M, T, W, F | 8:00 | 8:15 | 14:30 |

Table 1.1: Operating rooms working hours

The difference in working hours and days between the first two rooms and the third one is due to the shortage of staff (for any operation, they require 3 general nurses and 1 anaesthetist). Generally, there are a total number of 14 available room day (shift) each week, where 12 of these shifts are distributed between the surgeons and 2 are marked as ambulatory

shifts. The ambulatory shifts can be used by any surgeon and are always reserved at room 3 since the closing hour of this room matches the condition of the outpatient surgeries (they have to finish before 15:00). Due to the large number of surgeries to perform, an operating room may stay open exceptionally after closing hours for up to 3 hours. In addition, the CHU imposes an average occupancy rate of the operating rooms of 80% to each of its surgery units because of the substantial fixed costs induced by opening an OR.

1.3 The surgical kits

The instruments, medical devices and tools that are used for surgeries are organized and packed in small boxes that are called kits (see Figure 1.2). Each kit contains tools that are used usually together for specific types of operations. A surgery can use more than one kit, and each kit can be used for more than one type of operation. When a kit is not being used, it's stored in the store room where it can stay up to 6 months (after 6 months, it must be re-sterilized before being used again).



Figure 1.2: Surgical kit example. Downloaded from <http://jkinstrumente.com> in June 2018

There are 212 different types of kits at the OSU that are available in limited quantities (between 1 and 15 kits for each kit type, with a total of 482 available kits at the OSU). In addition to these available kits, there are some kits that are too expensive and rarely used at the OSU, which does not justify the investment in buying and stocking them. These expensive kits are used around once every 6 months and the OSU outsources these kits from other pharmacies. This procedure can cause a big problem if the date of the surgery is changed after the outsource request was made.

Two additional types of tools and instruments are available in the OSU:

- The backup tools, which are used when an instrument from a kit breaks down or falls on the ground (they use one of these instead of opening one full kit just for one instrument).
- The unique use tools (e.g. implanted medical devices).

1.4 The sterilizing unit

The Sterilization Unit (SU) opens from Monday to Friday at 07:00 and closes at 20:30. It centralizes the sterilizing process and provides its services to all the units at the CHU.

The staff at the SU processes the received kits from the different units in a First In First Out (FIFO) order. They start by opening the kits and preparing the instruments (e.g. they open the scissors to be washed sufficiently) and they place the instruments back in the kit and deposit the kits on a metal holder that can take around 20 kits. Finally, they place the holder on the automatic feed conveyor belt for the washing machine with the minimum load. This stage takes around 15 minutes.

All the instruments can be washed in the washing machines safely except the electrical devices (e.g. motors) which must be washed by hand. The SU has 2 types of washing machines (2 machines of each type):

- The first type is a machine that is divided into 3 stages as shown in the next figure:

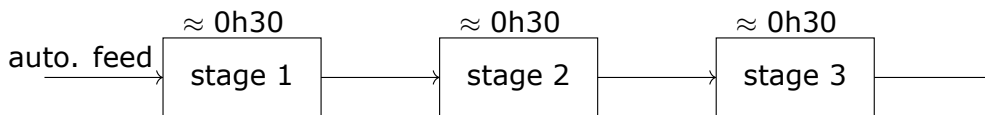


Figure 1.3: Washing machine type 1.

The total process time for the 3 stages in these machines is around 1 hour and half. When the machine finishes processing the first stage on a holder, it moves the holder to stage 2 and takes a new holder at stage 1. This makes the waiting time for a holder to around 30 minutes.

- The second type has only 1 stage as shown in the next figure:

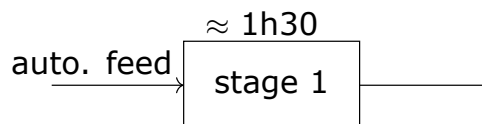


Figure 1.4: Washing machine type 2.

In these machines, once a holder enters, the next one should wait around 1 hour and half to start its turn.

After the holder leaves the washing machines, the repacking stage starts. The staff checks if the kits in the holder are washed properly (if not, they send them back to the washing stage). Then, a nurse from each block gets the kits for her block. The nurse then prepares the instruments (e.g. close back the scissors) and then packs the instruments into their kits and put them on a conveyor belt. It is hard to determine the exact time needed for this stage since it depends on many factors like the availability of the nurses and the load of each basket, but on average this step takes approximately 15 minutes. At the end of the belt, the staff takes

the kits and wraps each one and puts a label on them (see Figure 1.5). Finally, the wrapped kits are placed in a metal basket and sent to the sterilization machines.

The last step is the sterilization stage. In the SU, they have 4 identical sterilizing machines (autoclaves). Each machine takes 2 baskets at a time and each basket carries between 10 and 30 kits (it depends on the size of the kits). After the machine finishes, they take the baskets and leave them to cool off. The next thing is to check if the kits were sterilized properly (if not, they are sent back to be re-sterilized). Finally, they start by packing the kits for each block in a green marked container where it waits for its pickup. The total time for the sterilizing machines is around 1h30 and the rest takes about 1h00 thus this step takes around 2h30.



Figure 1.5: Wrapped kits example. Downloaded from <https://shadeguide.com.au/> in June 2018

On average, when a kit arrives at the SU, it needs around 4h30 to be ready for pickup. The whole sterilizing process is summarized in Figure 1.6. Moreover, it is important to note that the SU does not allow the sterilising process of a kit to start on a day t and end on the next day $t + 1$. In other words, if there is not enough time to sterilise a kit on a day, then the staff at the SU will delay the sterilising process of this kit to the next day. Thus, the deadline for the last batch that enters the sterilising process at any day is 16:00.

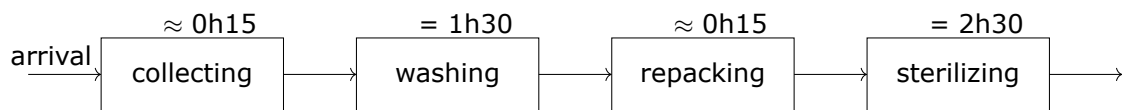


Figure 1.6: Kits sterilization 4 stages at the SU.

1.5 The work process at the OSU

There are 9 secretaries at the OSU and each surgeon has one secretary to make his/her appointments. The process starts usually between 3 weeks to 3 months before the date of the surgery when the patient comes to see the surgeon for the consultation. At this stage, the surgeon decides if the patient needs a surgery or not. If the surgeon decides for the surgical procedure, he defines during consultation the exact date of the surgery depending on many factors (e.g. the surgeon’s load, the type of the surgery, the urgency level, .. etc.), the type of

this surgery (ambulatory or not), the required kits for the surgery and the quantity of each kit type and finally, the surgeon specifies an estimated duration for the surgery. If the surgery is for an outpatient case then the surgeon can chose to assign it to the date of one of his normal shifts or to an ambulatory shift.

The estimated duration of a surgery is automatically calculated by the system as the average duration of surgeries from the same type that were performed by the surgeon. Surgeons can override the proposed estimated duration if needed to take into account the specificities of the case like patient’s age or medical history.

Next, the secretary creates the file of the patient in the system and fills it with the data from the consultation document. In this step the secretary only specifies the date and the duration of the operation, she neither specifies the order of the operations at that date nor the starting and finishing times for each operation. The system shows her only the shifts for her surgeon and the ambulatory shifts (if it was an outpatient case). The system will colour the available shifts with a certain colour to show the remaining time of the shift after adding the duration of the current surgery. These colours are:

- **Green:** More than 50% of the shift’s time is available.
- **Yellow:** Between 50% and 20% of the shift’s time is available.
- **Red:** Less than 20% of the shift’s time is available.
- **Black:** The shift is full.

Each Monday, one specialized nurse prepares the schedule for the next week surgeries. She specifies the order and the start and finish time of each operation using their estimated durations, while taking into account that there are limited number of kits. If she finds that for a certain day there are more scheduled operations that uses the same type of kits than the total available number of these kits, she will choose to cancel the surgery with the least priority. In this case she calls the secretary and the surgeon to inform them about the cancellation and the secretary will reschedule the patient again on the nearest possible time. The weeks schedule also assigns for each surgery, 3 general nurses and 1 anaesthetist.

Every day, the staff in the OSU picks the required kits for the next day’s surgeries from the store room. This preparation happens at 2:30pm. The choice of this hour gives the SU enough time to prepare and send the missing kits for the next day’s surgeries. If some kits are missing, they send a fax to the SU and then wait for their response. The SU will then check if they received the kits and if so, they will tell the OSU at which stage exactly the kits are. If the missing kits were still at the OSU, they put the green label on it (high priority) and prepare it to be sent urgently to the SU by a special shuttle. Figure 1.7 summarizes the work process at the OSU from the consultation date till the date of the surgery.

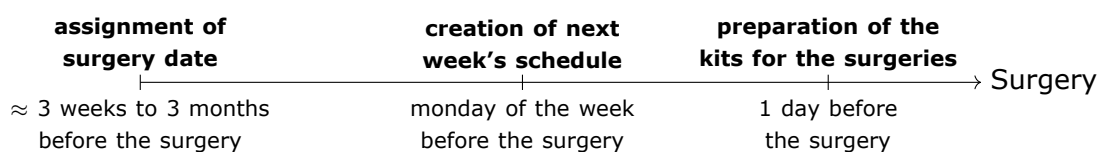


Figure 1.7: The work process at the OSU.

Finally, each day, the team at the OSU performs the surgeries that are specified in their planned schedule. 20 minutes are necessary between each two consecutive surgeries to clean the room. In most cases, surgeries estimated durations differ from the actual real ones.

After each surgery, the kits that were used are kept in water for 30 minutes for pre-disinfection, then the staff collects these used kits and store them in a red marked container to be picked up and sent to the SU at the next pickup hour, while the kits that were not used are sent back to the store room. Since the SU provides its services to all units at the CHU, they have a timetable that defines the kits pickup and delivery hours to each unit in order to distribute the load. In the case of the OSU, these hours are defined in Table 1.2.

| | | | | | | | |
|-----------------|-------|-------|-------|-------|-------|-------|-------|
| pickup | 06:50 | 11:30 | 13:00 | 14:30 | 16:00 | 17:30 | 18:30 |
| delivery | 06:50 | - | - | 14:30 | - | 17:30 | - |

Table 1.2: SU's pickups and deliveries from the OSB

During the transport, kits are stored in a metal container. The containers that are marked with a red symbol are for the used kits and the containers that are marked with a green symbol are for the sterilized kits. A driver collects the red marked containers at the OSU and brings them to the SU. The driver makes sure that there is a container at the OSU at any time by leaving an empty one when he picks up the red marked ones from the unit. Finally, the driver drops the containers that he picked up at a marked place at the SU. A summary of OSU kits movement is illustrated in Figure 1.8.

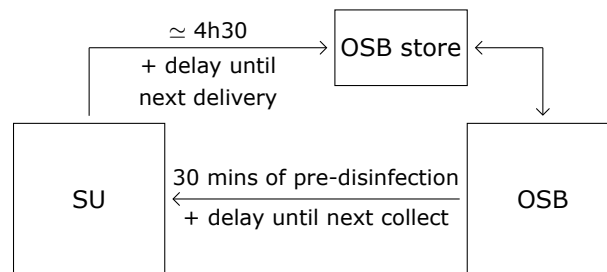


Figure 1.8: OSU kits movement at the CHU.

1.6 The work process dysfunctions at the OSU

The main problem faced by the CHU is the overtime. This problem can be attributed to the lack of global planning as each surgeon defines the date of the surgery during the consultation based on his/her shifts, and the condition and availability of the patient, without considering the schedule of the other surgeons. Indeed, there are 28 hours of overtime on average per month at the OSU with around 10 ambulatory surgeries that finish after the 15:00 deadline on average per month.

The second problem of this work flow is that the major kits check happens at the end of each week by a nurse that verifies that there are no kit constraint violations in the next

week's schedule. The main check consists in verifying that the surgeries scheduled each day are compatible with the number of kits owned by the OSU, without considering surgeries from the past day. Despite the effort of increasing the working hours of the SU, kits are regularly treated urgently there.

Indeed, since the sterilization process takes approximately 4 hours and 30 minutes, and taking into account the pickup and delivery hours, in a "normal" situation a delay of at least 7 hours is required between the collecting of a kit at the OSU, and its sterilisation for another operation. For example, a kit used for a surgery S_1 during the morning and collected at 14:30 can be used again the next day from 14:30 for a surgery S_2 (see Figure 1.9).

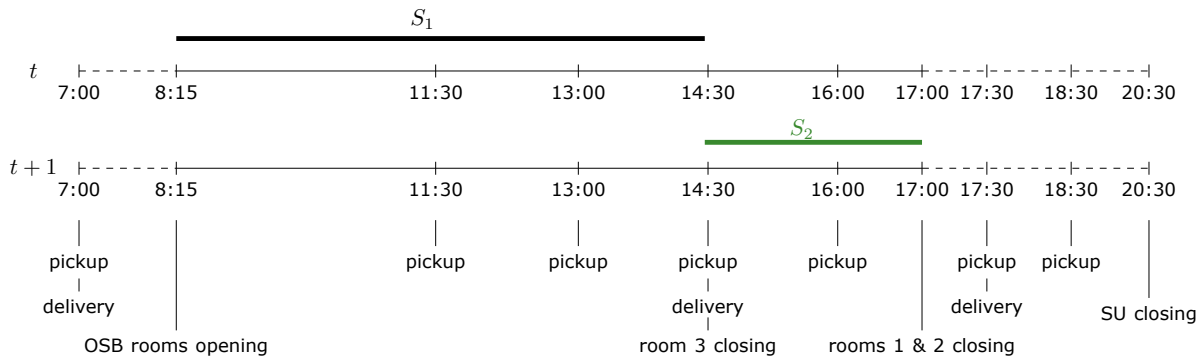


Figure 1.9: Normal kit example.

If these delays are not respected, the kits are then considered as urgent or priority kits depending on the case, as explained below:

- Priority kits

If a kit arrives at the SU with between 5-7 hours to be re-sterilized before the need to use it again, then the SU has to treat it as a priority kit. The kit is then added to the start of the queue (high priority). This situation can happen in two cases:

1. **Priority kit type 1:** if the kit is collected after the first surgery (S_1) by the shuttle of 14:30 on day (t) at most, and must be delivered by the shuttle of 7:00 on day ($t+1$) for a surgery (S_2) planned between 8:15 and 14:30, as shown in Figure 1.10.

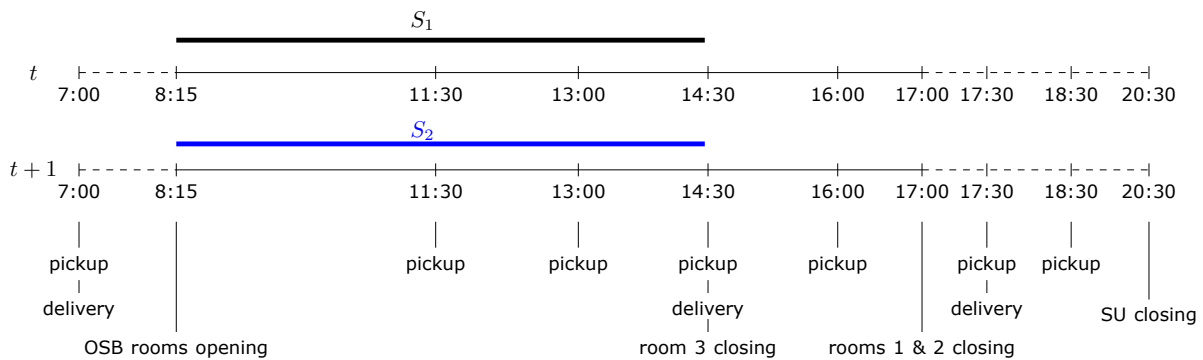


Figure 1.10: Priority kit (type 1) example.

2. **Priority kit type 2:** if the kit is collected after the first surgery (S_1) by the shuttle of 17:30 on day (t), and must be delivered by the shuttle of 14:30 on day ($t + 1$) for a surgery (S_2) planned between 14:30 and 17:00, as shown in Figure 1.11.

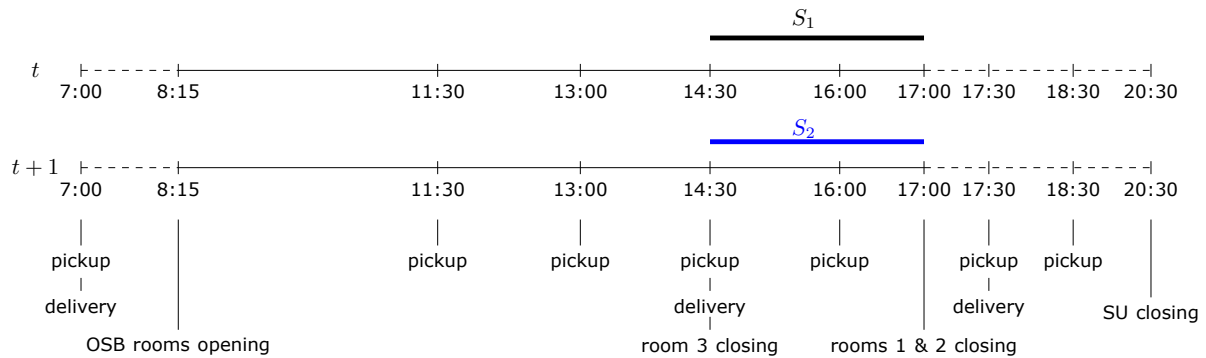


Figure 1.11: Priority kit (type 2) example.

- Urgent kits

If a kit arrives at the SU with barely enough time to be re-sterilized before the need to use it again (less than 5 hours), then the SU will treat it as an urgent kit (highest priority). In order to treat an urgent kit, the staff of the SU puts the kits at the start of the queue, and eventually delays the start of washing machines waiting for the arrival of the urgent kit. This situation happens if the kit is collected after a surgery (S_1) by the shuttle of 16:00 on day (t) at most, and must be delivered by the shuttle of 7:00 on day ($t + 1$) for a surgery (S_2) starting between 8:15 and 14:30, as shown in Figure 1.12.

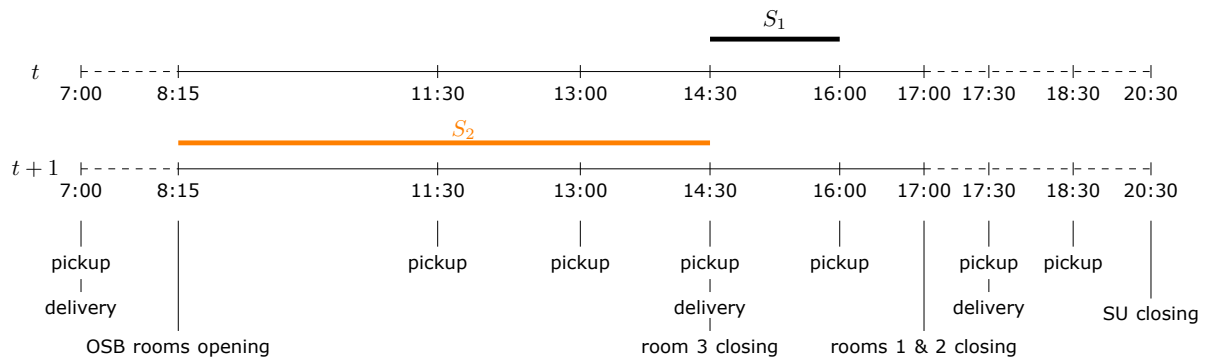


Figure 1.12: Urgent kit example.

Other emergencies may arise when the delay between the 2 surgeries that will use the same kit is large enough for the sterilizing process ($> 4h30$), but the fixed pickup and delivery hours does not allow for the kit to be delivered to the SU for sterilization and then delivered back to the OSU before the next planned usage. This situation can happen in two cases: if the two surgeries are planned in the same day t , or if the two surgeries are planned during two consecutive working days t and $t + 1$. An example of the first case is illustrated in Figure 1.13.

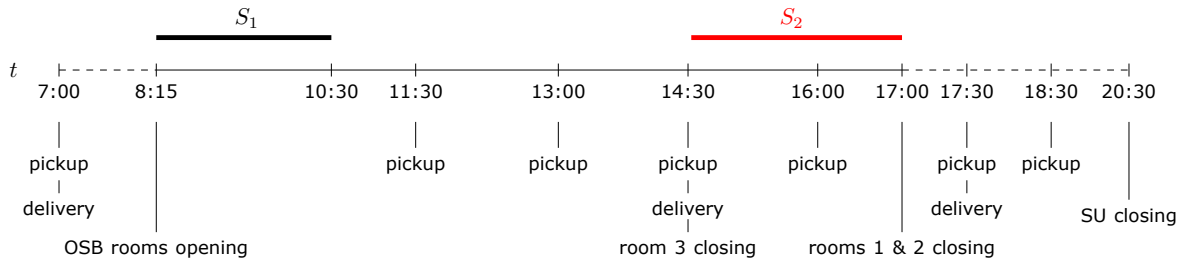


Figure 1.13: Not allowed kit example 1.

Figure 1.14 gives an example of the second case.

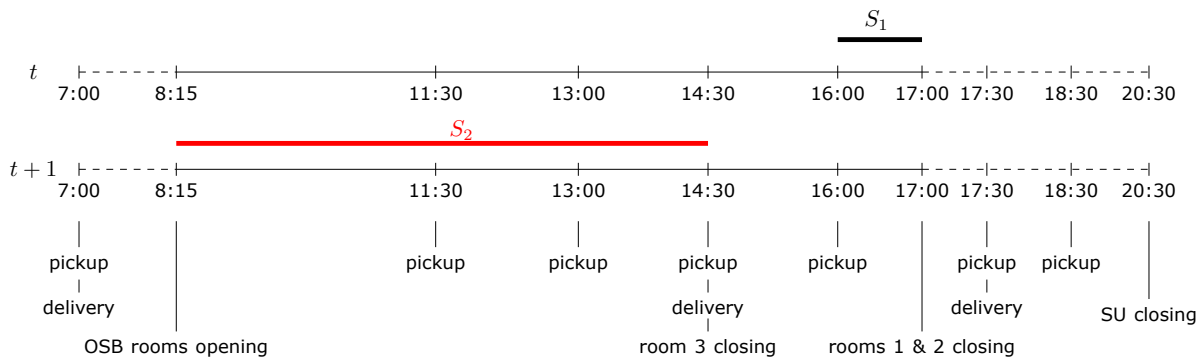


Figure 1.14: Not allowed kit example 2.

Even if these cases must be strictly avoided, they can occur as the SU can send a special shuttle outside the collecting hours just to collect these kits so that there will be enough time to sterilize them before sending them back possibly by another special shuttle outside the delivery hours. We will refer to these kits by **“Not allowed kits”**.

In the current state at the CHU, priority, urgent and not allowed kits are a common problem. They treat an average of 50 priority kits, 8 urgent kits and 6 not allowed kits per month. Priority kits have negative impact on the work flow at the different units at the CHU as they take the place of other kits at the start of the queue and cause delays to the other kits from other units. The impact of urgent kits is even bigger as they generate a lot of stress for the staff of the SU, in addition to the delays they cause to the other kits from other units. Finally, the not allowed kits have the worst impact as in addition to the stress caused at the SU, they generate more delays by reserving a special shuttle that would be normally delivering and picking up kits from other units.

1.7 CHU objectives

In light of the problems faced at the CHU that we identified in the previous section, the goal of the CHU’s management is to minimize the operational costs while maintaining the same level of quality of service represented by the total number of operated patients per month at the OSU.

The first objective is to identify the potential costs reductions that can be achieved using operations research techniques. This study will help toward the main goal, which is to present a complete framework that automates the whole patients scheduling process from the consultation till the surgery date.

Using a unified scheduling procedure for all the surgeons at the OSU will eliminate the lack of global planning problem currently faced, while also minimizing the operating costs and the inconvenience in the SU. The costs considered are represented by the overtime at the OSU and the costs for opening operating rooms. In the actual context of budgetary restriction and of hard working conditions for the hospital staff, the objective for the CHU is to schedule all surgeries in order to minimize the total overtime of the staff members of the OSB, thus reducing costs while taking into account the quality of work life of the staff. The second objective consists in minimizing the number of used operating rooms, as opening an OR generates substantial costs. Finally, the third objective is to keep the number of urgent and priority kits as low as possible. The minimization of these objectives has to be achieved while maintaining the same current level and quality of service represented by the total number of operated surgeries per month at the block.

1.8 The provided data

The CHU provided us with real data that corresponds to the activity of the OSU during the period from September 2014 till June 2015. Table 1.3 shows the characteristics of the received data where the '*#Working days*' column represents the number of working days for the OSU at the given month. On average, the OSU works 21 days a month. Similarly, the '*#Surgeries*' column shows the number of surgeries performed at the OSU: in average, 200 surgeries are performed each month. Finally, the '*#Surgeons*' column shows the total number of surgeons: on average, 13 surgeons were active each month. In addition to the 2000 surgeries found in the instances, 69 more OSU surgeries were performed outside the unit during the 10 concerned months. We however had to consider these surgeries due to their impact on the numbers of kits. In order to do so, we created a fourth room that is exclusive to these surgeries performed outside the unit and fixed the dates and times for these surgeries as in the original schedule in this new room. Furthermore, 38 surgeries from other units were performed at the OSU. Again, we had to consider these surgeries due to their impact on the total operating time at the ORs, and we achieved this by decreasing the total operating hours of the corresponding ORs at the dates of these surgeries by the duration of each of these surgeries.

In order to better understand and analyse the provided data, we studied the following aspects:

1. OSU's planned schedule analysis.
2. OSU's achieved schedule analysis.
3. Difference between planned and achieved schedules.

In the reminder of this section, we will show and analyse each of these studies in order to give a better understanding of the problems faced at the OSU.

| Month | #Working days | #Surgeries | #Surgeons |
|----------------|---------------|------------|-----------|
| Sept. | 22 | 218 | 14 |
| Oct. | 23 | 212 | 13 |
| Nov. | 20 | 189 | 14 |
| Dec. | 23 | 189 | 13 |
| Jan. | 21 | 220 | 13 |
| Feb. | 20 | 199 | 11 |
| March. | 22 | 223 | 12 |
| April | 21 | 188 | 12 |
| May | 17 | 165 | 10 |
| June | 22 | 197 | 13 |
| Average | 21 | 207 | 13 |

Table 1.3: OSU’s provided data characteristics.

1.8.1 OSU’s planned schedule analysis

As explained before in the work process of the CHU section, the staff at the OSU plans the surgeries using their estimated durations and the schedule that they generate is called the planned schedule. Table 1.4 shows the violated constraints in the planned schedule of the CHU during the course of the 10 months of the provided instances. We can see that there are approximately 7.6 late ambulatory surgeries on average each month as shown in the ‘#Late amb.’ column. In addition, the ‘#Not allowed kits’ column shows the number of the not allowed kits that the SU had to treat each month. In average, the OSU had approximately 10.2 not allowed kits per month.

| Month | #Late amb. | #Not allowed kits |
|----------------|------------|-------------------|
| Sept. | 11 | 5 |
| Oct. | 9 | 7 |
| Nov. | 5 | 10 |
| Dec. | 9 | 18 |
| Jan. | 12 | 12 |
| Feb. | 7 | 5 |
| March | 4 | 25 |
| April | 5 | 9 |
| May | 8 | 7 |
| June | 6 | 4 |
| Average | 7.6 | 10.2 |

Table 1.4: Violated constraints in the OSU’s planned schedule.

Next, table 1.5 shows the overtime in the planned schedule of the OSU. As shown in column ‘Overtime (in min)’ which shows the total overtime in each month in minutes, the average overtime is approximately 804.1 minutes (more than 13 hours) per month. In addition, the ‘max’ column shows the maximum overtime in minutes found in a day during the considered

month. We can see that the rule of maximum of 3 hours (180 minutes) of overtime per room per day is violated in 2 months (February and June).

| Month | Overtime (in min) | max |
|----------------|--------------------------|------------|
| Sept. | 590 | 131 |
| Oct. | 714 | 176 |
| Nov. | 502 | 81 |
| Dec. | 656 | 77 |
| Jan. | 1159 | 120 |
| Feb. | 1278 | 183 |
| March | 641 | 109 |
| April | 561 | 65 |
| May | 870 | 105 |
| June | 1070 | 266 |
| Average | 804.1 | |

Table 1.5: OSU’s planned schedule overtime.

Table 1.6 states the number of opened rooms each month in the OSU’s planned schedule. In the ‘#opened rooms’ column, we can see that the average number of opened rooms is 53.9 per month. Column ‘Avg. Occ. rate’ shows the average occupancy rate in each month: in average, the OSU had approximately 80.9% occupation rate per month with 3 months dropping below the 80% occupancy rate target specified by the management of the CHU. Moreover as shown in column ‘Min. Occ. rate’ which contains the minimum occupancy rate found in each month, several rooms are not used efficiently: only one month had a minimum rate greater than 50% with 4 months having less than 20%.

| Month | #opened rooms | Min Occ. rate | Avg. Occ. rate |
|----------------|----------------------|----------------------|-----------------------|
| Sept. | 59 | 27.6% | 80.2% |
| Oct. | 59 | 13.8% | 77.5% |
| Nov. | 48 | 18.3% | 80.7% |
| Dec. | 48 | 42.4% | 86.5% |
| Jan. | 59 | 28.8% | 85.1% |
| Feb. | 52 | 51.2% | 83.3% |
| March | 59 | 34.3% | 81.5% |
| April | 49 | 26.1% | 82.9% |
| May | 46 | 11.6% | 79.4% |
| June | 60 | 9.5% | 72.3% |
| Average | 53.9 | | 80.9% |

Table 1.6: OSU’s planned schedule opened rooms.

Finally, table 1.7 states the number of urgent and priority kits (both cases) that are found in the OSU’s planned schedule. On average, the OSU had approximately 6.6 urgent kits and 62.6 priority kits to process per month with a total of 69.2 problem kits (urgent + priority) per month.

| Month | #urgent | #priority |
|----------------|----------------|------------------|
| Sept. | 2 | 51 |
| Oct. | 2 | 56 |
| Nov. | 8 | 68 |
| Dec. | 13 | 82 |
| Jan. | 6 | 68 |
| Feb. | 2 | 59 |
| March | 28 | 98 |
| April | 0 | 56 |
| May | 2 | 53 |
| June | 3 | 35 |
| Average | 6.6 | 62.6 |

Table 1.7: OSU’s planned schedule urgent and priority kits.

1.8.2 OSU’s achieved schedule analysis

We analyse now the achieved schedule of the OSU which represents what actually happened after performing the surgeries and replacing the estimated durations of the surgeries with the actual (real) ones. Following the same order as in the planned schedule analysis, we start with Table 1.8 which compares the violated constraints found each month. On average, the OSU had approximately 10.1 ambulatory surgeries that finish after 15:00 and 6.2 not allowed kits per month.

| Month | #Late amb. | #Not allowed kits |
|----------------|-------------------|--------------------------|
| Sept. | 14 | 5 |
| Oct. | 14 | 5 |
| Nov. | 6 | 10 |
| Dec. | 10 | 8 |
| Jan. | 14 | 5 |
| Feb. | 8 | 4 |
| March | 6 | 13 |
| April | 9 | 2 |
| May | 10 | 7 |
| June | 10 | 3 |
| Average | 10.1 | 6.2 |

Table 1.8: Violated constraints in the OSU’s achieved schedule.

Next, table 1.9 shows the overtime in the achieved schedule of the OSU. The average overtime is approximately 1717.4 minutes (almost 29 hours) per month. In addition, the rule of maximum of 3 hours (180 minutes) of overtime per room per day is violated in 4 months (September, October, February and June).

| Month | Overtime (in min) | max |
|----------------|-------------------|-----|
| Sept. | 1382 | 190 |
| Oct. | 1692 | 227 |
| Nov. | 1322 | 113 |
| Dec. | 1800 | 98 |
| Jan. | 2370 | 130 |
| Feb. | 2140 | 201 |
| March | 1454 | 170 |
| April | 1584 | 100 |
| May | 1500 | 126 |
| June | 1930 | 326 |
| Average | 1717.4 | |

Table 1.9: OSU's achieved schedule overtime.

Table 1.10 states the number of opened rooms each month in the OSU's achieved schedule. Of course, the number of opened rooms does not change from the planned schedule since the staff are not allowed to use other rooms. But, the average occupancy rate dropped on average to 78.8% per month with 6 months below the 80% occupancy rate target specified by the management of the CHU, with only one month having a minimum occupancy rate above 50%.

| Month | #opened rooms | Min Occ. rate | Avg. Occ. rate |
|----------------|---------------|---------------|----------------|
| Sept. | 59 | 29.9% | 79.2% |
| Oct. | 59 | 12.9% | 76% |
| Nov. | 48 | 19% | 79.3% |
| Dec. | 48 | 44% | 82.9% |
| Jan. | 59 | 29.7% | 81.9% |
| Feb. | 52 | 50.8% | 79% |
| March | 59 | 33.1% | 80.6% |
| April | 49 | 23.7% | 81.5% |
| May | 46 | 12.3% | 77.3% |
| June | 60 | 9.9% | 70.7% |
| Average | 53.9 | | 78.8% |

Table 1.10: OSU's achieved schedule opened rooms.

The OR idle time is the time where the room is opened but no surgery is taking place. Normally, the idle time is supposed to be 20 minutes between each two consecutive surgeries, which is the time required by the OSU to clean and prepare the room for the next surgery. We can calculate the total expected idle time as:

$$\text{Total expected idle time} = (\text{Number of surgeries} - \text{Number of opened rooms}) \times 20$$

Table 1.11 compares the ORs expected idle times with the actual attained ones from the OSU's achieved schedules. From this, we can clearly see that there is an increase of around 50

hours on average per month between the expected and attained idle times. In other words, the OSU is wasting approximately 50 hours per month from the ORs time.

| Month | Rooms | Attained | Expected | Increase |
|------------|-------------|-------------|-------------|---------------|
| Sept. | 59 | 6767 | 3220 | 110.15% |
| Oct. | 59 | 7145 | 3140 | 127.54% |
| Nov. | 48 | 4852 | 2840 | 70.84% |
| Dec. | 48 | 4473 | 2860 | 56.39% |
| Jan. | 58 | 5681 | 3280 | 73.2% |
| Feb. | 52 | 6947 | 3160 | 119.84% |
| March | 59 | 6733 | 3360 | 100.38% |
| April | 49 | 4681 | 2860 | 107.12% |
| May | 46 | 5421 | 2540 | 113.42% |
| June | 60 | 7939 | 3360 | 136.28% |
| Avg | 53.9 | 6064 | 3062 | 98.04% |

Table 1.11: OSU’s achieved schedule expected and achieved idle times in minutes.

Finally, table 11 states the number of urgent and priority kits (both cases) that are found in the OSU’s achieved schedule. On average, the OSU had approximately 7.9 urgent kits and 49.4 priority kits per month with a total of 57.3 problem kits (urgent + priority) per month.

| Month | #urgent | #priority |
|----------------|------------|-------------|
| Sept. | 3 | 39 |
| Oct. | 3 | 44 |
| Nov. | 8 | 49 |
| Dec. | 14 | 64 |
| Jan. | 6 | 58 |
| Feb. | 1 | 53 |
| March | 28 | 72 |
| April | 4 | 46 |
| May | 8 | 42 |
| June | 4 | 27 |
| Average | 7.9 | 49.4 |

Table 1.12: OSU’s achieved schedule urgent and priority kits.

1.8.3 Difference between planned and achieved schedules

In the previous sections (1.8.1 and 1.8.2) we showed both the planned schedules and their corresponding achieved ones for the OSU. When comparing each criterion, we can see that there is a big difference happening when applying the real durations of surgeries.

Starting with the violated constraints, we see that the number of late ambulatory surgeries increased from an average of 7.6 to 10.1 surgeries per month (+32.9%), while the average number of not allowed kits dropped from 10.2 to 6.2 kits per month (-39.2%).

Next, the average overtime increased from 804.1 minutes ($\approx 13h$) to 1717.4 minutes ($\approx 28h$) per month (+113.6%). Of course, the number of opened ORs stayed the same, but the average occupancy rate dropped from 80.9% to 78.8% (-2.1%) making it lower than the goal of CHU's management ($\geq 80\%$).

Finally, the average number of urgent kits increased from 6.6 to 7.9 kits per month (+19.7%), while the average number of priority kits decreased from 62.6 to 49.4 kits per month (-21.1%).

The degradation in solution qualities is a result of the big uncertainty in surgeries estimated durations. Since the estimated duration is calculated using the average of the past surgeries from the same type, we will start by showing the number of surgeries for the different types as shown in Figure 1.15. Due to the fact that many surgery types have low number of surgeries (75 types have less than 5 surgeries for each), this make it very difficult to get a good estimation for the duration of the new surgeries from that type.

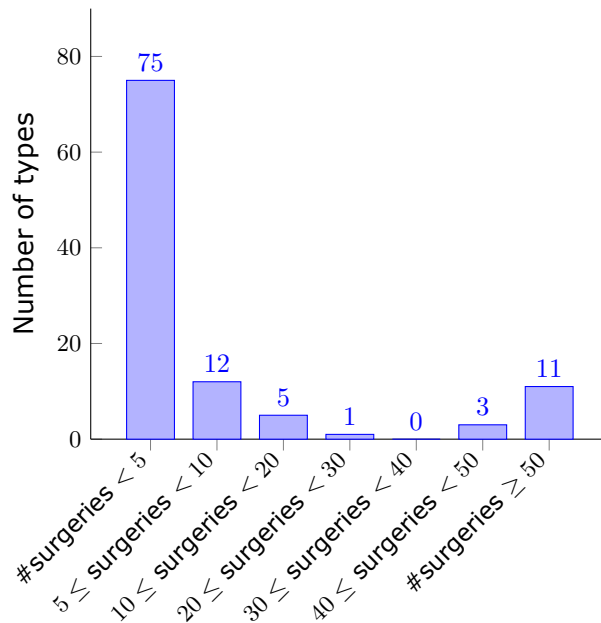


Figure 1.15: Number of surgeries for the different types.

When comparing the difference between the achieved (real) and planned durations for all the surgeries, only 987 surgeries out of the total 2069 surgeries (approximately 47.7%) had a real duration that differed from the planned duration by less than 20 minutes, while 939 surgeries (approximately 45.4%) had a difference between 20 minutes and 1 hour. For the rest of the surgeries, the difference can reach up to 3 hours. These big differences highlight the stochastic nature of the scheduling problem of the CHU and are shown in more details in Figure 1.16.

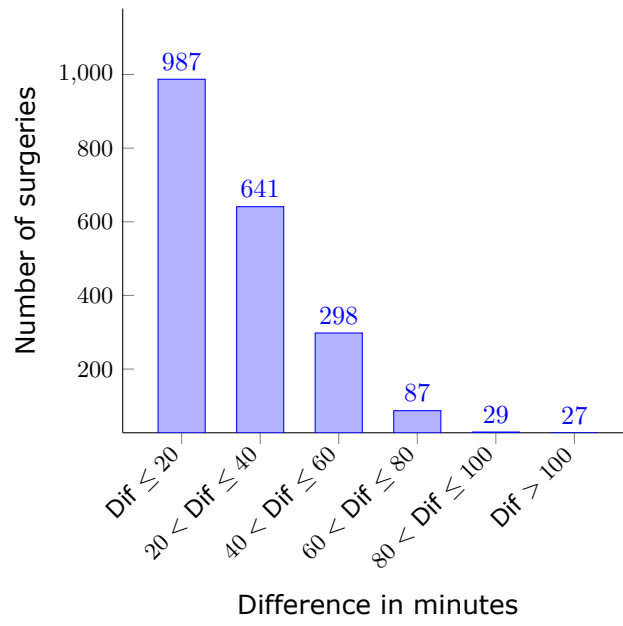


Figure 1.16: Difference between surgeries planned and real durations.

1.9 Conclusion

In this chapter, we explained the surgical case scheduling problem of the CHU in the first part. We started by describing the concerned elements, namely the OSU, the SU and the surgical kits. We then explained the current work process at the CHU, and the problems faced there with this work flow. The main cause for these problems is the lack of global planning, where each surgeon is responsible for scheduling his/her own surgeries. Finally, we described the objectives of the CHU's management. These objectives are mainly to minimize the costs.

In the second part we analysed the data that we received from the OSU. We explored the planned schedules, which use the estimated durations of the surgeries, and the achieved schedules that use the real durations of surgeries and represent what actually happened at the OSU after performing the surgeries. We then showed that there is quite a big degradation in solution qualities when applying surgeries real durations in place of the estimated ones. This degradation is a result of the stochastic nature of surgeries durations, which is showed in details in the last section by comparing surgeries estimated and real durations.

LITERATURE REVIEW

2.1 Introduction

Surgery is one of the most important activities in hospitals. It has been recognized as a main source of income as it generates approximately two thirds of hospital revenues [Jackson, 2002]. In addition, it counts for approximately 40% of hospital costs [Macario et al., 1995] through the use of facilities (operating rooms, intensive care beds, etc.) and the personnel costs (surgeons, nurses, anaesthetists, etc.).

Furthermore, the operating room scheduling and planning problems involve many stakeholders, such as hospital management, surgeons and patients, which makes it very hard to find a solution that satisfies the requirements of all the involved parties and improves the overall performance of the surgical unit. For these reasons, numerous researchers studied the operating room planning and scheduling problems [Cardoen et al., 2010, Guerriero and Guido, 2011, Hulshof et al., 2012, Zhu et al., 2018]. One can note that a different classification method for the literature was used in each of the available literature reviews. In [Cardoen et al., 2010], six categories are presented: patient characteristics, performance measures, decision delineation, research methodology, uncertainty and applicability of research. On the other hand, the authors in [Zhu et al., 2018] organized their literature in six fields: decision level, scheduling strategies, patient characteristics, problem features, mathematical models and solutions and methods. In our work, we will classify the literature based on the three decision level that make up the problem (strategic, tactical and operational levels).

Indeed, the OR planning and scheduling problems can be viewed as being made up of three phases corresponding to three decision levels [Testi et al., 2007]. The first phase (strategic level) corresponds to the problem of determining on a long term horizon (more than a year): the number, the type and the working hours of the available operating rooms (ORs), and of assigning these ORs to surgical specialities [Zhang et al., 2009, Dexter et al., 2002, VanBerkel and Blake, 2007, Dexter et al., 1999]. The second phase (tactical level) corresponds to the problem of creating a timetable, often cyclic and referred to as Master Surgical Schedule (MSS) on a medium term horizon (quarterly/semi-annually). This timetable defines the specific assignment of ORs to surgeons. A new MSS must be generated whenever the total amount of OR time changes or when the make-up of some specialities changes. This can occur not only as a response to long-term changes in the overall OR capacity or fluctuations in staffing, but also in response to seasonal fluctuations in demand [Blake et al., 2002, Blake and Donald, 2002, Beliën and Demeulemeester, 2007, van Oostrum et al., 2008]. Finally, the last phase (operational level) is referred to as the Surgical Case Scheduling (SCS) problem. This phase is generally separated into two sub-problems referred to as "advance scheduling" and "allocation scheduling" [Magerlein and Martin, 1978].

This chapter provides a comprehensive survey of the research done on operating room

planning and scheduling problem. We start by exploring the three decision levels involved in the problem with more attention towards the SCS problem that resides in the operational level. In addition, we explore the different variants of the SCS problem and take into account the dynamic aspect and the uncertainties that arise in the problem. Finally, we present a recap of the discussed literature in the form of a summary table.

2.2 Strategic level

At the strategic level (also referred to as session planning problem), the scheduling problems have a long planning horizon and decisions are based on gathered information and forecasts. However, there is a big ambiguity in the literature regarding whether a problem belongs to the strategic or tactical level [Choi and Wilhelm, 2014, Zhu et al., 2018, Blake and Donald, 2002]. For this, we will consider the classification mentioned in [Choi and Wilhelm, 2014], where at the strategic level, two main problems are identified namely: capacity planning and capacity allocation.

2.2.1 Capacity planning

In this problem, the goal is to determine the quantity of required resources in order to cover the demands in a cost-effective way [Choi and Wilhelm, 2014]. In other words, the goal of this problem is to determine the working hours of the available ORs. Many researchers viewed this problem in a detailed taxonomic classification [Hulshof et al., 2012], where the objective is to provide an accurate OR session capacity (OR working hours) in order to avoid under-utilization and over-utilization of ORs as both are expensive. In [Lehtonen et al., 2013], the authors studied the effect of implementing duration categories on the OR productivity to optimize the time constituting the end of the workday. On the contrary, the authors in [Koppka et al., 2018] studied the usability of having different capacities for the different ORs at a hospital by having a model that tactically distributes the OR time over the ORs. Moreover, the authors in [Ma and Demeulemeester, 2013] proposed a multilevel integrative an Integer Linear Programming (ILP) model to solve the deterministic version of the capacity planning problem. They consider the resource allocation and take into account the variability and its effect on resource utilizations, such as the expected bed occupancy of each ward.

2.2.2 Capacity allocation

The capacity allocation problem consists in assigning surgical specialities to operating-room days. In [Dexter et al., 2003], the authors focus on filling the allocated OR time at different rates to maximize the OR efficiency. They investigate how and when to release allocated OR time appropriately from a surgical speciality when a new surgical case arrives to another surgical speciality that does not have enough time for it. They show that the best approach resulted in only slightly less OR efficiency achieved when the service whose time is released is the service that has the most allocated but unused OR time at the moment the new surgical case is scheduled. More recently, the authors in [Creemers et al., 2012] present a model for assigning time slots to different classes of patients at strategic decision level with the objective of minimizing the total expected weighted waiting time of a patient. They use a service queueing model to obtain the expected waiting time of a patient of a particular class, given a feasible allocation of service time slots. Their solutions provide a valuable tool for OR management to estimate the effect of OR capacity allocation on different patient categories waiting times.

2.3 Tactical level

The problem addressed at the tactical level concerns the development of a timetable called the Master Surgical Schedule (MSS), which specifies the assignment of surgeons (or surgical groups) to ORs time block each day.

Many researchers [[Wachtel and Dexter, 2008](#), [Beliën et al., 2006](#), [Beliën and Demeulemeester, 2007](#)] showed that a better MSS can have a big positive impact on the overall usage of the involved resources (e.g. wards, Post-Anaesthesia Care Unit (PACU) beds, and Intensive Care Unit (ICU)) when these resources are considered at the MSS building stage.

The assignment of surgeons (or surgeon groups) to ORs and consequently the use of an MSS at any hospital are directly affected by the management procedures implemented at the operating theatre. The authors in [[Main, 1995](#)] presented and analysed the following three different management procedures:

- **Open-Scheduling:**

This strategy represents a first come, first served model, where ORs are not assigned to surgeons, but instead are open to whoever calls first. In other words, surgeons can assign surgeries to any available OR at their own convenience. This strategy is a good option to surgeons who can schedule their surgeries far in advance such as plastic surgery and ophthalmology, while it is not favourable to specialities that cannot forecast schedule far in advance, such as general surgery and cardiac surgery.

- **Block-Scheduling:**

In the block-Scheduling strategy, blocks of OR time are assigned to specific surgeons or groups of surgeons each week to schedule their surgical cases. Generally speaking, the surgeon or group owns the block and it cannot be released even when unused, which affects the OR utilization in a negative way. This strategy provides surgeons with a predictable schedule in which they can book time, but reallocating and assigning block time might raise difficult political issues.

- **Modified Block-Scheduling:**

Block scheduling can be modified in two ways to create a more flexible version. In one way, a mix of both Open and Block scheduling procedures are implemented, where some time is blocked and some is left open for any surgeon (or group) to use. In the second, unused OR time blocks are released at a predefined time before surgery (such as 72 hours). This procedure combines both block and open scheduling while taking the advantages of both methods by balancing the needs of specialities that can schedule in advance with the ones that cannot.

The MSS construction problem has been studied extensively in the literature. However, it is not always clear in the literature which procedure was used. For this, we will try to our best to mention the used procedure when possible.

In [[Lehtonen et al., 2013](#)], the authors used a newsvendor model to construct the MSS. They show that OR productivity can be improved markedly by increasing flexibility in the OR team's working hours. Moreover, the authors in [[Penn et al., 2017](#)] used a MIP solver to build their MSS in a reasonable amount of time and showed that using solvers to generate MSS is possible for medium sized hospitals. Furthermore, the authors in [[Blake and Donald,](#)

2002] developed an integer-programming model and a post-solution heuristic based on the characteristics found in the case of the surgical suite at Mt. Sinai Hospital in Toronto, Canada. Their method is successfully implemented in the hospital and the results show remarkable administrative savings and the ability to generate quickly an equitable MSS.

In [Kharraja et al., 2006], the authors presented a decision making tool for OR managers that includes two different MSS implementation approaches that work in both Block-Scheduling and modified Block-Scheduling systems. Their first approach is an ILP model which constructs an MSS for all surgeons regardless of their specialities with the objective of minimizing the gap between the requests of all surgeons and the total supply. On the contrary, the second approach is based on the same ILP model, but provides an MSS for the surgical groups. The objective function of the second approach is to minimize the gap between the total supply and the request of all surgical groups. The second approach provides better results as surgeons have more flexibility inside groups to perform their surgeries. In contrast, the authors in [Kuo et al., 2003] considered a Block-Scheduling procedure and presented a LP based approach with the objective of determining the best mix of surgical OR time allocations so that it either minimizes the hospital total costs or maximizes the professional receipts. Their financial productivity based model was tested on real data obtained from the division of general surgery at Duke University medical centre, and the computational results showed an increase in the financial efficiency but their MSS can only be implemented in an ideal scenario due to the assumptions made. Similar results were presented in [Beliën et al., 2009], where the authors developed a solution approach based on Column Generation (CG) technique which shows that considerable savings to the staffing costs can be obtained by integrating the OR and nurse scheduling process together. Despite the potential of their IP model to determine the human resource management waste sources, the real-life savings are probably much smaller because of the difficulty that accompanies modifying an MSS in a real-life environment.

A number of MIP based and metaheuristic approaches were presented in [Beliën and De-meulemeester, 2007], in which the authors took into account the stochastic numbers of arriving patients and the stochastic length of stay for each operated patient. Their methods considered the OR capacity and surgery demand constraints and the daily expected bed occupancy, the variance on this occupancy, the expected bed shortage and the probability of a shortage on each day performance measures with the objective of levelling the resulting bed occupancy. Both approaches performed well in the experiments, but the metaheuristic method took the edge with better results and faster computation times.

2.4 Operational level

The last hierarchical phase is mainly concerned with generating a schedule of elective surgeries and is called "surgical case scheduling".

As explained in [Magerlein and Martin, 1978], the surgical case scheduling problem is generally separated into two sub-problems referred to as "advanced scheduling" and "allocation scheduling". The Advanced Scheduling Problem (AdvSP) assigns patients to ORs and days, while the Allocation Scheduling Problem (AllocSP) provides the appropriate sequence for the assigned patients each day at each OR. In the literature, some researchers focused on one sub-problem while others included both sub-problems in their research.

In addition, such operational real-world optimisation problems can be classified into 4

categories according to the quality of the data (potential uncertainty in the available data) and their evolution (possible changes during the execution of a plan):

- **Deterministic static problems:** all inputs are known in advance and do not change during the execution of a plan. This is the case in SCS problems when the list and durations of the surgeries are known in advance, and they do not change during the realization of the planning.
- **Deterministic dynamic problems:** part or all the data are revealed during the process, and the information dynamically revealed is known with certainty. For example in the case of the SCS problem, new patients that require urgent surgeries arrive during the realization of the planning, but the durations of the surgeries are known for certain at the patient arrival.
- **Non-deterministic static problems:** data do not change during the realization of the process, but are partially known as random variables, or range of values. For example in the case of the SCS problem, the list of patients is known in advance, but the durations of the surgeries are only known as a range estimate of their real duration.
- **Non-deterministic dynamic problems:** part or all the inputs are revealed during the execution of a plan, but exploitable knowledge is available before-hand. For example, the case of the SCS problem which includes both elective (known in advance) and non-elective (unexpected) surgeries, and a range estimate of the durations of the surgeries is in this category.

The remainder of this section is divided in three parts: the first one explores the advanced scheduling problem, the second one explores the allocation scheduling problem, and the last one focuses on the work that considers both advanced and allocation scheduling simultaneously. In addition, each part will be further divided following the 4 categories mentioned before.

2.4.1 Advanced scheduling problem

Advanced scheduling is the process of assigning a surgery date and an OR to a patient. This problem determines the date only and does not deal with the order of the surgeries in each day nor the starting time of each surgery.

Deterministic static AdvSP

Several research studies address the AdvSP with the deterministic static settings, where no stochastic aspects are considered and the full list of considered patients is known in advance (offline). For instance, in [Fei et al., 2008] the authors solved the assignment problem in a Block-Scheduling strategy with a decomposition-based branch-and-price algorithm that minimizes the total overtime and ORs underutilization costs. They assumed that human and instruments resources are available whenever needed and tested their approach on randomly generated instances and their results showed that their method worked well for the instances. Later on, the same authors in [Fei et al., 2009] studied the same problem but in an Open-Scheduling strategy in multiple multi-functional ORs within one week. They developed a binary

IP model that considers the ORs opening hours, surgeries deadline and surgeon availability to minimize the total overtime and ORs underutilization costs. They reformulate their model due to the high computational time as a set-partitioning problem and then they solved it using a CG procedure. Their results showed that if the involved resources are well organized, an efficient surgeries assignment can be obtained.

In [Ozkarahan, 2000], the authors present a goal programming model with the objective of minimizing ORs overtime and idle times while maximising the satisfaction of the surgeons, staff and patients. Their approach involved sorting the requests for a particular day on the basis of block restrictions, room utilization, surgeon preferences and intensive care capabilities. Their experiments show that their method managed to give better OR utilization while satisfying all the requirements.

Another approach was presented in [Molina and Framinan, 2009], where the authors considered three different policies to allocate patients to ORs:

1. P-S-OR: This is the normal approach that guaranties the continuity of care, where patients are first assigned to surgeons, and then the set "patient-surgeon" is allocated to an OR.
2. P-OR-S: Patients are first assigned to an OR, and then surgeons are allocated to the set "patient-OR". This policy is more flexible than the previous one since patients do not depend on the availability of surgeons, but does not take into account the patient preferences regarding the operating surgeon.
3. Hybrid: Patients who are previously assigned to a surgeon (P-S-OR) are allocated to a suitable OR, whereas other patients are assigned based on the (P-OR-S) policy, where a "knapsack surgeon" is created at each OR shift and these patients are assigned to this surgeon. The surgeon who is not assigned to this shift will perform the surgeries in the (P-OR-S) set.

They test the three policies in a Block-Scheduling system with the objective of maximizing the quality of service represented by the total weight of the performed surgeries, which depends on the type of surgery (normal or microsurgery), while considering surgery deadline constraints. Their experiments show that the (P-OR-S) yields the best results as it is more flexible, while the (P-S-OR) had the worst improvement. In all three policies, their results were better than the ones reported in the real data they used for the tests.

Deterministic dynamic AdvSP

By extension, some researchers propose to address the deterministic version of the problem with dynamic setting. For example in [Velásquez et al., 2008], the authors proposed a heuristic approach based on a bin packing problem which takes into account the limited resource availability with the objective of minimising a penalty sum of not satisfying the time window of a patient and using a resource that uses additional capacity. They tested their method on instances from a German hospital and the results showed high quality solutions. Moreover in [Ogulata and Erol, 2003], the authors proposed a mathematical model and a solution approach that formulates the weekly AdvSP as a multi-objective binary IP model. Their objectives were to minimize the patients waiting time, maximize the OR capacity utilization and balance the distribution of surgeries among surgical groups. Their proposed solution approach works by dividing the scheduling problem into three logical steps:

1. Patients admission planning: a subset of patients is selected from the initial list based on their priorities and arrival date.
2. Patients assignment: the patients of the selected subset are assigned to surgical groups.
3. Day and OR assignment: a date and an OR are assigned to each surgery of each surgical group independently from the other groups.

In [[Dios et al., 2015](#)], the authors present a Decision Support System (DSS) that embeds a number of optimization procedures (exact and approximate) to solve the dynamic AdvSP. Their system is capable of producing a medium-term plan (up to six months) by selecting assigned patients from a waiting list. The objective of their DSS was to maximise an indicator of the quality of service combining patient's medical priority and the need to fulfil regional imposed standards. These standards state that for each illness type, the time between the diagnosis and the surgery should not exceed a predefined maximum number of days called 'clinical guarantee'.

Moreover, the authors in [[Testi et al., 2007](#)] addressed the three decision levels of the operating theatre management problem. In the operational level, they considered the dynamic AdvSP and proposed a simulation method that uses a priority rank to schedule surgeries in order to minimise the number of cancelled patients and the total overtime. The authors tested several dispatching rules and found that using the shortest processing time rule yields the best results.

Finally, in [[Luo et al., 2016](#)], the authors developed a Mixed Integer Linear Programming (MILP) and integrated it into a rolling horizon approach to solve the scheduling problem on a day-to-day basis. In their work, surgeries exact durations and maximum waiting times are known in advance and they didn't include any resource constraints. The objective considered is to minimise the idle times at the ORs in order to balance the occupancy rates among the ORs. They tested their approach on generated data and found that using a rolling horizon method allows a more flexible planning of the pool of surgeries and minimises the total idle time in comparison to using a non-rolling horizon method.

Non-deterministic static AdvSP

Some researchers addressed the non-deterministic static variant of the AdvSP. In [[Hans et al., 2008](#)], the authors proposed constructive and local search heuristics for the assignment problem with the objective of freeing operating rooms and capacity while taking into account the uncertainty that may occur (e.g. patient cancellation). Moreover, the authors in [[Rachuba and Werners, 2017](#)] developed a fuzzy multi-objective model while simultaneously reserving time windows dedicated for randomly arriving emergencies. The considered objectives are to minimise the waiting time of patients (patients perspective), minimise the total overtime (staff perspective), and maximise the total number of operated patients (management perspective). In their tests, they showed that only small changes in the absolute levels of the stakeholder's goals were necessary to find an acceptable solution.

Another approach was presented in [[Denton et al., 2010](#)], where the authors proposed two models with the objective of minimizing the total costs represented by the sum of fixed OR opening costs and the variable cost of overtime. Their first model was a two-stage stochastic linear program with binary decisions in the first stage and simple recourse in the second stage.

The second model was a robust formulation counterpart. They then compared the results of the two models and showed that the robust method performs way faster than the stochastic model and it has the benefit of limiting the worst-case outcome of the resource problem.

The authors in [Hans et al., 2008] proposed constructive and local search heuristics to maximize the capacity utilization by minimizing the total planned slack and to minimize the risk of overtime, and thus cancelled patients. Whereas the authors in [Addis et al., 2014] proposed a robust formulation based on the formulation presented in [Bertsimas and Sim, 2004]. They considered the objective of minimising the costs associated with patients tardiness and waiting time while allowing overtime when needed. The authors used generated data to test two versions for both the deterministic and robust MILPs, with one that allows overtime and another that does not. The results show that allowing overtime reduces the overall penalty for both models, and the robust model outperforms the deterministic by reducing the numbers of cancelled patients and total overtime.

Non-deterministic dynamic AdvSP

Despite the huge real applications for the robust dynamic version of the problem, very few researches have been done on this particular problem. In [Herring and Herrmann, 2012] the authors presented a stochastic dynamic model for the single-day single-OR version of the problem. In addition, they proposed several threshold-based heuristics to solve the scheduling problem. Their threshold identifies how much open time to preserve for future primary cases. Another example in [Addis et al., 2016], where the authors developed two ILP to solve both the deterministic and robust problems and integrated these models in a rolling horizon method. In their work, each surgery has maximum waiting time (due date) and estimated duration and the objective is to minimize the total waiting time and tardiness of patients. The rescheduling happens because of the semi-elective surgeries and cancelled surgeries (no time to perform).

In [Min and Yih, 2010a], the authors presented a stochastic dynamic programming model to address the scheduling problem of patients with different priorities. In their work, a decision is made at the beginning of each period to determine the number of patients to be scheduled based on the trade-offs between the cost for overtime work and the cost for surgery postponement. The authors proposed a stochastic dynamic programming model to solve the problem. Their experiments show that the consideration of patient priority results in significant differences in surgery schedules compared to the schedule that ignores the patient priority, where the decision regarding the level of detail in classifying patient priority depends on the trade-offs between computation time and solution quality.

2.4.2 Allocation scheduling problem

Given the advanced schedule, the second sub-problem then determines the precise sequence of surgical procedures and the allocation of resources for each OR time block in order to implement it as efficiently as possible. Similar to the AdvSP, many researchers tackled the AllocSP with various performance criteria. No dynamic approaches to this problem can be found in the literature due to the nature of the allocation problem, where the goal is to provide the sequence for patients that are already assigned to an OR at a given date.

Deterministic static AllocSP

The first contribution concerning this problem was presented by [[Sier et al., 1997](#)]. The authors proposed a discrete-time MIP nonlinear model that assigns patients to OR timeslots. They also developed a simulated annealing approach to obtain near-optimal solutions and defined a weighted penalty function that takes into account the equipment usage conflicts, OR usage collision and patients age.

Several mixed integer linear programming solution approaches were developed in [[Cardoen et al., 2009a](#)], which are either exact or heuristic, and consider multiple objectives including minimising the starting time for children and priority patients and minimising the stay in recovery after the closure of the day-care centre. Furthermore in [[Cardoen et al., 2009b](#)], the same authors used a branch-and-price methodology with the objective of minimising the peak use of recovery beds, the occurrence of recovery overtime and the violation of various patient and surgeon preferences.

A two-stage no-wait flow shop scheduling problem variant was studied by [[Hsu et al., 2003](#)] to tackle the allocation problem in an ambulatory surgical centre, while considering the PACU capacity. The two-stage were represented by OR with independent and non-identical surgeons for the first stage, and PACU for the second stage. They developed a tabu based heuristic to solve the two sub-problems iteratively: the objective of the first one is to determine the minimum nurses number required in a single PACU so that the completion time of the last patient is less than a pre-determined threshold; in the second sub-problem, the objective is to minimize the makespan given a fixed number of nurses. They tested their method on real data from an ambulatory surgical centre in a university hospital and showed that it finds near optimal solutions.

The impact of sequencing rules on PACU and ORs is tested in [[Marcon and Dexter, 2006](#)]. The authors applied seven sequencing rules (some are borrowed from the classical scheduling theory) to each surgeon's list of cases independently. The behaviour of the different rules has been tested using discrete event simulation, while considering the PACU completion time, delays in PACU admission, PACU staffing and ORs over-utilization performance measures. The results showed that the Longest Cases First (LCF) rule performs the worst, generates more over-utilized ORs, requires more PACU nurses each working day and generates more days with at least one PACU admission delay.

In addition to the PACU resource, the authors in [[Latorre-Núñez et al., 2016](#)] considered the surgery required resources. They formulated an integer linear programming model for the sequencing problem and developed a genetic algorithm based metaheuristic and a constructive heuristic to solve the problem. One interesting point was investigated in [[Augusto et al., 2010](#)], where patients are allowed to recover in the operating room if no recovery room was available at the end of his/her surgery. The authors modelled the problem as a 4-stage hybrid flowshop problem with blocking constraints and solved it using Lagrangian relaxation method. They tested their approach against not allowing the recovery in OR on randomly generated instances.

Non-deterministic static AllocSP

Very few researches were done on the robust version of the allocation problem. In [[Denton et al., 2007](#)], the authors proposed a stochastic optimization model and three practical heuristics for computing OR schedules. They focused on the effects of sequencing surgeries and

scheduling start times in a single day single OR environment. The tests were made on real instances from a local hospital and the results show that it is possible to generate substantial reductions in total surgeon and OR team waiting, OR idling, and overtime costs.

2.4.3 Advanced and allocation scheduling problem

The decomposition of the SCS problem into two sub-problems (AdvSP and AllocSP) is motivated by the need to reduce the problem complexity. The AdvSP, which is generally treated as a generalized bin-packing problem with unequal bins, has a strong NP-hard complexity as shown in [Hans et al., 2008]. In addition, the allocation step is argued in [Cardoen et al., 2009b] to be also an NP-hard problem.

Such decomposition leads to significant disadvantages, as pointed out by [Cardoen et al., 2009a]: the quality of the achieved surgeries sequence in the allocation scheduling step depends heavily on the quality of the achieved surgeries assignment that was generated in the advance scheduling step. The reason for this is that the sequencing is bound by the previous assignment decisions, which were made without considering the objectives of the sequencing step. This claim is also supported by the work of [Jebali et al., 2006], where the authors show that allowing some degree of modification to the previous assignment of surgeries leads to better results in the sequencing step, but this comes at the price of higher computational times.

To avoid such disadvantages, many researchers proposed approaches that solve the advance and allocation scheduling sub-problems at the same time in a single procedure. We will refer to this problem as the Advanced and Allocation Scheduling Problem (AASP).

Deterministic static AASP

Some research studies address the two sub-problems with the deterministic and static settings. For example, in [Roland et al., 2006] the authors proposed a mathematical model based on the literature classical resource constraint project scheduling model that aims at minimizing the overtime and operating costs of ORs. Their model takes into account the renewable resources, such as surgeons, nurses and anaesthetists, and the non-renewable resources, such as sterile materials and pharmaceuticals. They developed a genetic-algorithm approach to solve the problem and tested it on real instances from a Belgian hospital. More recently, the same authors proposed a modified version of their previous model in [Roland et al., 2010] that focuses on human resources availability by allowing renewable resources (nurses and anaesthetists) to leave the surgery before its end based on a consumption rate. In their work, they Similarly to their previous approach, they developed a genetic-algorithm approach to solve the problem and tested it on real instances from a Belgian hospital.

The Integrated Physician (surgeon) and Surgery Scheduling Problem (IPSSP) was presented in [Van Huele and Vanhoucke, 2014], where surgeries can only be performed in a predefined set of ORs by a pre-specified number of surgeons (based on their skills). In their work, the authors developed a MILP formulation for the problem based on the most frequently observed objective (overtime) and restrictions of the surgery scheduling and the physician rostering problem in the literature (recovery beds and staff availability). They analysed the schedules by relaxing both surgeon and surgery constraints and then measured the effects of

integrating the surgeon preferences on the surgery schedule. Their experimental results allowed to make a ranking system for the surgery and surgeon constraints, where the constraints at the bottom of the ranking list can be satisfied more easily without disrupting the surgery schedule.

One of the very few papers that included medical devices sterilisation aspect in the problem is [Beroule et al., 2016], where the authors propose several metaheuristics (genetic algorithm, particle swarm optimisation, and tabu search) to solve an advanced and allocation scheduling problem including the medical device sterilisation step with the objective of minimising the number of medical devices required at the same time to respect a schedule.

Deterministic dynamic AASP

More studies can be found concerning the deterministic dynamic version of the advanced and allocation problem. Some researchers tend to solve the problem in two steps, where a step corresponds to each sub-problem (AdvSP and AllocSP). For example, the authors in [Fei et al., 2006] considered a Block-Scheduling strategy and formulated a binary programming model for the weekly AdvSP and solved it using Column-Generation-Based Heuristic (CGBH) that assigns surgeries to pre-determined time blocks with the objective of minimizing the overtime costs. The daily AllocSP was formulated as a flow shop scheduling problem that aims to minimize the cost of ORs and recovery rooms and solved using a hybrid genetic algorithm. They tested their approach on randomly generated instances and obtained good quality solutions. Later on, the same authors formulated a multi-objective mathematical model in [Fei et al., 2010] that aimed at minimising both overtime cost in the operating theatre and unexpected idle time between surgical cases, while maximising the ORs utilization to solve the problem in the same Block-Scheduling setting. The weekly AdvSP is formulated as a set-partitioning IP model that is solved by a CGBH, using a tabu search procedure. Then the daily AllocSP, which takes into account the recovery beds' availability, is solved as a two-stage hybrid flow-shop problem by a hybrid genetic algorithm (similar to their previous approach in [Fei et al., 2006]). The experiments were conducted on real data from a Belgian university hospital and the results show that the obtained surgery schedules using the proposed method have less idle time between surgical cases, much higher utilisation of operating rooms and produce less overtime.

Moreover, the authors in [Jebali et al., 2006] also solved the problem in two steps. The first step consists of assigning surgical operations to operating rooms while minimizing the overtime, under-time and patient waiting times. The second step is formulated as a two stage hybrid flow shop and consists in sequencing the assigned operations while considering various resource-related constraints (e.g. recovery beds) in order to improve the overall ORs usage. Two strategies for the sequencing step are presented, where in the former, surgeries assignment to ORs obtained from the first step is not reconsidered, whereas in the latter, the process is less constrained by allowing to redefine the surgeries assignment to ORs. Their experiments were performed on a set of randomly generated instances and show good performance of the operations sequencing without reconsidering the assignment problem as well as the accuracy of the assignment step in terms of patients selection while minimizing total cost.

Other researchers used a one step approach to solve both sub-problems in a single unified procedure. A local search method based on iterated local search and variable neighbourhood descent was presented in [Riise and Burke, 2011] to solve the problem. The algorithm uses

simple Relocate and Two-Exchange neighbourhoods. The authors also analyse the search space associated with these move operators for three typical fitness surfaces that represent different compromises between surgeon overtime, patient waiting time, and waiting time for children in the morning of the surgery day. The method showed good results when tested on real instances from a Norwegian hospital. In [Guinet and Chaabane, 2003], the authors proposed a model that divides each OR day into 8 periods. The authors considered the staff and equipment availability, with the objective of minimising the total overtime and patients' waiting times, and they solved the problem using a primal-dual heuristic. The experiments were conducted on generated data and showed good results.

Another method was presented in [Marques et al., 2012]. The authors allocate along a weekly time horizon elective surgical cases to a specific time period in a day at a certain OR with the objective of maximizing the utilization of surgical suits. The authors developed a MILP for the problem with an improvement heuristic that improves the non-optimal solutions. The experiments were done on instances from a local hospital in Lisbon and the results show that the addition of an improvement heuristic improved almost all non-optimal solutions.

Moreover, the authors in [Dexter et al., 2000] proposed a method to schedule surgeries to ORs while minimizing the staffing costs and taking into account the preferences of surgeons and patients regarding the date and time of the case. They suggested the use of an "overflow block time", which refers to a time block reserved for surgeons who can not schedule all their surgeries in their regular reserved shifts. The authors tested their approach using computer simulation and the results showed that the use of the overflow time block provided surgeons and patients more flexibility, while increasing OR staffing costs slightly over the minimum achieved using regular shifts strategy. Furthermore, the authors of [Everett, 2002] proposed a simulation model to schedule patients from a waiting list to operating rooms. Their goal was to provide a tool that will respect the budget limits at the hospital described as the opening hours and the number of recovery beds while also minimizing the waiting time of patients from the waiting list based on the urgency of their surgery. Similarly, the authors in [Niu et al., 2011] proposed a simulation model to analyse the performance of operating rooms by scheduling surgeries to ORs and detecting possible bottlenecks from the number of available resources.

Finally, the authors in [Vali Siar et al., 2017] developed a MILP to solve the scheduling (both AdvSP and AllocSP) and rescheduling problem. In their work, surgeries are scheduled one day in advance. They assumed that surgeries exact durations are know in advance and they considered the recovery beds resources. Their objective was to minimize the tardiness, idle time, and overtime. In addition, they assumed that surgeries are assigned to a specialty instead of a single surgeon, thus any surgeon from that specialty can perform the surgeries assigned to his/her specialty.

Non-deterministic static AASP

Concerning the robust static version of the problem, the authors in [Lamiri et al., 2008b] proposed a stochastic mathematical programming model and a CG approach to solve the elective surgery scheduling problem where emergency patients can arrive randomly. The objective of the method is to minimise patient-related costs and the expected operating rooms' utilization costs. Their approaches reported solutions within 2% of the optimum in short computation times for their instances.

Furthermore, the authors in [Min and Yih, 2010b] considered the problem of scheduling

a set of elective surgeries to ORs with downstream capacity constraints with stochastic surgery durations and stochastic length of stay at the ICU unit. The authors proposed a stochastic optimisation model and solved the problem using a sample average approximation method while minimizing the total of patient related costs represented by their length of stay at the ICU and overtime costs. The tests were conducted on slightly modified real data and the results show promising values in terms of overtime and patients costs. By contrast, the authors in [Neyshabouri and Berg, 2017] proposed a two stage robust optimization model to address the uncertainty in surgeries durations and length of stay in the intensive care unit after the surgery with the objective of minimising patients assigning costs (patient priority and waiting time), the total overtime, and the total cost of lack of ICU capacity. They propose an adapted column-and-constraint generation method to obtain exact solutions for the proposed formulations. The tests were conducted over generated data and show potential for the model to manage multi-stage care operations.

In [Wang et al., 2014], the authors proposed a stochastic model with the goal of minimizing the total costs of opening ORs and overtime. They developed a CGBH to solve the integer programming problem. They tested their approach on randomly generated instances and the results show that the CGBH method was able to obtain results within a 5% gap of the lower bound obtained by the linear program for large scale problems that can not be solved using the solver.

Non-deterministic dynamic AASP

Finally, some researchers studied the non-deterministic dynamic variant of the problem. In [Persson and Persson, 2009], the authors proposed an approach using simulation including optimization to schedule patients to ORs from a waiting list while considering the uncertainty in surgeries durations. Their objective is to minimize the patient related costs for delayed and cancelled (outsourced) patients and the hospital related costs expressed as extra beds and staff overtime. Experiments were carried over instances from a local hospital in Sweden and the results shows that it is possible to reduce the waiting time for patients by around 16 weeks. Whereas in [Saadouli et al., 2015], the authors considered the problem of scheduling surgeries to ORs from a waiting list while taking into account the recovery phase for each patient on one of the limited recovery beds and the stochastic aspect of surgeries durations. They propose an approach in which a slack is given to each surgery in the waiting list and then a knapsack model chooses the surgeries for a given day from the list. Finally, the assignment of these surgeries to the ORs is done using a mixed integer programming model with the objective of minimizing the makespan and patients' waiting times.

Different stakeholders perspectives were taken into account in [Marques and Captivo, 2017]. The authors proposed three mixed integer linear programming models. The first one focuses on the management needs as the objective is to maximise the use of the OR time, while the second model focuses on surgeons perspective with a morning shifts objective of minimising the waiting time for surgeries and an evening shifts objective of maximising the use of available resources and number of scheduled surgeries. The last model provides a middle ground with the administrative objective being used for the morning shifts and the surgeons objective for the evening shifts. In addition, the authors compare the results of the deterministic versions with a robust one that they developed based on the objective functions presented in each version of the models.

In [Bowers and Mould, 2004], the authors developed a simulation approach to maximize the utilization of the ORs and minimize the overtime. In addition, they proposed a model with a number of approximations and showed that the model offers competitive results to their simulation approach. In addition, the authors in [Lamiri et al., 2008a] proposed a stochastic mathematical programming model to solve the surgical case scheduling problem combining the elective surgeries that can be planned in advance and the emergency surgeries that arrive randomly and have to be performed at the day of arrival. Their objective is to minimize the sum of elective surgeries related costs and operating rooms overtime costs. They also proposed a Monte Carlo optimization method that combines the Monte Carlo simulation and Mixed Integer Programming and showed that this method is proved to converge to a real optimum as the computation budget increases.

2.5 Synthesis and analysis

This section provides a summary of the discussed literature on the surgical case scheduling problem. Tables 2.1, 2.2, and 2.3 classify each reference according to:

- The sub-problem it solves: whether advanced scheduling, allocation scheduling or both.
- The setting of the solved problem: Deterministic Static (DS), Deterministic Dynamic (DD), Non-deterministic Static (NS), or Non-deterministic Dynamic (ND).
- The considered resources constraints:
 1. Recovery beds
 2. Pre-operative beds
 3. ICU
 4. Equipment (e.g. medical instruments)
 5. Staff
 6. Instruments sterilisation
 7. Others
- The considered objective function:
 1. Makespan
 2. Overtime
 3. OR utilization
 4. Patient waiting time
 5. Penalty for constraint violation
 6. Freeing ORs
 7. Distribute load on surgeons
 8. Staff satisfaction
 9. Patients satisfaction
 10. Management satisfaction

11. Quality of service
 12. Peak use of recovery beds
 13. Total completion time
 14. Objectives based on medical instruments sterilisation
 15. Other
- The used solution method.

Table 2.1: Classification of the AdvSP bibliographical references

| Reference | Problem setting | | | | Resources | Objective | Solution method |
|------------------------------|-----------------|----|----|----|-----------|-----------|---|
| | DS | DD | NS | ND | | | |
| [Fei et al., 2008] | ✓ | | | | - | 2, 3 | branch-and-price |
| [Fei et al., 2009] | ✓ | | | | - | 2, 3 | CGBH |
| [Hans et al., 2008] | | | ✓ | | - | 2, 3, 6 | constructive heuristics with local search |
| [Velásquez et al., 2008] | | ✓ | | | 4, 5 | 5 | heuristic based on bin packing |
| [Ogulata and Erol, 2003] | | ✓ | | | - | 3, 4, 7 | multi-objective binary IP model |
| [Ozkarahan, 2000] | ✓ | | | | 3, 5 | 2, 3, 8 | goal programming |
| [Dios et al., 2015] | | ✓ | | | - | 8, 9 | MILP based framework |
| [Roshanaei et al., 2017] | | ✓ | | | - | 4, 6, 9 | logic-based benders' decomposition |
| [Molina and Framinan, 2009] | ✓ | | | | - | 11 | MILP |
| [Rachuba and Werners, 2017] | | | ✓ | | - | 2, 4, 10 | fuzzy optimization |
| [Herring and Herrmann, 2012] | | | | ✓ | - | 3, 9 | threshold based heuristics |
| [Denton et al., 2010] | | | ✓ | | - | 2, 6 | stochastic and robust MILPs |
| [Hans et al., 2008] | | | ✓ | | - | 2, 3, 6 | constructive and local search heuristics |
| [Addis et al., 2014] | | | ✓ | | - | 2, 4, 5 | ILP |
| [Addis et al., 2016] | | | | ✓ | - | 4, 5 | rolling horizon with ILP |
| [Luo et al., 2016] | | ✓ | | | - | 3 | rolling horizon with MILP |
| [Testi et al., 2007] | | ✓ | | | - | 2, 15 | simulation with dispatching rules |
| [Min and Yih, 2010a] | | | | ✓ | - | 2, 4 | stochastic dynamic programming |
| [Guido and Conforti, 2017] | | ✓ | | | - | 2, 3, 15 | hybrid genetic algorithm |

Table 2.2: Classification of the AllocSP bibliographical references

| Reference | Problem setting | | | | Resources | Objective | Solution method |
|------------------------------|-----------------|----|----|----|-----------|-------------|--|
| | DS | DD | NS | ND | | | |
| [Sier et al., 1997] | ✓ | | | | - | 5 | simulated Annealing |
| [Cardoen et al., 2009a] | ✓ | | | | 1 | 5, 9, 12 | MILP based approaches |
| [Cardoen et al., 2009b] | ✓ | | | | 1, 6 | 2, 5, 9, 12 | branch-and-price |
| [Hsu et al., 2003] | ✓ | | | | 1 | 1 | tabu search based heuristic |
| [Marcon and Dexter, 2006] | ✓ | | | | 1 | 3, 5, 12 | constructive heuristics |
| [Latorre-Núñez et al., 2016] | ✓ | | | | 1, 4, 7 | 1, 13, 15 | MILP |
| [Denton et al., 2007] | | | ✓ | | - | 2, 3, 8 | stochastic heuristic |
| [Augusto et al., 2010] | ✓ | | | | 1, 7 | 1 | lagrangian relaxation method for a 4-stage hybrid flowshop with blocking constraints |

Table 2.3: Classification of the AASP bibliographical references

| Reference | Problem setting | | | | Resources | Objective | Solution method |
|-----------------------------|-----------------|----|----|----|-----------|-----------|---------------------------------|
| | DS | DD | NS | ND | | | |
| [Roland et al., 2006] | ✓ | | | | 4 | 2, 6 | genetic algorithm |
| [Roland et al., 2010] | ✓ | | | | 4, 5 | 2, 6 | genetic algorithm |
| [Riise and Burke, 2011] | | ✓ | | | - | 2, 9, 15 | local search |
| [Guinet and Chaabane, 2003] | | ✓ | | | 4, 5 | 2, 9 | primal-dual heuristic |
| [Marques et al., 2012] | | ✓ | | | 1 | 2, 3, 15 | MILP with improvement heuristic |

Table 2.3: Classification of the AASP bibliographical references

| Reference | Problem setting | | | | Resources | Objective | Solution method |
|---------------------------------|-----------------|----|----|----|-----------|-----------|--|
| | DS | DD | NS | ND | | | |
| [Fei et al., 2006] | | ✓ | | | 1 | 2, 3 | CG for the advanced scheduling and hybrid genetic algorithm for the allocation part |
| [Fei et al., 2010] | | ✓ | | | 1 | 2, 3, 15 | tabu search heuristic for the advanced scheduling and hybrid genetic algorithm for the allocation part |
| [Van Huele and Vanhoucke, 2014] | ✓ | | | | 1, 5 | 2 | MILP |
| [Jebali et al., 2006] | | ✓ | | | 1 | 2, 3, 4 | MILP |
| [Min and Yih, 2010b] | | | ✓ | | 3 | 2, 15 | sample average approximation |
| [Lamiri et al., 2008b] | | | ✓ | | - | 2, 3, 15 | stochastic LP and CG |
| [Persson and Persson, 2009] | | | | ✓ | 1 | 2, 15 | simulation-based optimization |
| [Saadouli et al., 2015] | | | | ✓ | 1 | 1, 4 | knapsack model with dynamic programming and bi-objective MILP |
| [Neyshabouri and Berg, 2017] | | | ✓ | | 3 | 2, 4, 15 | adapted column-and-constraint generation |
| [Marques and Captivo, 2017] | | | | ✓ | - | 3, 4, 15 | MILP |
| [Bowers and Mould, 2004] | | | | ✓ | - | 9 | monte carlo simulation |
| [Dexter et al., 2000] | | ✓ | | | - | 9, 10 | simulation method |
| [Everett, 2002] | | ✓ | | | 1 | 4 | simulation method |
| [Lamiri et al., 2008a] | | | | ✓ | - | 2, 4, 15 | monte carlo simulation and MIP |
| [Wang et al., 2014] | | | ✓ | | - | 2, 6 | CGBH |
| [Vali Siar et al., 2017] | | ✓ | | | 1, 2 | 2, 6 | MILP |
| [Beroule et al., 2016] | ✓ | | | | 6 | 15 | genetic algorithm, particle swarm optimisation, and tabu search |

From this summary of the studied literature, it is clear that the SCS problem is a very varied problem. This variance is mainly related to the differences in hospitals work-flows and stakeholders preferences. In addition, the used methods in the literature to solve the problem are vary varied as well.

Starting with the AdvSP (Table 2.1), we can see that the majority of the researchers opted not to consider any resources. This is due to the fact that resources are usually considered in the allocation scheduling step where the start and end time of each surgery will be known, and thus resources violations can be measured. Moreover, we can distinguish the most considered objectives to be the overtime, OR utilization, and patients' waiting time.

Moving to the AllocSP (Table 2.2), we can note the absence of any dynamic setting considerations. This is due to the fact that at this step, the set of considered surgeries and their assignment to ORs is already known. Moreover, most researchers focus in this problem on the recovery beds constraint (e.g. PACU and ICU). This also is reflected on the considered objectives as most researches consider the minimisation of a penalty for constraint violations in addition to the makespan objective.

Finally, we can note that the majority of the research done on the SCS problem considers the AASP (Table 2.3). Moreover, the most considered constraints are the recovery beds and the most considered objectives are the overtime, the ORs utilization, and freeing ORs. Despite the huge impact for the surgery department over the whole hospital, most researchers studied did not consider the non-deterministic version of the AASP. This is not true in most cases as most hospitals operate in dynamic on-line environments where patients keep arriving to the hospital for surgery and some inevitable real-time events may cause a change in the schedule. These events are usually caused by the change in the estimated duration for the surgery due to some complications. A good view for the gap between scheduling theory and practice is shown in [Cowling and Johansson, 2002] where they show also that scheduling models are not making use of real-time information.

In addition, one major component of the SCS problem that lacked behind in terms of researchers attention is the medical instruments sterilising process. This claim can be further supported in Tables 2.1, 2.2, and 2.3, where the only researches that considered the sterilizing step are [Beroule et al., 2016, Cardoen et al., 2009b]. As we explained before, medical instruments are an important resource that is available in limited quantities, and any delays in providing the required instruments at the time of the surgery can cause delays and even surgery cancellation.

The work reported in this thesis concerns the AASP of the CHU, where the consideration of the medical instruments sterilisation is a big priority. As we can note from this summary, there are to the best of our knowledge only two articles that consider resource 6 (Instruments sterilisation), with only one of these two articles that treats the AASP problem. Thus, the importance of this work is justified from the fact that none of the reported articles in the literature treats the AASP while considering all proposed objectives by the CHU.

2.6 Conclusion

In this chapter, we reviewed the literature on the operating rooms planning and scheduling problems. In the review, we visited the 3 decision levels that make the problem. We then focused on the operational level as it is where our problem resides. Next, we explored the two sub-problems that combine the SCS problem (AdvSP and AllocSP) and explained that decomposing the problem and solving each sub-problem individually negatively affects the quality of obtained solutions. For this, we explored the literature for the approaches that solve both sub-problems in a single step (AASP). Finally, we presented a summary and synthesis for the presented literature, where we showed that there are no existing researches in the literature that cover the needs of the CHU, while considering the medical instruments sterilisation step.

In the rest of this thesis, we will solve the SCS problem of the CHU by simultaneously solving the Advanced and Allocation Scheduling Problem (AASP). We will consider both the medical instruments resource and the sterilising step, while taking into account the dynamic and non-deterministic aspects of the problem. For the sake of clarity, we classify the 4 contributions of this thesis based on the evolution of available data over time (static or dynamic) and the quality of the data (deterministic or non-deterministic) in Table 2.4.

| Chapter | Problem setting | | | | Resources | Objective |
|-----------|-----------------|----|----|----|-----------|--------------|
| | DS | DD | NS | ND | | |
| Chapter 3 | ✓ | | | | 4, 6 | 2, 6, 14 |
| Chapter 4 | | ✓ | | | 4, 6 | 2, 4, 14, 11 |
| Chapter 6 | | | ✓ | | 4, 6 | 2, 6, 14 |
| Chapter 7 | | | | ✓ | 4, 6 | 2, 4, 14, 11 |

Table 2.4: Thesis chapters position

PART II

The Deterministic Surgical case scheduling problem

DETERMINISTIC STATIC SURGICAL CASE SCHEDULING PROBLEM

3.1 Introduction

In this chapter, we will tackle the deterministic and static (DSSCS) problem of the CHU in which the surgeries and their durations are known in advance. The goal is to schedule a set of surgeries on a given time horizon while minimizing several objectives. According to the CHU, the first objective is to minimize the total overtime of the staff members of the OSU. The second objective consists in minimizing the number of used operating rooms, and finally the third objective is to keep the number of urgent and priority kits as low as possible. The following section presents two approaches to solve the problem.

First, we start by defining formally the problem. Then, we prove that the problem is NP-hard, and we propose a mathematical formulation and a constructive heuristic method to solve it. Finally, we show the numerical experiments and compare our planned schedules with the one of the OSU.

3.2 Formal problem presentation

The problem consists in scheduling O surgical operations using K types of kits and performed by S surgeons in R operating rooms on an horizon of T days.

In this problem, each surgeon s must perform a set λ_s of surgeries and is assigned to work on specific days in specific rooms (with parameter $\delta_{srt} = 1$ if surgeon s can use room r on day t , and 0 otherwise). For each surgery i we know its duration p_i in minutes, its type (we set parameter $a_i = 1$ if surgery i is ambulatory, and 0 otherwise), and its required quantity q_{ik} of each type of kits k . And for each kit type k , we know the number Q_k of available kits for this type.

Starting from the different kits reusing conditions listed in Chapter 1, we can observe that some of the pick-up hours, although they allow to distribute the load on the SU, do not have any impact on surgeries schedule. For example, all kits collected at 11:30, 13:00 and 14:30 will be reusable from the same hour the next day. Thus, we can consider only the pick-up shuttle of 14:30. We can also ignore the pick-up shuttle of 17:30 and 18:30 as the kits collected at these hours will be reusable at the same hour as those collected at 7:00 the next day. Furthermore, since used kits must be pre-disinfected for 30 minutes at the OSU after a surgery, a kit can be collected by a shuttle if the surgery that used it ends at least 30 minutes before the shuttle hour. This leads us to consider the pick-up "preparation limits" at 14:00 and 15:30 instead of the actual pick-up times of 14:30 and 16:00.

Similarly, for the delivery shuttles, the one of 17:30 can be ignored since no surgery is supposed to start after 17:30. We can also assume that the first delivery shuttle arrives at 8:15 (room opening) instead of 7:00.

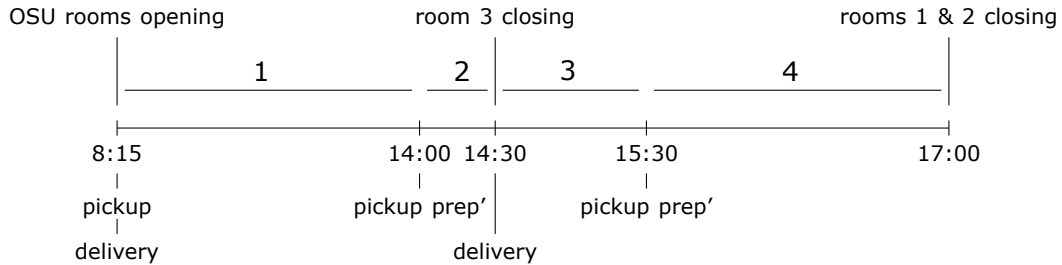


Figure 3.1: The four periods of an operating room.

From these key hours, we propose to divide the working day into $J = 4$ periods, as shown in Figure 3.1. In terms of periods, the different kits reusing conditions listed in Section 1.6 become:

1. A kit used for a surgery that ends in period 1 of day t is collected by the shuttle of 14:30 and:
 - It cannot be used again in period 2, 3 or 4 of day t (thus a kit cannot be used more than once in any given day).
 - It can be used again for another surgery that starts in period 1 or 2 of day $t + 1$. In that case, the kit will be considered as a **priority kit** in the OSU (case 1).
 - If it is not treated as a priority, it can be used again for another surgery that starts in period 3 or 4 of day $t + 1$.

2. A kit used for a surgery S_1 that ends in period 2 or 3 of day t is collected by the shuttle of 16:00 and:
 - It cannot be used again in period 4 of day t .
 - It can be used again for another surgery S_2 which starts in period 1 or 2 of day $t + 1$. In that case, the kit will be considered as an **urgent kit** in the OSU.
 - If it is not treated urgently, it can be used again for another surgery S_2 which starts in period 3 or 4 of day $t + 1$.
3. A kit used for a surgery S_1 that ends in period 4 of day t is collected by the shuttle of 8:15 on day $t + 1$ and:
 - It cannot be used again in period 1 and 2 of day $t + 1$.
 - It can be used again for another surgery S_2 which starts in period 3 or 4 of day $t + 1$. In that case, the kit will be considered as a **priority kit** in the OSU (case 2).
 - If it is not treated as a priority, it can be used again for another surgery S_2 which starts in any period of day $t + 2$.

This interaction between period of surgeries and kit reuse can be summarized as shown in table 3.1.

| End of kit utilization on day t | Next possible reuse on day $t + 1$ | | | |
|-----------------------------------|------------------------------------|-----------------------|-----------------------|-----------------------|
| | Period 1 | Period 2 | Period 3 | Period 4 |
| Period 1 | Priority kit (case 1) | Priority kit (case 1) | Normal | Normal |
| Period 2 or 3 | Urgent kit | Urgent kit | Normal | Normal |
| Period 4 | - | - | Priority kit (case 2) | Priority kit (case 2) |

Table 3.1: Reusing conditions for the kits.

For each room r and each day t , the length of period j in minutes is denoted by d_{rt}^j (0 if the room is closed that day and cannot be used). For added convenience, the parameter $d_{rt}^{bf} = \sum_{j=b}^{j=f} d_{rt}^j$ will specify the total duration from period b to period f of day t for room r . Finally, priority kits and urgent kits are penalized through respective cp and cu unit costs.

From the problem description, we have the following constraints:

- All surgical operations must be scheduled.
- Each surgeon can only perform surgeries in a room if he is allowed to use the room on that day.
- Each day, a surgeon can use at most one room.
- Each day, the overtime in each room cannot exceed ε_{max} .
- Ambulatory surgeries must finish before time limit A_{max} .

- Quantities of resources (kits) must be respected, taking into account the reusing conditions given in Table 3.1.

The objective is to schedule all the surgical operations in order to minimize the total overtime, the total number of rooms opened, and the total penalties of emergencies and priorities in the SU.

We will solve the problem in 3 steps:

1. In the first step, we seek to minimize the overtime (in minutes) in order to be able to schedule all the surgeries on the planning horizon.
2. Then, we schedule all surgeries while minimizing the total number of rooms opened, allowing the minimum overtime found in step 1.
3. Finally, we schedule all surgeries while minimizing the total penalties of urgent and priority kits at the SU without exceeding the number of opened rooms found in step 2 nor the total overtime from step 1.

3.3 Proof of complexity

Proposition 1:

SCS is NP-hard

Proof : We will show that a simplified version of SCS is an NP-hard problem starting from the 3-PARTITION problem. More precisely, we will consider the decision version of the two problems.

SCS simplified version : We will consider the problem SCS' ; a particular case of SCS with only 1 OR, 1 surgeon, a horizon of length m , 1 period per day, n number of surgeries to schedule, and no kits.

SCS' associated decision problem, $DSCS'$: Given an instance of SCS' , is there a valid schedule for the instance with no overtime?

3-PARTITION decision problem : Let us consider an integer B , a set $A = \{a_1, a_2, \dots, a_n\}$ of $n = 3m$ integers such that the following relations hold:

$$\sum_{i=1}^n a_i = mB$$

$$\frac{B}{4} < a_i < \frac{B}{2}, \quad \forall i \in \{1, \dots, n\}$$

Can A be partitioned in m subsets A_1, A_2, \dots, A_m such that the sum of the numbers in each subset is equal to B ?

1. **It is clear that $SCS_d \in NP$:** A proposition of a YES solution is polynomially verifiable (given a schedule, it is easy to check the amount of overtime).

2. Polynomial construction of an instance of $DSCS'$ starting from a 3-PARTITION instance .

Our reduction will assign an instance I' of the SCS to any instance I of 3-PARTITION as follows:

- We have only one room ($R = 1$) used by one surgeon ($S = 1$) available for $T = m$ days.
- Each day has only one period ($J = 1$) and the legal duration of a day is set to B (we have $d_{1t}^1 = B$ for all days t).
- There are no kits ($K = 0$).
- We are given m operations to be scheduled over the planning horizon with $p_i = a_i$ for each operation i and there are no required kits.

It is clear that this construction is polynomial.

3. There is a solution for I' with no required overtime if and only if there exists a 3-PARTITION of A .

If we consider a 3-partition A_1, A_2, \dots, A_m of A such that the sum of the numbers in each subset is equal to B , we can clearly plan to schedule each operation i related to given $a_i \in A_k$ to the corresponding day k . Since the duration of each day is set to B , then we clearly have no overtime over the whole planning horizon.

4. There exists a 3-PARTITION of A if and only if there is a solution for I' with no required overtime

Let us assume that we are given a solution I' with no overtime. We can partition A into m subsets, assigning a_i into subset A_k if operation i is scheduled on day k . Since there is no overtime, the sum of the numbers in each subset must be lower or equal to B and we have a 3-PARTITION of A .

5. Conclusion : we have a reduction from 3-PARTITION to SCS' . Thus, the particular case SCS' of SCS is NP-Hard, and hence, SCS is NP-Hard too.

3.4 A MILP formulation

The parameters, variables and constraints of the MILP we propose are as follows.

Parameters:

T total number of days in the horizon

J number of periods in each day

K total number of kit types

O total number of operations

R total number of operating rooms

| | |
|---------------------|--|
| S | total number of surgeons |
| p_i | duration of operation i |
| λ_s | set of operations for surgeon s |
| q_{ik} | required quantity of kits of type k for operation i |
| Q_k | total quantity of kits of type k owned by the block |
| cu | urgent kit penalty |
| cp | priority kit penalty |
| d_{rt}^j | duration of period j of day t for room r (0 if the room is closed that day) |
| d_{rt}^{bf} | duration from period b to period f ($\geq b$) of day t for room r |
| δ_{srt} | binary parameter equal to 1 if surgeon s can use room r on day t , 0 otherwise |
| a_i | binary parameter equal to 1 if operation i is ambulatory, 0 otherwise |
| A_{max} | latest time for ambulatory operations to be performed at |
| ε_{max} | maximum allowed overtime for any room on any given day |
| u_γ | binary parameter equal to 1 if $\gamma = J$ and 0 otherwise |

We introduce the following decision variables:

| | |
|--------------------|---|
| w_{itr} | binary variable equal to 1 if operation i is scheduled at day t in room r |
| x_{itr}^{bf} | binary variable equal to 1 if surgery i begins at period b and finishes at f , on day t , in room r |
| ε_{tr} | integer variable representing the total overtime in room r at day t |
| L_{tr} | binary variable equal to 1 if room r is used at day t |
| U_{srt} | binary variable equal to 1 if surgeon s is using room r at day t |
| E_{tk} | integer variable representing the total urgent kits of type k at day t |
| Y_{tk}^1 | integer variable representing the total priority kits (case 1) of type k at day t |
| Y_{tk}^2 | integer variable representing the total priority kits (case 2) of type k at day t |

The constraints are the following:

Constraints (3.1) ensure that all surgeries are scheduled and constraints (3.2) state that each surgery must start and finish the day when it is scheduled.

$$\sum_{t=1}^T \sum_{r=1}^R w_{itr} = 1, \quad \forall i \in \{1, \dots, O\} \quad (3.1)$$

$$\sum_{b=1}^J \sum_{f=b}^J x_{itr}^{bf} = w_{itr}, \quad \forall i \in \{1, \dots, O\}, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\} \quad (3.2)$$

Constraints (3.3) state that each day in each room no more than one surgery can be scheduled over the time γ which separates two periods (surgeries with $b \leq \gamma$ and $f > \gamma$). In other words, they prevent the case where more than one surgery start in a period and extend to other periods.

$$\sum_{i=1}^O \sum_{b=1}^{\gamma} \sum_{f=\gamma+1}^J x_{itr}^{bf} \leq 1, \quad \forall \gamma \in \{1, \dots, J-1\}, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\} \quad (3.3)$$

Constraints (3.4) control the workload of surgeries for each interval of the day and each room. The first term of constraints (3.4) calculates the total duration of surgeries that start and finish between β and γ . The second term computes the sum, over all surgeries starting before β and finishing after γ , of the part of the surgeries that overlap periods β to γ . The sum of the two terms must be less than or equal to the total duration of periods from β to γ if the room is open (plus overtime ε_{tr} if γ is the last period).

$$\sum_{i=1}^O \sum_{b=\beta}^{\gamma} \sum_{f=b}^{\gamma} p_i \cdot x_{itr}^{bf} + \sum_{i=1}^O \sum_{b=1}^{\beta-1} \sum_{f=\gamma+1}^J d_{rt}^{\beta\gamma} \cdot x_{itr}^{bf} \leq d_{rt}^{\beta\gamma} \cdot L_{tr} + u_{\gamma} \cdot \varepsilon_{tr}, \quad (3.4)$$

$$\forall \beta \in \{1, \dots, J\}, \forall \gamma \in \{\beta, \dots, J\}, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\}$$

Constraints (3.5) state that a surgeon is using a room if there is at least one operation scheduled for him in that room on that day, while constraints (3.6) state that a room is considered used on a day if there is at least one operation scheduled in the room for that day.

$$w_{itr} \leq U_{srt}, \quad \forall s \in \{1, \dots, S\}, \forall i \in \lambda_s, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\} \quad (3.5)$$

$$w_{itr} \leq L_{tr}, \quad \forall i \in \{1, \dots, O\}, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\} \quad (3.6)$$

Constraints (3.7) indicate that a surgeon is only allowed to use rooms that he is assigned to, while constraints (3.8) ensure that each surgeon does not use more than one room each day.

$$U_{srt} \leq \delta_{srt}, \quad \forall s \in \{1, \dots, S\}, \forall r \in \{1, \dots, R\}, \forall t \in \{1, \dots, T\} \quad (3.7)$$

$$\sum_{r=1}^R U_{srt} \leq 1, \quad \forall s \in \{1, \dots, S\}, \forall t \in \{1, \dots, T\} \quad (3.8)$$

Constraints (3.9)-(3.10) ensure that there is enough time to sterilize each kit between two utilizations at the OSU. Constraints (3.9) state that the number of used kits per day must respect the available quantity of these kits. Constraints (3.10) ensure that the used kits at last period of day t are not used at the first and second periods of the next day $t+1$.

$$\sum_{i=1}^O \sum_{r=1}^R q_{ik} \cdot w_{itr} \leq Q_k, \quad \forall t \in \{1, \dots, T\}, \forall k \in \{1, \dots, K\} \quad (3.9)$$

$$\sum_{i=1}^O \sum_{r=1}^R q_{ik} \cdot \left(\sum_{b=1}^J x_{itr}^{bJ} + \sum_{b=1}^2 \sum_{f=b}^J x_{i(t+1)r}^{bf} \right) \leq Q_k, \quad (3.10)$$

$$\forall t \in \{1, \dots, T-1\}, \forall k \in \{1, \dots, K\}$$

The expression of urgent kits is stated in constraints (3.11) and the expressions of priority kits (case 1 and 2) are stated in constraints (3.12)-(3.13).

$$\sum_{i=1}^O \sum_{r=1}^R q_{ik} \cdot \left(\sum_{f=2}^J \sum_{b=1}^f x_{itr}^{bf} + \sum_{b=1}^2 \sum_{f=b}^J x_{i(t+1)r}^{bf} \right) - Q_k \leq E_{tk}, \quad (3.11)$$

$$\forall t \in \{1, \dots, T\}, \forall k \in \{1, \dots, K\}$$

$$\sum_{i=1}^O \sum_{r=1}^R q_{ik} \cdot \left(\sum_{f=1}^J \sum_{b=1}^f x_{itr}^{bf} + \sum_{b=1}^2 \sum_{f=b}^J x_{i(t+1)r}^{bf} \right) - Q_k - E_{tk} \leq Y_{tk}^1, \quad (3.12)$$

$$\forall t \in \{1, \dots, T\}, \forall k \in \{1, \dots, K\}$$

$$\sum_{i=1}^O \sum_{r=1}^R q_{ik} \cdot \left(\sum_{f=2}^J \sum_{b=1}^f x_{itr}^{bf} + \sum_{b=1}^J \sum_{f=b}^J x_{i(t+1)r}^{bf} \right) - Q_k - E_{tk} \leq Y_{tk}^2, \quad (3.13)$$

$$\forall t \in \{1, \dots, T\}, \forall k \in \{1, \dots, K\}$$

Next, constraints (3.14) ensure that all ambulatory surgeries finish before A_{max} . The first part of the left term of these constraints calculates the total duration of all surgeries that are assigned to finish in the first and second periods and the second part adds the total duration of only the ambulatory surgeries that finish in the third period. This sum must be less than or equal to A_{max} . This is due to the fact that the third period ends after A_{max} and thus we consider that ambulatory surgeries are scheduled before the others in that period.

$$\sum_{i=1}^O \sum_{b=\beta}^2 \sum_{f=b}^2 p_i \cdot x_{itr}^{bf} + \sum_{i=1}^O \sum_{b=\beta}^3 a_i \cdot p_i \cdot x_{itr}^{b3} \leq A_{max} - \sum_{j=1}^{\beta-1} d_{rt}^j, \quad (3.14)$$

$$\forall \beta \in \{1, \dots, 3\}, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\}$$

Constraints (3.15) states that no ambulatory surgeries finish at the fourth period.

$$\sum_{b=1}^J a_i \cdot x_{itr}^{bJ} = 0, \quad \forall i \in \{1, \dots, O\}, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\} \quad (3.15)$$

Constraints (3.16) impose that the overtime at each room at any day does not exceed ε_{max} .

$$\varepsilon_{tr} \leq \varepsilon_{max}, \quad \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\} \quad (3.16)$$

Finally, constraints (3.17) to (3.24) define the range of the variables.

$$w_{itr} \in \{0, 1\}, \quad \forall i \in \{1, \dots, O\}, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\} \quad (3.17)$$

$$x_{itr}^{bf} \in \{0, 1\}, \quad (3.18)$$

$$\forall i \in \{1, \dots, O\}, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\}, \forall b \in \{1, \dots, J\}, \forall f \in \{b, \dots, J\}$$

$$\varepsilon_{tr} \geq 0, \quad \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\} \quad (3.19)$$

$$L_{tr} \in \{0, 1\}, \quad \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\} \quad (3.20)$$

$$U_{srt} \in \{0, 1\}, \quad \forall s \in \{1, \dots, S\}, \forall r \in \{1, \dots, R\}, \forall t \in \{1, \dots, T\} \quad (3.21)$$

$$E_{tk} \geq 0, \quad \forall t \in \{1, \dots, T\}, \forall k \in \{1, \dots, K\} \quad (3.22)$$

$$Y_{tk}^1 \geq 0, \quad \forall t \in \{1, \dots, T\}, \forall k \in \{1, \dots, K\} \quad (3.23)$$

$$Y_{tk}^2 \geq 0, \quad \forall t \in \{1, \dots, T\}, \forall k \in \{1, \dots, K\} \quad (3.24)$$

The multiple objective functions (3.25) are taken into account by using a lexicographic method.

First, we minimize f_1 the total overtime that is required to schedule all surgeries. Then we add the resulting value of $(\sum_{t=1}^T \sum_{r=1}^R \varepsilon_{tr})$ as an upper bound in our model and we minimize f_2 the total number of opened rooms without exceeding the overtime found for f_1 . Finally, we set the resulting $(\sum_{t=1}^T \sum_{r=1}^R L_{tr})$ as an upper bound in our model and minimize f_3 the total penalty cost of urgent and priority kits without exceeding the total overtime found for objective f_1 and the total number of opened rooms found for objective f_2 .

$$\text{Minimize } Lex \left(\begin{array}{l} f_1 : \sum_{t=1}^T \sum_{r=1}^R \varepsilon_{tr}; \\ f_2 : \sum_{t=1}^T \sum_{r=1}^R L_{tr}; \\ f_3 : \sum_{t=1}^T \sum_{k=1}^K [cu \cdot E_{tk} + cp \cdot (Y_{tk}^1 + Y_{tk}^2)] \end{array} \right) \quad (3.25)$$

3.5 Heuristic approach

Due to the high complexity of the problem, we propose a heuristic approach based on the one of [Kirca and Kökten, 1994] to solve it. This method uses an iterative surgeon-by-surgeon strategy for generating solutions to the problem.

This approach generates solutions to the Deterministic Static SCS problem iteratively. In each iteration, a surgeon s is selected from the list of unscheduled surgeons ϑ . Then, a schedule for λ_s (the list of surgeries of surgeon s) is generated by solving the MILP from the previous section considering only the surgeries of surgeon s , while taking into account all the fixed surgeries from previous iterations. Next, the surgeries in λ_s are fixed and added to the list of fixed surgeries ζ . An outline for the heuristic is shown in Algorithm 1.

Algorithm 1 Constructive heuristic method

```
1:  $\vartheta = \{1, \dots, S\}$  ▷ define unscheduled surgeons list
2:  $\zeta \leftarrow \phi$  ▷ set of fixed surgeries
3: while  $\vartheta \neq \phi$  do
4:   Determine candidate surgeon  $s$  such that  $s \in \vartheta$ 
5:   Schedule surgeries of  $\lambda_s$  using MILP from (3.4) while fixing all surgeries in  $\zeta$ 
▷ schedule surgeries of current surgeon  $s$ 
6:    $\zeta = \zeta + \lambda_s$  ▷ Add current surgeon surgeries to list of fixed surgeries
7:    $\vartheta = \vartheta - s$  ▷ Remove current surgeon from list of unscheduled surgeons
8: end while
```

We tried the following dispatching rules for surgeons selecting step:

1. Highest Load Surgeons First (HLSF).
2. Lowest Load Surgeons First (LLSF).

After several tests, we found that the surgeons dispatching rule has big influence on the quality of solution, due to the limited resources nature of the problem expressed by the number of available kits and ORs time. Despite that the HLSF rule should theoretically work best since surgeons with the most load affect the objectives values the most, this did not hold true as it leads to infeasibility in most instances. This is due to the fact that surgeons with the least load have 1-2 shifts that are shared with other surgeons with more load, and these shifts were filled by these surgeons with more load in previous iterations. On the other hand, using the LLSF rule yields feasible solutions for all instances. For this, we will use the LLSF rule going onward.

3.6 Experimental results

3.6.1 Test Data

In order to test our MILP, we used the 10 test instances described before in Table 1.3.

The values of the parameters are as follows:

- Latest time A_{max} for ambulatory surgeries : 3 p.m.
- Maximum overtime ε_{max} allowed each day : 3 hours
- Penalty cost c_u of urgent kits : 5
- Penalty cost c_p of priority kits : 1

Note that the urgent and priority penalties have been evaluated by the CHU according to the disturbance and stress perceived at the SU.

In addition, Table 3.2 shows the duration of each period in the 3 operating rooms for day t when the rooms are open (0 when a room is closed).

The original data received from the CHU included the planned schedule (with surgeries planned durations, i.e. the durations estimated by the surgeons at the consultation date). As explained before in Section 1.8, some surgeries from other blocks were performed at the OSU. We considered these surgeries by decreasing the duration of the corresponding periods at the surgeries dates by the duration of each of these surgeries.

Our testing environment is:

| d_{rt}^j | Rooms 1 & 2 | Room 3 |
|------------|-------------|--------|
| d_{rt}^1 | 5h45 | 5h45 |
| d_{rt}^2 | 0h30 | 0h30 |
| d_{rt}^3 | 1h00 | 0h00 |
| d_{rt}^4 | 2h30 | 0h00 |

Table 3.2: Periods duration for the ORs.

- Intel® Core™ i3-2120 @ 3.30 GHz
- 8 GB of RAM
- IBM® ILOG® CPLEX® 12.5

3.6.2 Comparison of planned schedules

In order to test the performance of our model, we ran the experiments on the instances described in Table 1.3. We used the planned surgery durations to compare our planned schedules with the planned schedules of the OSU described before in Section 1.8.1. The time limits for these experiments were set to 1 hour per objective (3 hours per instance).

| Month | #Late ambs. | #Not allowed kits |
|----------------|-------------|-------------------|
| Sept. | 11 | 5 |
| Oct. | 9 | 7 |
| Nov. | 5 | 10 |
| Dec. | 9 | 18 |
| Jan. | 12 | 12 |
| Feb. | 7 | 5 |
| March | 4 | 25 |
| April | 5 | 9 |
| May | 8 | 7 |
| June | 6 | 4 |
| Average | 7.6 | 10.2 |

Table 3.3: Violated constraints in the OSU’s planned schedule.

We start by comparing the violated constraints between the two planned schedules. Table 3.3 represents the violated constraints of the OSU, where the ‘#late ambs’ column represents the number of ambulatory surgeries that end after 3 p.m. and the ‘#Not allowed kits’ column represents the total number of kits that do not respect the reuse conditions defined in Table 3.1. These constraints were violated regularly in the original data with an average of 7.6 late ambulatories and 10.2 violated kits per month, which have a negative impact on the performance of the CHU units and the SU in general, in addition to the delays in surgery times while waiting for the violated kits. In both our solutions, no such constraints are violated as they are considered as hard constraints in the mathematical model.

Following the order of the multiple objectives, the tables below compare the three objective functions between the planned schedules acquired using our MILP and heuristic method, and that of the OSU. In these tables, the (*) symbol represents an optimal solution found using the MILP.

Table 3.4 compares the total overtime between the three schedules. In average, the overtime in the schedules obtained by our MILP is around 215.5 minutes (3 hours 35 minutes) per month, while it is 372 minutes (6 hours and 12 minutes) per month using our heuristic method, and 804.1 minutes (13 hours 24 minutes) per month in the original data. The average difference between the MILP and the data from the OSU is approximately 9 hours 48 minutes per month, representing a decrease of approximately 73.2%, while the heuristic method managed a decrease of approximately 53.7% (savings of 7 hours and 12 minutes per month). In addition, the 'max' column shows the maximum overtime in minutes found in a day during the considered month. Our MILP provided lower values than the solution of the OSU in 6 instances, and the heuristic in 5 instances.

The 'LB' column shows the lower bound found by the MILP. These bounds represent the inevitable overtime caused by the fact that sometimes, the total durations of the surgeries assigned to some surgeons exceed the total duration of their shifts. We see that 6 solutions are optimal in our schedules, and the others are very close to the lower bound (only few more minutes). Finally, the 'CPU' column represents the execution time for our MILP. The solver in the MILP approach reached the time limit of 1 hour in 4 instances. In all cases, the big improvements in the MILP solutions happen in the first 20 minutes and very small improvements are found in the remaining time, while the average execution time in the heuristic method is approximately 45 seconds per instance.

| Instance | OSU planned schedule | | MILP planned schedule | | | | Heuristic planned schedule | | |
|----------------|----------------------|-----|-----------------------|----------------|-----|---------|----------------------------|-----|---------|
| | overtime (min) | max | LB | overtime (min) | max | CPU (s) | overtime (min) | max | CPU (s) |
| 1 | 590 | 131 | 0 | 0* | 0 | 16 | 230 | 59 | 11 |
| 2 | 714 | 176 | 368 | 384 | 90 | 3600 | 518 | 108 | 5 |
| 3 | 502 | 81 | 89 | 89* | 23 | 1089 | 124 | 47 | 21 |
| 4 | 656 | 77 | 314 | 318 | 94 | 3600 | 432 | 117 | 174 |
| 5 | 1159 | 120 | 312 | 312* | 123 | 58 | 605 | 113 | 6 |
| 6 | 1278 | 183 | 347 | 350 | 84 | 3600 | 459 | 122 | 87 |
| 7 | 641 | 109 | 225 | 232 | 130 | 3600 | 431 | 179 | 122 |
| 8 | 561 | 65 | 334 | 334* | 106 | 1619 | 455 | 154 | 4 |
| 9 | 870 | 105 | 136 | 136* | 37 | 1031 | 333 | 148 | 6 |
| 10 | 1070 | 266 | 0 | 0* | 0 | 781 | 133 | 77 | 9 |
| Average | 804.1 | | | 215.5 | | | 372 | | |

Table 3.4: DSSCS planned overtime comparison (Objective 1).

Next, table 3.5 shows the number of ORs opened with the MILP, the heuristic method, and in the original data, and also compares the room occupancy rates in the three schedules. In the MILP schedule, we were able to reduce the number of rooms opened in 7 instances out of 10 compared with the schedule of the OSU, while in the heuristic schedule, we managed to reduce the number of rooms in 6 instances out of the 10. We were able to close approximately 1.9 ORs on average each month using the MILP approach and 1.1 ORs using the heuristic

method, where these freed rooms can be used to perform more surgeries. The 'LB' column shows the lower bound found by the solver concerning the rooms opened. The 'CPU' column represents the execution time for our MILP and heuristic approach concerning only the rooms opened objective. In the 7 instances where the time limit was reached in our MILP approach, the solver found the best solution in the first 20 minutes and was unable to find a better solution nor prove the optimality of the best found solution. We tried to increase this time limit to 12 hours without any improvements. Consequently, setting the time limit for this objective to 30 minutes appears to be a good option. On the other hand, the average execution time in the heuristic method is approximately 38 seconds per instance.

When comparing the occupancy rates, we observe approximately 84.4% on average in our MILP solutions, 83.6% in the heuristic solutions, and 80.9% for the OSU. In addition, the 'Min Occ. rate' column represents the minimum occupancy rate found in each schedule. In our MILP schedule, 6 instances had a minimum occupancy rate greater than 50%, compared with 2 instances with similar minimum occupancy rate using the heuristic method, and only 1 instance that had a minimum occupancy rate greater than 50% for the OSU with 4 instances below 20%. These values show that our MILP solutions have better room load distribution than those of the heuristic method and OSU.

| Instance | OSU planned schedule | | | MILP planned schedule | | | | | Heuristic planned schedule | | | |
|----------------|----------------------|---------------|----------------|-----------------------|----------------|---------|---------------|----------------|----------------------------|---------|---------------|----------------|
| | # opened rooms | Min Occ. rate | Avg. Occ. rate | LB | # opened rooms | CPU (s) | Min Occ. rate | Avg. Occ. rate | # opened rooms | CPU (s) | Min Occ. rate | Avg. Occ. rate |
| 1 | 59 | 27.6% | 80.2% | 55.1 | 58 | 3600 | 48.6% | 81.7% | 58 | 9 | 24.3% | 82.2% |
| 2 | 59 | 13.8% | 77.5% | 52.3 | 55 | 3600 | 51.6% | 82.4% | 57 | 7 | 36.9% | 81.7% |
| 3 | 48 | 18.3% | 80.7% | 44.6 | 46 | 3600 | 55.2% | 85.4% | 47 | 15 | 47.5% | 85.6% |
| 4 | 48 | 42.4% | 86.5% | 46.8 | 48 | 3600 | 40.7% | 87.4% | 48 | 114 | 40.6% | 87.1% |
| 5 | 59 | 28.8% | 85.1% | 55.2 | 58 | 3600 | 59.8% | 83.9% | 59 | 8 | 35.8% | 82.3% |
| 6 | 52 | 51.2% | 83.3% | 52 | 52* | 23 | 51.6% | 83.2% | 52 | 105 | 36.1% | 83.8% |
| 7 | 59 | 34.3% | 81.5% | 55.3 | 59 | 3600 | 47.9% | 84.1% | 59 | 98 | 37.8% | 84.3% |
| 8 | 49 | 26.1% | 82.9% | 48 | 48* | 13 | 63.7% | 85.5% | 48 | 7 | 60.2% | 85.0% |
| 9 | 46 | 11.6% | 79.4% | 43 | 43* | 14 | 58.2% | 86.0% | 44 | 4 | 62.9% | 84.2% |
| 10 | 60 | 9.5% | 72.3% | 50.1 | 53 | 3600 | 49.3% | 84.5% | 56 | 8 | 25.9% | 80.3% |
| Average | 53.9 | 80.9% | | | 52 | | 84.4% | | 52.8 | | 83.6% | |

Table 3.5: DSSCS planned opened rooms comparison (Objective 2).

Finally, Table 3.6 shows the total number of urgent and priority kits obtained. In the original data, the average number of urgent kits is 6.6 per month and the average number of priority kits is 62.6 with the average number of problem kits (urgent + priority) approximately 69.2 per month. The MILP was able to eliminate all the urgent kits and provide an average of 3.7 priority kits per month, with a decrease of 94.65% from the OSU's solution in the total number of problem kits. On the other hand, the heuristic managed an average number of problem kits (urgent + priority) of approximately 9.1 kits per month, consisting of an average of 0.6 urgent kits and 8.5 priority kits per month with a decrease of 86.8% from the OSU's solution. The 'CPU' column represents the execution times for the MILP and heuristic method concerning only the urgent and priority kits objective. In the 3 instances where the MILP found an optimal solution, the average execution time is 953 seconds. Following the same behaviour as in the previous objectives, the model finds the large improvements in the solutions in the first 20 minutes and very small improvements during the remaining 40 minutes, while the

average execution time in the heuristic method is 70 seconds per instance.

| Instance | CHU planned schedule | | MILP planned schedule | | | Heuristic planned schedule | | |
|----------------|----------------------|--------------|-----------------------|--------------|---------|----------------------------|--------------|---------|
| | # urgent | # priorities | # urgent | # priorities | CPU (s) | # urgent | # priorities | CPU (s) |
| 1 | 2 | 51 | 0* | 0* | 1199 | 0 | 6 | 21 |
| 2 | 2 | 56 | 0 | 2 | 3600 | 1 | 10 | 12 |
| 3 | 8 | 68 | 0 | 7 | 3600 | 1 | 9 | 27 |
| 4 | 13 | 82 | 0 | 5 | 3600 | 0 | 15 | 336 |
| 5 | 6 | 68 | 0 | 9 | 3600 | 0 | 13 | 8 |
| 6 | 2 | 59 | 0 | 11 | 3600 | 0 | 17 | 134 |
| 7 | 28 | 98 | 0* | 0* | 1165 | 1 | 4 | 108 |
| 8 | 0 | 56 | 0* | 0* | 497 | 0 | 3 | 9 |
| 9 | 2 | 53 | 0 | 1 | 3600 | 2 | 5 | 11 |
| 10 | 3 | 35 | 0 | 2 | 3600 | 1 | 3 | 25 |
| Average | 6.6 | 62.6 | 0 | 3.7 | | 0.6 | 8.5 | |

Table 3.6: DSSCS planned urgent and priority kits comparison (Objective 3).

Although the execution times are much longer, the MILP provides results that are better in every aspect than the heuristic. In addition, it shows that the actual schedules currently implemented at the OSU can be improved significantly.

3.7 Conclusion

In this chapter, we tackled the deterministic static surgical case scheduling problem of the CHU. We started by formulating the problem and proving that this problem is NP-hard. Next, we proposed a mixed integer linear programming formulation which is solved in a lexicographic fashion and a constructive heuristic method to solve the problem.

First, we solved the problem with both our proposed MILP and heuristic method in order to create the planned schedules using surgeries estimated durations. Both our results significantly improve those of the CHU in terms of overtime and urgent kits at the SU. Moreover, the MILP provides better results than the heuristic, but needs much longer execution times.

In the next chapter, we will tackle the dynamic nature of the CHU's SCS problem by solving the deterministic dynamic version of the problem, where the list of surgeries to schedule is not known in advance.

DETERMINISTIC DYNAMIC SURGICAL CASE SCHEDULING PROBLEM

4.1 Introduction

In the previous chapter, we solved the deterministic static version of the SCS problem. Despite the good results that were obtained, the method itself can not be applied at the CHU because of the dynamic nature of the problem, which comes from the uncertainty in patients arrival. In the OSU, each surgeon is responsible for managing and fixing the date of his/her surgeries during the consultation. This lack of global planning leads to poor resources management and higher costs.

In this chapter, we propose an approach to solve the Deterministic Dynamic SCS problem in which, during the consultation, the surgeons indicate only a due date for the surgeries. These surgeries are then added to a waiting list in order to be scheduled later. In other words, we address the problem of selecting iteratively a subset of patients from a given waiting list and assigning them a surgery date and an OR.

The objective of this new work flow is to schedule as much patients from the waiting list as possible, while minimizing the operational costs. For this, minimising the number of opened rooms is not an objective any more. Instead, we will consider the patients perspective in the form of minimising the surgeries tardiness. Hence, the considered objectives are first to maximise the number of scheduled patients, then minimise the total overtime at the OSU, then minimise the total cost of urgent and priority kits processed at the sterilizing unit, and finally minimise the total tardiness of surgeries from their due dates.

4.2 Technical background

In this section, we will provide a technical review on the dynamic scheduling problem. Due to the scarce number of papers on the dynamic SCS problem, we will study also the dynamic scheduling problem in production and manufacturing as both problems are similar. A review about the dynamic scheduling problem in manufacturing can be found in [Ouelhadj and Petrovic, 2009].

4.2.1 Dynamic scheduling problem

In the literature, dynamic scheduling problems concern scheduling problems where the problem characteristics change during the solving process due to real-time events. These real-time events are classified into two categories [Cowling and Johansson, 2002, Vieira et al., 2003] namely Resource-related and Job-related. In the case of dynamic SCS where the ORs can be considered as machines and surgeries as jobs, example of such events are:

1. **Resource-related:** OR closing, staff unavailability, medical kits failure or shortages, delays in the arrival of the medical kits, etc.
2. **Surgery-related:** Surgery cancellations, new surgeries arrival, etc.

Following the requirements of the SCS problem of the CHU, we are only concerned with surgery-related events. In the next section, we will explore the different approaches proposed in the literature to solve the deterministic dynamic scheduling problems.

4.2.2 Solution approaches

The main solution approaches found in the literature to solve the deterministic dynamic scheduling problems are called "*reactive approaches*" [Aytug et al., 2005, Herroelen and Leus, 2005]. They can be classified into two categories depending on whether a base-line (starting) schedule is generated in advance or not.

In the first category, no base-line schedule is generated in advance and decisions are made locally in real-time [Ouelhadj and Petrovic, 2009]. Priority dispatching rules are frequently used to select the next job (surgery) with the highest priority to be processed for a waiting list that contains jobs that are waiting for the machine to be free. These dispatching rules are quick and easy to implement and the priority of a job is determined based on the attributes of the job and the machine. Despite the simplicity of these rules, it is hard to predict the system performance as decisions are made locally in real-time. Thus, global scheduling has the potential to significantly improve shop performance compared to these dispatching rules.

On the other hand, second category methods use a base-line schedule that was computed beforehand and then implement a rescheduling strategy that reacts to real-time events. The application of such methods leads to two main issues that need to be addressed: when to reschedule, and how to reschedule?

- **How to reschedule**

There are two strategies in the literature that are used for rescheduling namely: schedule repair and complete rescheduling [Sabuncuoglu and Bayiz, 2000, Cowling and Johansson, 2002, Vieira et al., 2003].

Schedule repair refers to some local adjustment of the current schedule and may be preferable because of the potential saving in CPU times and the stability of the system is preserved [Ouelhadj and Petrovic, 2009].

Complete rescheduling recreates a new schedule from scratch. In theory, complete rescheduling might be better in reserving optimal solutions, but these solutions require expensive computation time and can lead to instability in the overall schedules. The benefits of the schedule repair over the complete rescheduling in most reactive scheduling systems was reported in [Sun and Xue, 2001]. In addition, the authors in [Sabuncuoglu and Bayiz, 2000] showed that schedule repair has better potential success over the complete rescheduling in terms of schedule stability and computation times.

The problem of whether to use schedule repair or to regenerate the whole schedule from scratch has been addressed in the literature. In [Cowling and Johansson, 2002], the authors used utility and stability measures to estimate the performance of various schedule repair and complete scheduling strategies, and then to select the best rescheduling strategy. On the other hand, the authors in [Jensen, 2001] used robustness measures (efficiency and stability) to chose the best rescheduling strategy to apply.

- **When to reschedule**

There are three policies in the literature regarding this issue namely: event driven, periodic, and hybrid [Sabuncuoglu and Bayiz, 2000, Vieira et al., 2003]. Both periodic and hybrid policies are identified in the literature under the name *rolling time horizon* [Church and Uzsoy, 1992, Vieira et al., 2000a, Aytug et al., 2005].

In event driven policy, the rescheduling is triggered whenever an unexpected event occurs that changes the current system status. This is the most used policy in dynamic scheduling approaches. In [Vieira et al., 2000a], the authors described analytical models to estimate the performance of a single machine system under periodic and event driven rescheduling strategies in an environment where jobs arrive dynamically. They proposed to evaluate the performance of periodic rescheduling and event driven rescheduling using analytical models that can easily and quickly estimate important performance measures, such as average flow time and machine utilization. In [Vieira et al., 2000b], they extended that study by investigating parallel machine systems. It was shown that rescheduling frequency can significantly affect the system performance (average flow time). A lower rescheduling frequency lowers the number of set ups. A higher rescheduling frequency allows the system to react more quickly to disruptions but may increase the number of set-ups.

In the periodic policy, schedules are generated at regular intervals, which gather all available information from the shop floor. The dynamic scheduling problem is decomposed into a series of static problems that can be solved by using classical scheduling algorithms. The schedule is then executed and not revised until the next period begins, where the planning horizon is renewed by taking into account new information gathered from the current shop floor status. The periodic policy yields more schedule stability and less schedule nervousness. Unfortunately, following an established schedule in the face of significant changes in the shop floor status may compromise performance since unwanted products or intermediates may be produced. Determining the rescheduling period is also a difficult task.

A hybrid policy reschedules the system periodically and also when an exception occurs. Events usually considered are machine breakdowns, arrival of urgent jobs, cancellation of jobs,

or job priority changes. In [Church and Uzsoy, 1992], the authors developed a hybrid event driven rescheduling policy for rescheduling in a single machine and parallel machine environment with dynamic job arrivals. Their system does rescheduling periodically. Events classified as regular occurring between periodic rescheduling are ignored until the next rescheduling moment. However, when an event is classified as urgent, complete rescheduling is immediately performed. The results indicated that the performance of periodic scheduling deteriorates as the length of rescheduling period increases, while event driven method achieves a reasonably good performance.

For the problem of the CHU, rather than individually scheduling each new surgery at the consultation step, we propose to use a periodic rescheduling scheme. In this approach, the new consulted surgeries will be added to the waiting list and at the end of each period (a week), we will schedule all the surgeries from the waiting list. The goal of this change is to allow for better global optimisation. Such approach is known in the literature as a "Rolling Horizon approach". In the remainder of this section, we will provide a technical review focused on this approach.

4.2.3 The rolling horizon approach

The Rolling horizon algorithms are based on separating the scheduling problem in a sequence of iterations, each of which models only part of the planning horizon in detail ("the detailed time block"), while the rest of the horizon ("the aggregate time block") is represented in an aggregate manner [Dimitriadis et al., 1997]. In principle, this approach may produce close to optimal solutions with a significant reduction of the computational requirements.

In [Baker, 1977], the authors conducted an experimental study of the effectiveness of rolling horizon decision making in production planning. They noted that while most existing formulations in the production planning literature are finite horizon models, the production planning problems themselves occur in systems that will operate indefinitely. They suggested that there are two principal reasons why finite horizon models might be appropriate for decision-making in infinite horizon problems. First, the forecasts for the remote future tend to be unreliable and are, therefore, of limited usefulness. Second, the decisions must for practical reasons be based on limited information about the future. The purpose of their study was to use simulation to investigate the efficiency of decisions obtained from optimizing a finite multiperiod problem with concave costs and implementing those decisions on a rolling basis. The study suggested, with exceptions however, that rolling schedules are quite efficient.

More recently, the authors in [Guimaraesa et al., 2015] applied the rolling horizon on a lot sizing and scheduling problem. They decomposed the horizon in two parts: the initial periods explicitly consider the production sequences to obtain detail schedules, while in the remaining periods a rough plan is generated to give an estimation of future costs and capacity. They showed that the business practice establishes that the improved models and solutions that integrate the lot sizing and scheduling problems together are likely to be applied in a rolling horizon basis. Only the first part of the plan is actually implemented, corresponding to the initial time periods. The remaining part serves the purposes of estimating future costs and capacity shortages, in order to account for their impact on the nearer decisions. They incorporated these principles behind rolling planning on lot sizing and scheduling using two distinct approaches. One explores the idea of an internal rolling scheduling using an implicit time decomposition of rolling planning to be able to efficiently handle large instances. The other focuses on the external rolling horizon defined by the successive planning steps to develop

efficient mathematical formulations that trade-off the plans detail and the computational effort. These two approaches share a common time structure partitioning the planning horizon into three parts: fixed, detailed and simplified horizons (“the aggregate time block”). The fixed horizon is associated with previous iterations of the method or previous planning steps. The detailed horizon embeds at least all the time periods to be implemented in the next iteration and uses an accurate model to express the problem. The simplified horizon uses an approximation of the exact model.

The key idea of this iterative approach was detailed in [Dimitriadis et al., 1997] as:

1. The planning horizon is divided into two Time Blocks (TBs). The first TB (detailed) is modelled through a discrete time function, while the second TB (aggregate) is done using aggregate formulations. At the first iteration, the detailed TB spans a relatively small part of the entire planning horizon.
2. The corresponding MILP is solved to optimality. The algorithm stops if the detailed TB covers already the entire time horizon.
3. Some of the variables of the detailed TB are fixed to their current optimal values from the obtained optimal solution for all following iterations.
4. Increase the size of the detailed TB by a number of time periods, and decrease the size of the aggregate TB by an equal amount.
5. Repeat from Step (2).

One big challenge of this approach is to determine the set of variable that will be fixed at step 3. One option is to fix all variables in the detailed TB. However, this is considered to be unnecessarily restrictive, as the only goal of fixing variables is to reduce the computational complexity of the MILP from step 2 in the following iterations. Since this complexity is a function of the number of discrete decisions, then it may be sufficient to only fix these decisions while allowing the corresponding continuous variables to vary. The second option allows for re-assessing at each iteration any continuous decisions made at earlier iterations, which potentially compensates for any inaccuracies caused by using the aggregate formulation.

This scheme is outlined in Figure 4.1 for a 4 week long scheduling problem. At each iteration, the size of the detailed TB is increased by 1 week.

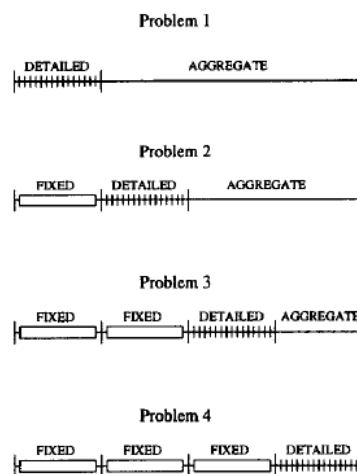


Figure 4.1: Rolling horizon algorithm as in [Dimitriadis et al., 1997]

4.3 Formal problem presentation

In the current work flow at the OSU (see Section 1.5), each surgeon is responsible of scheduling his/her own surgeries at the OSU. At the end of the consultation, the surgeons defines:

1. The date of the surgery.
2. Its estimated duration.
3. The list of required kits.
4. The type of the surgery (Ambulatory or normal).

As explain in Section 1.6, this myopic approach is responsible in particular for the large numbers of urgent and priority kits at the SU. In order to overcome this limitation, we propose to use a rolling horizon approach in which no actual surgery scheduling happens at the consultation, but instead the surgeon defines a due date for the surgery, and the surgery is added to a waiting list. Finally, at the end of each week t , the surgeries in the waiting list are scheduled starting from week $t + 3$. This 3 weeks gap assures that patients have enough time to prepare and organize their hospitalization.

Our rolling horizon framework consists of two elements:

1. The waiting list updater.
2. The scheduler unit.

The relationship between the two elements are described in Figure 4.2, where at each iteration, the waiting list updater prepares the waiting list surgeries and then this list is given to the scheduler unit in order to generate the schedule as described in Figure 4.3.

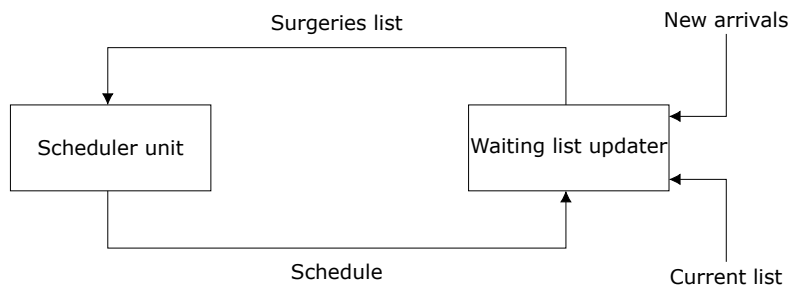


Figure 4.2: DDSCS scheduling framework

In other words, our framework works at the end of each week t as follows (see Figure 4.3):

1. Add the surgeries consulted at the current week t to the waiting list.
2. Calculate the horizon length for each surgeon T_s based on their surgeries load and shifts.
3. For each surgeon s , schedule the surgeries in the list over the horizon starting from week $t + 3$ and ending at $t + T_s$ using a MILP.
4. Fix the surgeries that were scheduled at week $t + 3$ and return the surgeries that were scheduled after $t + 3$ to the list.

In this procedure, the horizon length of each surgeon is calculated as follows:

1. Calculate the current load (total surgeon surgeries planned duration).
2. From $t + 3$, open shifts for the surgeon until their load is covered.
3. Increase the horizon by h_s

The reason why we are increasing the length of the horizon for each surgeon by h_s is that we can't know for sure the exact amount of shifts needed for each surgeon due to the fact that surgeons share some ORs occasionally, which makes it impossible to calculate the exact amount of time allocated for each surgeon.

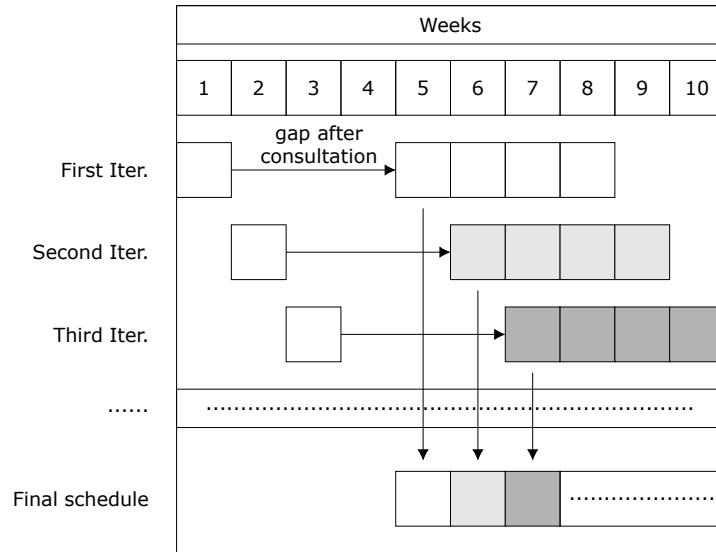


Figure 4.3: DDSCS rolling horizon scheme

4.4 Mathematical formulation

At each iteration, the scheduler unit solves the problem of scheduling all the surgeries in the waiting list over the horizon. In other words, a set of elective patients O is to be scheduled over a horizon of T_s days. The horizon length T_s represents the minimum number of days (shifts) that are required for each surgeon s to schedule all of his surgeries from the waiting list. For each surgery $i \in O$ a due date d_i is given in addition to its estimated duration p_i .

The value of the horizon length T_s can not be measured accurately since some surgeons share some of the ORs and we can not anticipate the exact amount of time each surgeon needs at the shared room. For this, we will introduce a form of relaxation represented in h_s which indicates the number of extra days (shifts) we add for the surgeon s over the length of his horizon T_s in order to surely cover all of his surgeries load.

The scheduling problem is solved using a modified version of the SCS problem MILP presented in section 3.2. For clarity sake, the changes in the model (parameters, variables, and constraints) are marked with a "*". These changes concern the addition of patients tardiness, the exclusion of ORs number as an objective, and removing the need to schedule all surgeries.

Parameters:

- * h_s number of extra shifts opened for surgeon s
- * T total number of days in the horizon (= $max(T_s)$)
- J number of periods in day
- K total number of kit types
- O total number of operations
- R total number of operating rooms
- S total number of surgeons
- p_i duration of operation i
- * d_i due date for operation i
- λ_s set of operations for surgeon s
- q_{ik} required quantity of kits of type k for operation i
- Q_k total quantity of kits of type k owned by the block
- cu urgent kit penalty
- cp priority kit penalty
- d_{rt}^j duration of period j of day t for room r (0 if the room is closed that day)
- d_{rt}^{bf} duration from period b to period f ($\geq b$) of day t for room r
- δ_{srt} binary parameter equal to 1 if surgeon s can use room r on day t , 0 otherwise
- a_i binary parameter equal to 1 if operation i is ambulatory, 0 otherwise
- A_{max} latest time for ambulatory operations to be performed at
- ε_{max} maximum allowed overtime for any room on any given day
- u_γ binary parameter equal to 1 if $\gamma = J$ and 0 otherwise

The decision variables:

- w_{itr} binary variable equal to 1 if operation i is scheduled at day t in room r
- x_{itr}^{bf} binary variable equal to 1 if surgery i begins at period b and finishes at f , on day t , in room r
- ε_{tr} integer variable representing the total overtime in room r at day t
- U_{str} binary variable equal to 1 if surgeon s is using room r at day t
- E_{tk} integer variable representing the total urgent kits of type k at day t
- Y_{tk}^1 integer variable representing the total priority kits (case 1) of type k at day t

Y_{tk}^2 integer variable representing the total priority kits (case 2) of type k at day t

* C_i integer variable representing the completion date t for surgery i

* T_i integer variable representing the tardiness of operation i

The constraints are :

In opposition to our MILP presented in 3.4, not all surgeries here can be scheduled. For this, Constraints (4.1) ensure that each surgery is scheduled at most once if at all.

$$* \sum_{t=1}^T \sum_{r=1}^R w_{itr} \leq 1, \quad \forall i \in \{1, \dots, O\} \quad (4.1)$$

$$\sum_{b=1}^J \sum_{f=b}^J x_{itr}^{bf} = w_{itr}, \quad \forall i \in \{1, \dots, O\}, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\} \quad (4.2)$$

$$\sum_{i=1}^O \sum_{b=1}^{\gamma} \sum_{f=\gamma+1}^J x_{itr}^{bf} \leq 1, \quad \forall \gamma \in \{1, \dots, J-1\}, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\} \quad (4.3)$$

Note that we removed from constraints 4.4 the variable L_{tr} (binary variable =1 if OR r is opened at day t and =0 otherwise) as minimising the total number of opened ORs is not an objective in this MILP.

$$* \sum_{i=1}^O \sum_{b=\beta}^{\gamma} \sum_{f=b}^{\gamma} p_i \cdot x_{itr}^{bf} + \sum_{i=1}^O \sum_{b=1}^{\beta-1} \sum_{f=\gamma+1}^J d_{rt}^{\beta\gamma} \cdot x_{itr}^{bf} \leq d_{rt}^{\beta\gamma} + u_{\gamma} \cdot \varepsilon_{tr}, \quad (4.4)$$

$$\forall \beta \in \{1, \dots, J\}, \forall \gamma \in \{\beta, \dots, J\}, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\}$$

$$w_{itr} \leq U_{srt}, \quad \forall s \in \{1, \dots, S\}, \forall i \in \lambda_s, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\} \quad (4.5)$$

$$U_{srt} \leq \delta_{srt}, \quad \forall s \in \{1, \dots, S\}, \forall r \in \{1, \dots, R\}, \forall t \in \{1, \dots, T\} \quad (4.6)$$

$$\sum_{r=1}^R U_{srt} \leq 1, \quad \forall s \in \{1, \dots, S\}, \forall t \in \{1, \dots, T\} \quad (4.7)$$

$$\sum_{i=1}^O \sum_{r=1}^R q_{ik} \cdot w_{itr} \leq Q_k, \quad \forall t \in \{1, \dots, T\}, \forall k \in \{1, \dots, K\} \quad (4.8)$$

$$\sum_{i=1}^O \sum_{r=1}^R q_{ik} \cdot \left(\sum_{b=1}^J x_{itr}^{bJ} + \sum_{b=1}^2 \sum_{f=b}^J x_{i(t+1)r}^{bf} \right) \leq Q_k, \quad (4.9)$$

$$\forall t \in \{1, \dots, T-1\}, \forall k \in \{1, \dots, K\}$$

$$\sum_{i=1}^O \sum_{r=1}^R q_{ik} \cdot \left(\sum_{f=2}^J \sum_{b=1}^f x_{itr}^{bf} + \sum_{b=1}^2 \sum_{f=b}^J x_{i(t+1)r}^{bf} \right) - Q_k \leq E_{tk}, \quad (4.10)$$

$$\forall t \in \{1, \dots, T\}, \forall k \in \{1, \dots, K\}$$

$$\sum_{i=1}^O \sum_{r=1}^R q_{ik} \cdot \left(\sum_{f=1}^J \sum_{b=1}^f x_{itr}^{bf} + \sum_{b=1}^2 \sum_{f=b}^J x_{i(t+1)r}^{bf} \right) - Q_k - E_{tk} \leq Y_{tk}^1, \quad (4.11)$$

$$\forall t \in \{1, \dots, T\}, \forall k \in \{1, \dots, K\}$$

$$\sum_{i=1}^O \sum_{r=1}^R q_{ik} \cdot \left(\sum_{f=2}^J \sum_{b=1}^f x_{itr}^{bf} + \sum_{b=1}^J \sum_{f=b}^J x_{i(t+1)r}^{bf} \right) - Q_k - E_{tk} \leq Y_{tk}^2, \quad (4.12)$$

$$\forall t \in \{1, \dots, T\}, \forall k \in \{1, \dots, K\}$$

$$\sum_{i=1}^O \sum_{b=\beta}^2 \sum_{f=b}^2 p_i \cdot x_{itr}^{bf} + \sum_{i=1}^O \sum_{b=\beta}^3 a_i \cdot p_i \cdot x_{itr}^{b3} \leq A_{max} - \sum_{j=1}^{\beta-1} d_{rt}^j, \quad (4.13)$$

$$\forall \beta \in \{1, \dots, 3\}, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\}$$

$$\sum_{b=1}^J a_i \cdot x_{itr}^{bJ} = 0, \quad \forall i \in \{1, \dots, O\}, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\} \quad (4.14)$$

$$\varepsilon_{tr} \leq \varepsilon_{max}, \quad \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\} \quad (4.15)$$

The expression of surgeries completion date C_i is given in constraints (4.16) and expression of surgeries tardiness T_i is given in constraints (4.17).

$$* \quad C_i \geq \sum_{t=1}^T \sum_{r=1}^R (t \cdot w_{itr}), \quad \forall i \in \{1, \dots, O\} \quad (4.16)$$

$$* \quad T_i \geq C_i - d_i, \quad \forall i \in \{1, \dots, O\} \quad (4.17)$$

Finally, constraints (4.18) to (4.26) define the range of the variables.

$$w_{itr} \in \{0, 1\}, \quad \forall i \in \{1, \dots, O\}, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\} \quad (4.18)$$

$$x_{itr}^{bf} \in \{0, 1\}, \quad (4.19)$$

$$\forall i \in \{1, \dots, O\}, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\}, \forall b \in \{1, \dots, J\}, \forall f \in \{b, \dots, J\}$$

$$\varepsilon_{tr} \geq 0, \quad \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\} \quad (4.20)$$

$$U_{srt} \in \{0, 1\}, \quad \forall s \in \{1, \dots, S\}, \forall r \in \{1, \dots, R\}, \forall t \in \{1, \dots, T\} \quad (4.21)$$

$$E_{tk} \geq 0, \quad \forall t \in \{1, \dots, T\}, \forall k \in \{1, \dots, K\} \quad (4.22)$$

$$Y_{tk}^1 \geq 0, \quad \forall t \in \{1, \dots, T\}, \forall k \in \{1, \dots, K\} \quad (4.23)$$

$$Y_{tk}^2 \geq 0, \quad \forall t \in \{1, \dots, T\}, \forall k \in \{1, \dots, K\} \quad (4.24)$$

$$* \quad T_s \geq C_i \geq 0, \quad \forall s \in \{1, \dots, S\}, \forall i \in \{1, \dots, O\} \quad (4.25)$$

$$* \quad T_i \geq 0, \quad \forall i \in \{1, \dots, O\} \quad (4.26)$$

The multiple objective functions (4.27) are taken into account by using a lexicographic method.

We start by first maximising f_1 the total number of scheduled surgeries from the waiting list. We then add the resulting value of $\sum_{i=1}^O \sum_{t=1}^T \sum_{r=1}^R w_{itr}$ as a lower bound in the model and move to minimising f_2 the total overtime required to schedule at least the number of scheduled surgeries found in f_1 . We then add the resulting value of $(\sum_{t=1}^T \sum_{r=1}^R \varepsilon_{tr})$ as an upper bound in the model and we move to minimising f_3 the total penalty cost of urgent and priority kits without scheduling less surgeries than what was found for objective f_1 nor exceeding the total overtime found for objective f_2 . We then add the value of $(\sum_{t=1}^T \sum_{k=1}^K [cu \cdot E_{tk} + cp \cdot (Y_{tk}^1 + Y_{tk}^2)])$ as an upper bound in our model and we minimise f_4 the total tardiness of surgeries without scheduling less surgeries than what we found in f_1 nor exceeding the total overtime calculated for f_2 nor the total penalty cost of urgent and priority found for f_3 .

$$Lex \left(\begin{array}{l} f_1 : \text{Maximise } \sum_{i=1}^O \sum_{t=1}^T \sum_{r=1}^R w_{itr}; \\ f_2 : \text{Minimise } \sum_{t=1}^T \sum_{r=1}^R \varepsilon_{tr}; \\ f_3 : \text{Minimise } \sum_{t=1}^T \sum_{k=1}^K [cu \cdot E_{tk} + cp \cdot (Y_{tk}^1 + Y_{tk}^2)]; \\ f_4 : \text{Minimise } \sum_{i=1}^O T_i \end{array} \right) \quad (4.27)$$

4.5 Experimental results

We used the data described in Section 1.8 as a single instance to test our method. Since the due dates of surgeries that we described in our suggested approach do not exist in the provided data, we will use the actual dates of surgeries as due dates. The characteristics of this single instance are as follows:

- Total number of surgeries $O = 2000$
- 69 OSU surgeries performed outside the block (considered for their kits).
- Horizon: 44 weeks.
- Total number of surgeons $S = 15$
- Due dates: actual date of the surgery.

Since the data that we have contains the schedule from 1/9/2014 till 30/6/2015 and we consider a gap of 3 weeks between the consultation and the start of schedule, we start our method from 11/8/2014, where all the surgeries that were consulted before this date are placed in the initial waiting list. Figure 4.4 shows the number of consulted surgeries per week in addition to the number of surgeries in the initial waiting list.

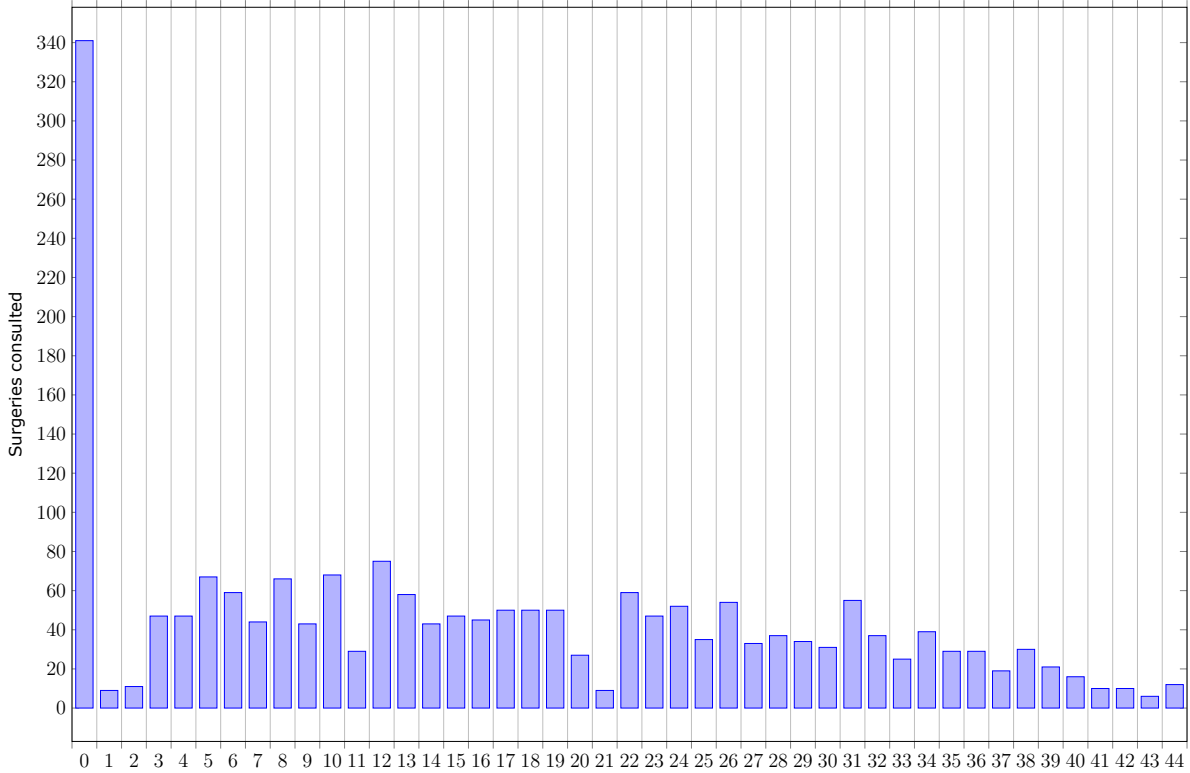


Figure 4.4: Consulted surgeries per week (initial waiting list at 0)

We performed 4 experiments using different values for the parameters as follows:

1. **Run 1:** $\varepsilon_{max} = 180$ min, $h_s = 2$
2. **Run 2:** $\varepsilon_{max} = 180$ min, $h_s = 4$
3. **Run 3:** $\varepsilon_{max} = 90$ min, $h_s = 2$
4. **Run 4:** $\varepsilon_{max} = 90$ min, $h_s = 4$

Where:

- ε_{max} is the maximum allowed overtime at each room r at each day t .
- h_s is the number of extra shifts opened for each surgeon s .

Our testing environment is:

- CPU: Intel Core i3-2120 @ 3.30GHz
- OS: Windows 7 64 bits
- RAM: 8 GB

- Solver: CPLEX 12.5
- Time limit: 4 hour per iteration (1 hour per objective).

Table 4.1 shows the planned schedules obtained with the 4 runs and the one of the OSU.

| Schedule | # Scheduled | Overtime | Max. overtime | # opened ORs | Min. Occ. rate | Avg. Occ. rate | # Urgent | # Priorities | $\sum T_i$ | Max T_i | Avg T_i |
|--------------|-------------|----------|---------------|--------------|----------------|----------------|----------|--------------|------------|-----------|-----------|
| Run 1 | 1973 | 5829 | 180 | 527 | 29.7% | 81.5% | 10 | 126 | 13221 | 121 | 12 |
| Run 2 | 1981 | 3181 | 177 | 530 | 32% | 82.5% | 2 | 34 | 7279 | 77 | 9 |
| Run 3 | 1966 | 2322 | 90 | 528 | 22.8% | 81.6% | 9 | 108 | 10874 | 145 | 11 |
| Run 4 | 1974 | 1547 | 90 | 530 | 43.5% | 82.2% | 0 | 30 | 7516 | 121 | 9 |
| OSU | 2000 | 8041 | 266 | 539 | 9.5% | 80.9% | 66 | 626 | | | |

Table 4.1: DDSCS planned schedule comparison

We start the comparison with the total number of schedules surgeries as shown in column '*# Scheduled*'. In average, our method managed to schedule over the period of 10 months around 98.7% of the total 2000 surgeries among the 4 runs with run 2 managing to schedule the most surgeries (1981 that is around 99% of the total 2000 surgeries). Next, column '*Overtime*' represents the total overtime found in each schedule. All of our 4 runs managed to use lower overtime than the OSU. In addition, using an $\varepsilon_{max} = 90$ as in run 3 and 4 yields the lowest overtime used in all of the schedules with run 4 decreasing the used overtime by 80.8% over the OSU. Similarly, all of our runs has less maximum overtime since it is a hard constraint as shown in column '*Max. overtime*'. To better understand the behaviour of our model, Figure 4.5 shows the overall overtime for our solutions and the one of the OSU. From this, we can clearly see that the overtime in our method substantially increases near the end of the horizon. This situation should not happen in real life as there should not be an end for the scheduling horizon.

Despite that we are not currently minimising the total number of opened ORs, the results reported in column '*# opened ORs*' shows that our 4 runs managed to decrease the number of used rooms by around 10 rooms on average, with run 1 managing the biggest decrease with a total of 12 closed ORs. Note that the model was unable to use these ORs to schedule more surgeries due to conflict in kits constraints, and the mandatory 3 weeks gap that we are leaving between the consultation and the earliest possible date for any surgery. This can be further justified by the fact that when giving the solver a higher flexibility margin (h_s), it managed to use more ORs to schedule more surgeries (as in Run 2 and 4). In addition, column '*Min. Occ. rate*' shows the minimum occupancy rate found at a room in each schedule. By comparing these results, we can clearly see that using a value of $h_s = 4$ yields the best minimum occupancy rates as runs 2 and 4 has the highest. The same also can be said about the average occupancy rates shown in column '*Avg. Occ. rate*', where all of our tests have better values than the OSU, with runs 2 and 4 having the highest among all results.

Next, column '*# Urgent*' shows the total number of urgent kits and column '*# Priorities*'

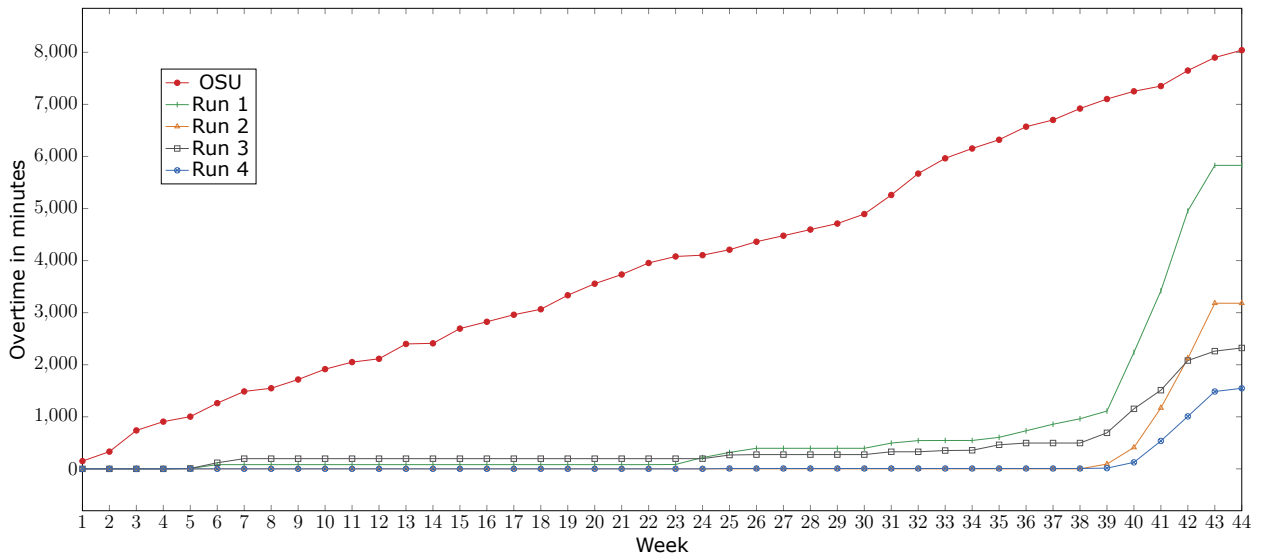


Figure 4.5: DDSCS planned cumulative overtime comparison

shows the total number of priority kits found in each schedule. Following the same behaviour of the model, using a value of $h_s = 4$ yields the best results as run 2 managed a decrease of 94.8% for the total number of problem kits (urgent + priorities) over the schedule of the OSU, while run 4 managed the best results with a decrease of 95.7% over the OSU. This can be seen in depth with Figure 4.6, which shows the total number of problem kits per week for each schedule.

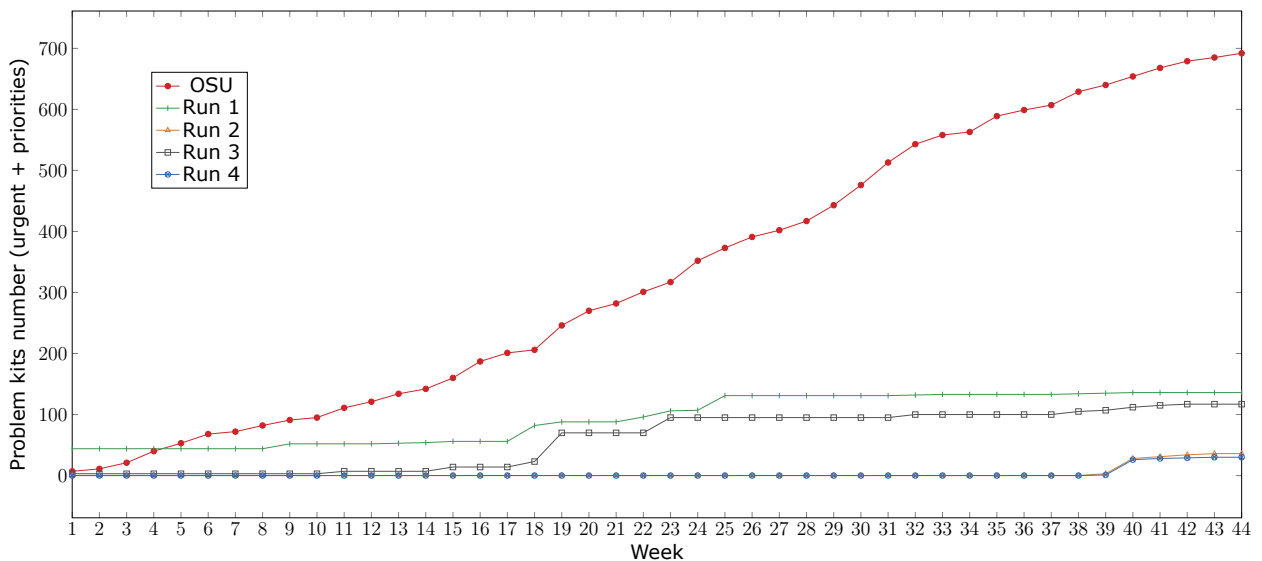


Figure 4.6: DDSCS planned cumulative urgent and priority kits comparison

Finally, we compare the tardiness of surgeries from their due dates measured in number of days. The reason behind the huge numbers shown in the total tardiness (column ' $\sum T_i$ ') in our results is the fact that using the actual date of surgery as a due date gave a very small margin of maneuver for most surgeries to be scheduled if any. This is also supported by the fact that 247 surgeries out of the 2000 total surgeries (around 12.3%) have less than 3 weeks between their consultation date and actual surgery date (due date), which makes them already

tardy even before beginning the scheduling process. Again we can see that runs 2 and 4 have the best results when comparing the total tardiness, maximum tardiness (column ' $Max T_i$ '), and average tardiness (column ' $Avg T_i$ '), with run 2 having the lowest values for the three criteria over all of our runs.

Through this results analysis, we can see that our method provided significant improvements in the given time limit over the one currently implemented at the OSU in terms of the total overtime used and the number of urgent and priority kits. In addition, we saw that this method performs better when giving the model a higher flexibility margin (h_s) as Run 2 and 4 performed better than the rest.

4.6 Conclusion

In this chapter, we tackled the deterministic dynamic surgical case scheduling problem of the CHU. Formally, the problem consists in selecting a subset of patients from a given waiting list and assigning them a surgery date and an OR.

We started by providing a technical background for the dynamic scheduling problem and rolling horizon method with some insights from the dynamic scheduling problem in production and manufacturing. We then presented a formal description of the problem, where we explained the dynamic nature of the scheduling problem at the OSU and presented a new work flow for the OSU in which, during the consultation, the surgeons indicate only a due date for the surgeries instead of the actual surgery date as in their current work flow. These surgeries are then added to a waiting list in order to be scheduled later. Then, we presented a MILP formulation for the problem, where the objectives are first to maximise the number of scheduled patients, then minimise the total overtime at the OSU, then minimise the total cost of urgent and priority kits processed at the sterilizing unit, and finally minimise the total tardiness of surgeries from their due dates. Finally, we presented our numerical experiments in the form of planned schedules comparisons. Our method proved to be better than the current applied approach at the OSU in terms of overtime, numbers of opened ORs, occupancy rates at the ORs and numbers of urgent and priority kits while being able to schedule around 99% of the total number of surgeries.

FROM PLANNED TO ACHIEVED SCHEDULES

5.1 Introduction

In chapters 3 and 4, we tackled the deterministic problem of the CHU in both static and dynamic settings. Our planned schedules showed great improvement over the ones of the OSU. However, these results were obtained on the basis of surgeries durations that are not easy to estimate. In order to better understand and analyse the performance of our methods, we compare the achieved schedules that uses surgeries real durations for both problems with the ones of the OSU.

In this chapter, we will start by presenting the procedure that generates the corresponding achieved schedule from an existing planned one. Next, we will present and analyse the achieved schedules obtained from the solutions of the deterministic static SCS problem and compare the results with the ones of the OSU. In addition, we will compare any degradation that happens between the planned and achieved schedules. We will also compare our achieved schedules with the best case scenario obtained by running our MILP directly with surgeries real durations.

Next, we will use the same algorithm to obtain the achieved schedules from the deterministic dynamic problem and compare these schedules to the ones of the OSU. Finally, we will study any degradation that we find when comparing the achieved schedules to their corresponding planned ones.

5.2 Achieved schedules generating algorithm

As the name suggests, the planned schedule relies on the estimated durations before the surgeries take place. In order to obtain the effective schedule based on the real durations, we implemented a procedure that simulates the work process of the OSU to create the corresponding achieved schedule from the planned schedule. This algorithm will go through the horizon simulating the start of each day and starting surgeries with their real durations. The goal is to eliminate the not allowed kits situation. The procedure fixes the start of the next surgery in the planning and allows the possibility to start the next surgery ($i + 1$) in place of i if i is waiting for a kit and $i + 1$ is not. But, it is not possible for a further surgery to start in place of i (i.e. $i + 3$ cannot start in place of i). This was done to stay true with the process of the CHU, as patients are prepared for their surgery not more than 1 hour in advance, so later patients in the queue ($> i + 1$) won't be ready to start directly.

In order to do this, we will keep track at any given time of:

- the occupancy of rooms,
- the kits current available quantities,
- the queue of surgeries to schedule (FIFO).

We will create an event queue which will hold events like:

- kit return event,
- surgery end event.

Algorithm 2 Achieved schedule simulation

```
1: for all  $t \in T$  do
2:    $currentTime \leftarrow 8 : 15$ 
3:   while  $surgeriesQueue$  is not empty do
4:     check events queue for events happening at  $currentTime$ 
5:     free resources based on the events found at  $currentTime$ 
6:     pop surgeries from  $surgeriesQueue$ 
7:     for all surgery  $i \in$  popped surgeries do
8:       if there is enough kits to schedule  $i$  then
9:         start surgery  $i$ 
10:      else
11:        if there is enough kits to schedule the next surgery after  $i$  in the same room
12:      then
13:        start the next surgery
14:      end if
15:    end if
16:    create surgery end event for started surgery (also adding the 20 minutes break)
17:    create kit return event for each kit for the started surgery
18:    decrease available quantities of used kits
19:  end for
20:   $currentTime \leftarrow currentTime + 1$ 
21: end while
22: end for
```

An outline for the algorithm is presented in Algorithm 2. In other word, this algorithm consists in replacing the planned duration of each surgery with the real duration and inserting

a break of 20 minutes between each two consecutive surgeries to clean the room (this break is already included in the planned durations). This algorithm provides a mean of keeping track of the current available number of each kit type and the current state of each room.

5.3 Deterministic Static SCS problem

In this section, we will focus on the Deterministic Static SCS problem. We start by presenting and analysing the generated achieved schedules for the planned schedules from 3.6.2. We then analyse the differences between those planned and achieved schedules.

5.3.1 Achieved schedules

Table 5.1 shows the achieved schedule of the OSU which represents what actually happened at the CHU, Table 5.2 shows the achieved schedule obtained using the MILP, and Table 5.3 shows the achieved schedule obtained using the heuristic method.

| Instance | Violated constraints | | Overtime | | ORs | | | Kits | |
|----------------|----------------------|----------------|---------------|-----|-------------|---------------|----------------|------------|-------------|
| | #late ambs | #violated kits | total | max | #opened | Min Occ. rate | Avg. Occ. rate | #urgent | #priorities |
| 1 | 14 | 5 | 1382 | 190 | 59 | 29.9% | 79.2% | 3 | 39 |
| 2 | 14 | 5 | 1692 | 227 | 59 | 12.9% | 76% | 3 | 44 |
| 3 | 6 | 10 | 1322 | 113 | 48 | 19% | 79.3% | 8 | 49 |
| 4 | 10 | 8 | 1800 | 98 | 48 | 44% | 82.9% | 14 | 64 |
| 5 | 14 | 5 | 2370 | 130 | 59 | 29.7% | 81.9% | 6 | 58 |
| 6 | 8 | 4 | 2140 | 201 | 52 | 50.8% | 79% | 1 | 53 |
| 7 | 6 | 13 | 1454 | 170 | 59 | 33.1% | 80.6% | 28 | 72 |
| 8 | 9 | 2 | 1584 | 100 | 49 | 23.7% | 81.5% | 4 | 46 |
| 9 | 10 | 7 | 1500 | 126 | 46 | 12.3% | 77.3% | 8 | 42 |
| 10 | 10 | 3 | 1930 | 326 | 60 | 9.9% | 70.7% | 4 | 27 |
| Average | 10.1 | 6.2 | 1717.4 | | 53.9 | | 78.8% | 7.9 | 49.4 |

Table 5.1: OSU achieved schedule.

| Instance | Violated constraints | | Overtime | | ORs | | | Kits | |
|----------------|----------------------|----------------|-------------|-----|-----------|---------------|----------------|------------|-------------|
| | #late ambs | #violated kits | total | max | #opened | Min Occ. rate | Avg. Occ. rate | #urgent | #priorities |
| 1 | 4 | 0 | 979 | 135 | 58 | 65.1% | 87.8% | 0 | 11 |
| 2 | 4 | 0 | 1268 | 189 | 55 | 31.4% | 89.7% | 0 | 24 |
| 3 | 3 | 0 | 1305 | 156 | 46 | 55.0% | 89.8% | 0 | 14 |
| 4 | 9 | 0 | 1745 | 130 | 48 | 53.9% | 90.7% | 1 | 17 |
| 5 | 5 | 0 | 1990 | 200 | 58 | 68.3% | 89.1% | 0 | 44 |
| 6 | 3 | 0 | 1034 | 206 | 52 | 71.4% | 89.4% | 0 | 34 |
| 7 | 1 | 0 | 965 | 125 | 59 | 48.1% | 89.5% | 5 | 31 |
| 8 | 3 | 0 | 1014 | 103 | 48 | 66.3% | 91.4% | 0 | 1 |
| 9 | 6 | 0 | 950 | 137 | 43 | 75.7% | 88.8% | 0 | 7 |
| 10 | 2 | 0 | 1060 | 240 | 53 | 54.3% | 87.0% | 0 | 6 |
| Average | 4 | 0 | 1231 | | 52 | | 89.3% | 0.6 | 18.9 |

Table 5.2: DSSCS MILP achieved schedule.

| Instance | Violated constraints | | Overtime | | ORs | | | Kits | |
|----------------|----------------------|----------------|---------------|-----|-------------|---------------|----------------|------------|-------------|
| | #late amb | #violated kits | total | max | #opened | Min Occ. rate | Avg. Occ. rate | #urgent | #priorities |
| 1 | 8 | 0 | 1604 | 182 | 58 | 26.6% | 86.9% | 0 | 18 |
| 2 | 6 | 0 | 1569 | 207 | 57 | 35.8% | 84.6% | 0 | 17 |
| 3 | 4 | 0 | 1576 | 158 | 47 | 57.1% | 88.2% | 1 | 13 |
| 4 | 12 | 0 | 1822 | 149 | 48 | 46.1% | 90.1% | 4 | 21 |
| 5 | 12 | 0 | 2116 | 256 | 59 | 44.8% | 84.2% | 1 | 39 |
| 6 | 5 | 0 | 1432 | 139 | 52 | 43.3% | 88.8% | 2 | 45 |
| 7 | 3 | 0 | 1314 | 226 | 59 | 47.7% | 87.4% | 3 | 44 |
| 8 | 7 | 0 | 1194 | 159 | 48 | 67.9% | 86.9% | 0 | 12 |
| 9 | 6 | 0 | 1349 | 153 | 44 | 66.3% | 87.4% | 2 | 19 |
| 10 | 1 | 0 | 747 | 325 | 56 | 23.5% | 83.4% | 0 | 18 |
| Average | 6.4 | 0 | 1472.3 | | 52.8 | | 86.8% | 1.3 | 24.6 |

Table 5.3: DSSCS heuristic achieved schedule.

We see that the MILP decreased the number of late ambulatories by approximately 60.4% compared with the OSU’s solution, while the heuristic method decreased it by approximately 36.6%. In addition, the OSU used 6.2 violated kits on average per month, while our method does not allow it. Likewise, the schedules obtained using the MILP managed to decrease the total overtime by approximately 28.3% compared with the OSU’s used overtime (savings of approximately 8 hours and 6 minutes on average per month), and heuristic based achieved schedules managed a decrease of approximately 14.3% (savings of approximately 4 hours and 5 minutes). In addition, the ‘max’ column shows the maximum overtime found in a day during the month considered. In both MILP and OSU’s solutions, 4 instances had a maximum overtime over the limit of 180 minutes set by the management of the CHU, while 5 instances had a maximum overtime over the 180 minutes limit in the heuristic solution.

Note that the number of rooms opened does not change since our method only uses rooms opened in the planned solution, thus the MILP schedules still save 1.9 ORs, and the heuristic schedules save 1.1 ORs on average per month over the OSU’s solution. But, the MILP schedules managed to achieve an average occupancy rate for the rooms opened of approximately 89.3% which is better than the average for both the heuristic approach (86.8%) and the OSU (78.8%). Similarly, the minimum occupancy rate found in the MILP schedules in 8 instances is greater than 50% (compared with 1 for the OSU and 3 for the heuristic method).

Concerning the last criterion, the MILP model generated approximately 0.6 urgent and 18.9 priority kits on average (19.5 problem kits per month), while these numbers are approximately 1.3 and 24.6 (25.9 problem kits per month) for the heuristic, and 7.9 and 49.4 (57.3 problem kits per month) for the OSU. Thus, the MILP model was able to decrease the number of urgent and priority kits by approximately 65.9% compared with the OSU’s solution, while the heuristic method managed a decrease of approximately 57.8%.

Similar to the planned schedules, we can see our achieved schedules (especially the ones obtained by the MILP) remain significantly better than the ones of the CHU.

5.3.2 Degradation

Despite the good results acquired using our method, we can note a significant degradation between the planned schedules and the achieved ones. This degradation is due to the big uncertainty regarding surgeries durations as shown in Section 1.8.3.

| | schedule | #late amb | #not allowed kits | overtime | #problem kits (urgent + priorities) |
|------------------|-----------------|------------------|--------------------------|-----------------|--|
| OSU | Planned | 7.6 | 10.2 | 804.1 | 69.2 |
| | Achieved | 10.1 | 6.2 | 1717.4 | 57.3 |
| | Degradation | +32.9% | -39.2% | +113.6% | -17.2% |
| MILP | Planned | 0 | 0 | 215.5 | 3.7 |
| | Achieved | 4 | 0 | 1231 | 19.5 |
| | Degradation | NA | - | +471.2% | +427% |
| Heuristic | Planned | 0 | 0 | 372 | 9.1 |
| | Achieved | 6.4 | 0 | 1472.3 | 25.9 |
| | Degradation | NA | - | +295.8% | +184.6% |

Table 5.4: DSSCS degradation analysis

Table 5.4 provides a summary for the difference between the planned and achieved schedules for both our MILP and heuristic methods and the OSU's solutions. Starting with the solutions of the OSU, we can see that the average numbers of late ambulatory surgeries increase from 7.6 to 10.1 surgeries per month (+32.9%), while the average number of not allowed kits decrease from 10.2 to 6.2 kits per month (-39.2%). Next, the average overtime increased from 804.1 to 1717.4 minutes per month (+113.6%). Finally, the average number of problem kits (urgent + priorities) decreased from 69.2 to 57.3 kits per month (-17.2%).

Concerning our MILP, 4 ambulatory surgeries were scheduled after 15:00 (late ambulatories) on average per month, while this number in our initial planned schedule was 0. Regarding the not allowed kits, our simulation method does not allow a surgery to start unless all required kits are available, and thus, the average number of not allowed kits stays at 0 for both our planned and achieved schedules. Next, the average overtime increases from 215.5 to 1231 minutes per month (+471.2%). Similarly, the average number of problem kits increases from 3.7 to 19.5 kits per month (+427%).

Finally, we note for our heuristic an average of 6.4 late ambulatory surgeries per month (from 0 in the planned schedule), while the average number of not allowed kits is 0 similar to our MILP results. Another quite big increase can be noted when analysing the average overtime in the heuristic solutions as it moves from 372 to 1472.3 minutes per month (+295.8%). Finally, the average number of problem kits increases from 9.1 to 25.9 kits per month (+184.6%).

Thus even if our solutions are indeed better than those of the OSU, we can note that the degradations are more present in ours. In other words, the better the planned solution is, the bigger the degradation. This probably can be explained by the fact that our solutions are more tightly packed and consequently are more sensible to uncertainties.

In order to calibrate the quality and robustness of our solutions, we ran our model on the instances using directly the real durations of the surgeries and included a 20 minutes gap between consecutive surgeries.

The results of this experiment are given in table 5.5. If we compare the new results with our original achieved schedule (table 5.2), we can see that the overtime can really be optimised (average of 265.3 minutes per month instead of 1231) if we knew in advance the real durations. Furthermore, it is possible to build a schedule with no problem kit nor late

ambulatory surgeries (at the cost of using one additional room).

| Instance | overtime | # rooms | # urgent | # priorities | # late amb |
|----------------|--------------|-------------|----------|--------------|------------|
| 1 | 79 | 58 | 0 | 0 | 0 |
| 2 | 350 | 55 | 0 | 0 | 0 |
| 3 | 291 | 46 | 0 | 0 | 0 |
| 4 | 395 | 48 | 0 | 0 | 0 |
| 5 | 679 | 58 | 0 | 0 | 0 |
| 6 | 100 | 52 | 0 | 0 | 0 |
| 7 | 55 | 59 | 0 | 0 | 0 |
| 8 | 389 | 48 | 0 | 0 | 0 |
| 9 | 279 | 44 | 0 | 0 | 0 |
| 10 | 36 | 53 | 0 | 0 | 0 |
| Average | 265.3 | 52.1 | 0 | 0 | 0 |

Table 5.5: Achieved schedule directly from MILP

5.4 Deterministic Dynamic SCS

In this section, we will study the results of the Deterministic Dynamic SCS problem. We start by presenting and analysing the generated achieved schedules from the planned schedules from 4.5. We then analyse the differences between those planned and achieved schedules.

5.4.1 Achieved schedules

Following the work process of the OSU, we applied our simulation method presented in Algorithm 2 to obtain the achieved schedules from the planned schedules obtained by our rolling horizon method.

| Schedule | # late amb | # Scheduled | Overtime | Max. overtime | # opened ORs | Min. Occ. rate | Avg. Occ. rate | # Urgent | # Priorities |
|--------------|------------|-------------|----------|---------------|--------------|----------------|----------------|----------|--------------|
| Run 1 | 13 | 1973 | 13304 | 371 | 527 | 28.7% | 86.5% | 17 | 213 |
| Run 2 | 12 | 1981 | 11346 | 408 | 530 | 39.3% | 88.5% | 2 | 181 |
| Run 3 | 11 | 1966 | 10822 | 257 | 528 | 32.8% | 85.9% | 3 | 236 |
| Run 4 | 12 | 1974 | 9443 | 258 | 530 | 45.4% | 87.8% | 7 | 152 |
| OSU | 101 | 2000 | 17174 | 326 | 539 | 9.9% | 78.8% | 79 | 494 |

Table 5.6: DDSCS achieved schedule comparison

The obtained achieved schedules are shown in Table 5.6. We start by comparing the numbers of late ambulatory surgeries shown in column '*# late amb*', where all 4 runs had better results than the ones of the OSU with an average of 12 late ambulatory surgeries among

all the runs and an average decrease of around 88.1% over the OSU. Next, the number of scheduled surgeries doesn't change here from the planned schedules since our simulation method doesn't add or remove surgeries from the planned schedule. Thus, run 2 still provides the best results with 1981 scheduled surgeries (around 99% of the total 2000 surgeries) and run 4 coming next with 1974 scheduled surgeries (around 98.7%).

Regarding the overtime, column '*Overtime*' shows the total overtime in minutes for each schedule. In addition to that all 4 runs had better values than the OSU and run 4 had the best overtime with a decrease of approximately 45% over the OSU (savings of 128 hours and 51 minutes), we can see that limiting the maximum overtime ε_{max} per room per day to a lower value (90 minutes instead of 180) rewards with better results as shown in runs 3 and 4 (both had $\varepsilon_{max} = 90$) compared with runs 1 and 2 (both had $\varepsilon_{max} = 180$). In addition, we can see that providing the solver with a larger flexibility margin represented by h_s gives better overtime when fixing ε_{max} , as shown with run 2 having better results than run 1, and run 4 having better results than run 3. Of course providing lower ε_{max} values yields lower maximum overtime in the end results as shown in column '*Max. overtime*' with runs 3 and 4 having the lowest values.

Similar to the number of scheduled surgeries, the total number of opened ORs shown in column '*# opened ORs*' doesn't change here from the planned schedules. Next, column '*Min. Occ. rate*' shows the minimum occupancy rate found at a room in each schedule. Similar to the planned schedules comparison, using a value of $h_s = 4$ yields the best minimum occupancy rates as runs 2 and 4 has the highest values, as well as a better average occupancy rates as shown in column '*Avg. Occ. rate*' with runs 2 and 4 having the best results. In addition, all of our schedules provided an average occupancy rate greater than the 80% goal fixed by the CHU, which is not the case for the schedule of the OSU.

Finally, column '*# Urgent*' shows the total number of urgent kits and column '*# Priorities*' shows the total number of priority kits found in each schedule. Again, using a value of $h_s = 4$ yields the best results as run 2 had a total number of problem kits (urgent + priorities) of 183 kits with a decrease of approximately 68% over the 573 kits obtained by the schedule of the OSU, while run 4 managed the best results with 159 problem kits equal to a decrease of approximately 72.3% over the OSU.

Similar to the planned schedules, we can see all of our achieved schedules still provide significant improvements over the solution of the CHU. In addition, these achieved schedules are coherent to the planned ones, where giving the solver a higher margin of flexibility yields better results (as in runs 2 and 4).

5.4.2 Degradation

When comparing the differences between the planned and achieved schedules, it becomes clear that there is a significant degradation for all 4 runs we used to test the method. Table 5.7 summarises the differences between the planned and achieved schedules for each of the 4 runs.

Starting with the overtime, we can see that all 4 runs are less robust than the OSU's solution. Of course this is justified by the fact that all 4 runs had better planned values compared to the OSU. This can be further supported as run 2 and 4 had the best planned values and thus suffer from the most degradation.

The same behaviour continues when comparing the total number of urgent and priority kits, where all 4 runs had worst robustness than the CHU with run 2 and 4 having the worst

| | schedule | #late amb s | #not allowed kits | overtime | #problem kits (urgent + priorities) |
|--|-----------------|--------------------|--------------------------|-----------------|--|
| OSU | Planned | 76 | 102 | 8041 | 692 |
| | Achieved | 101 | 62 | 17174 | 573 |
| | Degradation | +32.9% | -39.2% | +113.6% | -17.2% |
| Run 1 ($\epsilon_{\max} = 180, h_s = 2$) | Planned | 0 | 0 | 5829 | 136 |
| | Achieved | 13 | 0 | 13304 | 230 |
| | Degradation | NA | - | +128.2% | +69.1% |
| Run 2 ($\epsilon_{\max} = 180, h_s = 4$) | Planned | 0 | 0 | 3181 | 36 |
| | Achieved | 12 | 0 | 11346 | 183 |
| | Degradation | NA | - | +256.7% | +408.3% |
| Run 3 ($\epsilon_{\max} = 90, h_s = 2$) | Planned | 0 | 0 | 2322 | 117 |
| | Achieved | 11 | 0 | 10822 | 239 |
| | Degradation | NA | - | +366.1% | +104.3% |
| Run 4 ($\epsilon_{\max} = 90, h_s = 4$) | Planned | 0 | 0 | 1547 | 30 |
| | Achieved | 12 | 0 | 9443 | 159 |
| | Degradation | NA | - | +510.4% | +430% |

Table 5.7: DDSCS degradation analysis

robustness among all.

By analysing these results, one can note that similarly to the Deterministic Static SCS results, our results are better than the ones of the OSU in every way, but suffer from degradation. The same rule still holds true, where the better the planned schedule is, the more degradation its corresponding achieved one will have. This makes it clear that a more robust solution is a necessity in order to overcome this degradation.

5.5 Conclusion

In this chapter, we studied the results for both the Deterministic Static SCS (from Chapter 3) and Deterministic Dynamic SCS (from Chapter 4). For each problem, we presented the achieved schedules and provided an extensive analysis of the results and compared it with the ones of the OSU. Our results in both problems outperformed the ones of the OSU in every criteria. Despite this lead over the OSU, we noticed a certain amount of degradation in solution qualities when comparing each of our planned schedules to its corresponding achieved schedule.

This degradation was clear in both problems and represented a loss in solutions qualities. In the Deterministic Static SCS, we found the effects to be more than 400% increase in the amount of overtime and problem kits when passing from planned to achieved schedules. In the case of Deterministic Dynamic SCS, we found similar behaviour with increases of more than 250% in most cases in overtime and problem kits. Further analyses revealed that these big degradations are due to the big differences between surgeries estimated and real durations, and that in both static and dynamic context, the better the planned solution is, the bigger the degradation will be. This probably can be explained by the fact that our solutions are more tightly packed and consequently are more sensible to uncertainties.

In conclusion, the effects of the uncertainties found in surgeries durations is too huge to ignore. For this, we will tackle in the next part the non deterministic version of the problem, where we will try to generate more robust schedules, less sensible to these uncertainties.

PART III

The Non-Deterministic Surgical case scheduling problem

NON-DETERMINISTIC STATIC SURGICAL CASE SCHEDULING PROBLEM

6.1 Introduction

In the previous part, we tackled the deterministic version of the OSU's scheduling problem. During our experiments, we found that the results obtained with our method gave a huge competitive edge over the ones of the OSU when considering the planned schedules. This cannot be said for the achieved schedules as our results were still better than the ones of the OSU, but suffered from a big degradation in solution qualities. To overcome this, we will tackle the non-deterministic version of the surgical case scheduling problem and try to build more robust schedules against surgeries durations uncertainties.

In this chapter, we will tackle the Non-Deterministic Static Surgical Case Scheduling problem (NDSSCS) of the CHU. First, we will provide some technical background on the solutions methods used to deal with data uncertainties. Next, we propose two robust models and compare the numerical results of both models with the deterministic model and the original data. Finally, we analyse the differences between the two models and conclude with this analysis.

6.2 Technical review

This section provides some technical background on the solutions methods used to deal with data uncertainties, namely stochastic programming and robust optimization. We start by introducing both methods and comparing the cons and pros for each one. Then, we focus on the robust optimization approaches and justify our choice of RO approaches for our problem.

6.2.1 Non-Deterministic Optimisation

In real life problems, it is very common that the data includes some sort of uncertainty in the parameters values at the time a decision should be made. Of course it is possible to solve the problem using estimated or nominal parameter values, but as seen previously in Part II, this can lead to poor quality solutions and infeasibility in some cases. To deal with such uncertainties, two main methodologies are used in the literature: Stochastic Programming (SP) and Robust Optimisation (RO).

The difference between SP and RO, discussed in [[Bertsimas et al., 2011](#)], is that, contrary to SP methods, RO approaches consider that the uncertainty model is not stochastic, but rather deterministic and set-based. Instead of seeking to immunize the solution in some probabilistic sense to stochastic uncertainty, these methods construct a solution that is feasible for any realization of the uncertainty in a given set.

On one hand, SP approaches rely completely on the availability of historical data to derive probability distributions and generate scenarios if any (as in Monte Carlo sampling and Sample Average Approximation (SAA) methods). One of the main drawbacks of SP approaches is the difficulty of determining the number of scenarios to compute, as a higher number of generated scenarios leads to obtaining a better estimations quality of the uncertainties, but at the cost of higher computational times. Moreover, it may prove difficult to fit the probability distributions to the uncertain data. Although using SP approaches usually adds to the complexity of the problem, it provides more flexibility when compared to RO, as it is possible to compute or estimate the expectation function or to control the risk.

On the other hand, RO approaches require very little information about the uncertain parameters, as it needs only a convex or discrete description of the uncertainty. In addition, these methods are more understandable to decision makers and are often easier to implement than the SP approaches. Despite the lower flexibility that RO approaches provide in comparison with the SP counterpart, they prove to add less complexity to the problem and are computationally tractable if the uncertainty sets satisfies mild convexity and computability assumptions [[Ben-Tal et al., 2009](#)]. This made the RO an attractive alternative to the SP, which can be clearly seen in the increasing attention it is gaining lately [[Cardoen et al., 2010](#)].

The choice of which method to use in a particular problem is not always a clear one and depends heavily on the availability of historical data. In the case of the SCS problem of the CHU, the data provided to us lacked this historical depth, which made the choice of using RO a logical one. This can be further supported by analysing the number of surgeries in each type from the data provided to us by the OSU (see Figure 1.15), where 75 surgery types had less than 5 surgeries which is not enough to derive any probability distributions for the SP methods to function properly. In the remainder of this section, we present a more in depth technical review of the RO, then we explore the two main approaches used in the literature for RO.

6.2.2 Robust Optimisation

Robust optimization is a field of optimization theory that deals with optimization problems in which a certain measure of robustness is sought against uncertainty in the values of the parameters of the problem. Generally speaking, robust optimization is often known as worst-case optimization as the goal is to generate a solution that is feasible for any possible realisation of the uncertain parameters.

In the literature, there are two main approaches used to achieve such robustness in the solutions. The first approach is through applying a classical robust formulation, while the second approach is called Redundancy-based techniques.

Classical robust formulations

Three main formulations were presented in the literature. To compare the three approaches, we will consider as in [Bertsimas and Sim, 2004] the following nominal linear optimization problem:

$$\begin{aligned} & \text{maximise} && \sum_{j=1}^m c_j x_j \\ & \text{st.} && \sum_{j=1}^m a_{ij} x_j \leq b_i, \quad \forall i = 1 \dots n \\ & && x \in X \end{aligned}$$

where $c = (c_j) \in \mathbb{R}^n$, $b = (b_i) \in \mathbb{R}^m$, $A = (a_{ij}) \in \mathbb{R}^{n \cdot m}$, and $X \subseteq \mathbb{R}_+^m$.

In this formulation, the uncertainty affects only the parameter A and not the objective function. For each row i of the matrix A , let J_i represent a set of coefficients in row i that are subject to uncertainty. Each entry a_{ij} , $j \in J_i$ is modelled as a bounded random variable \tilde{a}_{ij} , $j \in J_i$ that takes values in $[\bar{a}_{ij} - \hat{a}_{ij}, \bar{a}_{ij} + \hat{a}_{ij}]$, where \bar{a}_{ij} (referred to as the nominal value of coefficient a_{ij}) and $\hat{a}_{ij} \geq 0$ are given in advance. Therefore, it exists a random variable ζ_{ij} that takes values in $[-1, 1]$, such that:

$$\tilde{a}_{ij} = \bar{a}_{ij} - \zeta_{ij} \hat{a}_{ij}$$

Then, the robust problem is equivalent to:

$$\begin{aligned} & \text{maximise} && \sum_{j=1}^m c_j x_j \\ & \text{st.} && \sum_{j=1}^m \bar{a}_{ij} x_j + \sum_{j=1}^m \zeta_{ij} \hat{a}_{ij} x_j \leq b_i, \quad \forall i = 1 \dots n \\ & && x \in X \end{aligned}$$

- **The robust formulation of Soyster**

The main idea of the formulation presented in [Soyster, 1973] is that the solution must be feasible for every possible realization \tilde{a}_{ij} of the uncertain data. Thus, this formulation yields the highest protection "Robustness".

$$\begin{aligned} & \text{maximise} && \sum_{j=1}^m c_j x_j \\ & \text{st.} && \sum_{j=1}^m (\bar{a}_{ij} + \hat{a}_{ij}) x_j \leq b_i \quad \forall i = 1 \dots n \\ & && x \in X \end{aligned}$$

Despite the high protection that this formulation presents, it is considered too conservative and hence the robust solution that it yields has an objective function value much worse than the objective value of the solution of the nominal problem.

To overcome this problem, two robust formulations that are quite similar in the concept, have been designed. Moreover, under the additional assumption of a symmetrical distribution for ζ_{ij} , these approaches offer a guarantee of performance.

- **The robust formulation of Ben-Tal and Nemirovski**

The authors in [[Ben-Tal and Nemirovski, 2000](#)] proposed the following formulation:

$$\begin{aligned}
 & \text{maximise} && \sum_{j=1}^m c_j x_j \\
 & \text{st.} && \sum_{j=1}^m \bar{a}_{ij} x_j + \sum_{j=1}^m \hat{a}_{ij} y_j + \Omega_i \sqrt{\sum_{j=1}^m \hat{a}_{ij}^2 z_j^2} \leq b_i \quad \forall i = 1 \dots n \\
 & && -y_j \leq x_j - z_j \leq y_j \quad \forall j = 1 \dots m \\
 & && y_j \geq 0 \quad \forall j = 1 \dots m \\
 & && x \in X
 \end{aligned}$$

In this formulation, the authors considered a spherical uncertainty set where for each row i , the L^2 -norm of the deviations is bounded by a predefined parameter $\Omega_i \geq 0$. The main drawback of this approach is that it is a quadratic (non-linear) model which makes it unattractive for solving robust discrete optimization models.

- **The robust formulation of Bertsimas and Sim**

The main idea behind the formulation presented in [[Bertsimas and Sim, 2004](#)] is to introduce a parameter Γ_i that controls the level of "robustness" against the level of conservatism of the solution. The parameter Γ_i takes values in the interval $[0, |J_i|]$. The authors suppose that it is unlikely that all of the $a_{ij}, j \in J_i$ will change and thus the goal is to protect against all cases that up to $\lfloor \Gamma_i \rfloor$ of these coefficients are allowed to change, and one coefficient a_{ij} changes by $(\Gamma_i - \lfloor \Gamma_i \rfloor) \hat{a}_{ij}$.

They start by considering the following (nonlinear) formulation:

$$\begin{aligned}
 & \text{maximise} && \sum_{j=1}^m c_j x_j \\
 & \text{st.} && \sum_{j=1}^m \bar{a}_{ij} x_j + \max_{\{S_i \cup \{t_i\} | S_i \subseteq J_i, |S_i| = \lfloor \Gamma_i \rfloor, t_i \in J_i \setminus S_i\}} \left\{ \sum_{j \in S_i} \hat{a}_{ij} y_j + (\Gamma_i - \lfloor \Gamma_i \rfloor) \hat{a}_{it_i} y_{t_i} \right\} \leq b_i \quad \forall i = 1 \dots n \\
 & && -y_j \leq x_j \leq y_j \quad \forall j = 1 \dots m \\
 & && y_j \geq 0 \\
 & && x \in X
 \end{aligned}$$

Then they prove that this nonlinear model has an equivalent linear formulation as follows:

$$\begin{aligned}
& \text{maximise} && \sum_{j=1}^m c_j x_j \\
& \text{st.} && \sum_{j=1}^m a_{ij} x_j + z_i \Gamma_i + \sum_{j=1}^m p_{ij} \leq b_i \quad \forall i = 1 \dots n \\
& && z_i + p_{ij} \geq \hat{a}_{ij} y_j \quad \forall i = 1 \dots n, j = 1 \dots m \\
& && -y_j \leq x_j \leq y_j \quad \forall j = 1 \dots m \\
& && l_j \leq x_j \leq u_j \quad \forall j = 1 \dots m \\
& && p_{ij} \geq 0 \quad \forall i = 1 \dots n, j = 1 \dots m \\
& && y_j \geq 0 \quad \forall j = 1 \dots m \\
& && z_i \geq 0 \quad \forall i = 1 \dots n \\
& && x \in X
\end{aligned}$$

One key point of this formulation is that it maintains the linearity of the problem, which makes solving the robust counterpart as tractable as the nominal problem. Moreover, it is less conservative than the formulation of [Soyster, 1973], as the level of conservatism can be controlled via the parameter Γ . For this, we will test this formulation in our problem.

Redundancy-based techniques

The main goal of the Redundancy-based techniques (also called fault tolerance techniques) in scheduling problems is to ensure that faults in the system does not cause overall system failure [Herroelen and Leus, 2005]. This can be achieved through two main ways:

1. **Resource redundancy:** achieved by keeping in standby multiple identical sets of resources [Ghosh and Mosse, 1996].
2. **Time redundancy:** scheduling backup tasks that reserve time for re-execution in the event of a fault [Ghosh et al., 1995].

One main problem with the resource redundancy approach is that it is unrealistic to implement it in a project environment, as doubling the various resources would be cost prohibitive. On the other hand, time redundancy approaches do not suffer from the same limitation and have been explored in the literature.

In the case of the non-deterministic SCS problem, the authors in [Hans et al., 2008] proposed a slack-time based approach, where a sufficient planned time slacks are added to their schedule to provide some sort of robustness. In their work, the authors assume that surgery durations are mutually independent and base the amount of planned slack on each OR-day to prevent overtime on the expected variance of the durations of the surgeries planned on that OR-day. Given that:

- K_{st} is an OR-day and represents an OR $k \in K$ that is available at day $t \in T$ for speciality $s \in S$.
- N_{skt} indicates the set of surgeries assigned by speciality s in operating room k , on day t .
- σ_i is the standard deviation of surgery $i \in N$.

- β ($\beta \geq 0$) is a parameter that influences the probability that the surgeries will complete on time.
- O_{skt} is the overtime on OR-day (s, k, t) .
- c_{kt} is the capacity of OR k at day t .

The expected duration of the planned surgeries on OR-day (s, k, t) , $k \in K_{st}$ is:

$$\mu_{skt} = \sum_{i \in N_{skt}} \mu_i$$

Next, the variance of the planned surgeries on OR-day (s, k, t) , $k \in K_{st}$ is:

$$\sigma_{skt}^2 = \sum_{i \in N_{skt}} \sigma_i^2$$

Then, the planned slack size δ_{skt} on each OR-day (s, k, t) , $k \in K_{st}$, is calculated as follows:

$$\delta_{skt} = \beta \cdot \sqrt{\sum_{i \in N_{skt}} \sigma_i^2}$$

Finally, the OR-day capacity constraint at each OR k for each speciality s is as follows:

$$\sum_{i \in N_{skt}} \mu_i + \delta_{skt} \leq c_{kt} + O_{skt}$$

6.3 First formulation

In our first approach, we will apply a robust optimization formulation following the one of Bertsimas and Sim ([Bertsimas and Sim, 2004]).

The random variables that we will consider in our case represent the stochastic duration \tilde{p}_i of each surgery i . The value of \tilde{p}_i may vary in a given interval $[\bar{p}_i - \hat{p}_i, \bar{p}_i + \hat{p}_i]$. Note that given the nature of our problem, we will assume that the distribution function $F(\tilde{p}_i)$ for each surgery type is symmetrical where, \hat{p}_i represents the standard deviation parameter of $F(\tilde{p}_i)$ and \bar{p}_i represents the central value. These values were calculated using 10 months' historical data for each surgery type and applied to all the surgeries in this type. Our goal is to protect against the maximum value of this random variable expressed as $\bar{p}_i + \hat{p}_i$.

In our case, the formulation of Bertsimas and Sim guarantees that any solution is feasible for all the constraints if at most Γ surgeries reach their maximum duration in each room r and each day t over a full length day (8h45), assuming that the other surgeries take their central value. These surgeries are chosen so that they have the worst impact on the total duration of surgeries assigned to room r on day t , and our solution is then forced to be feasible for this realisation.

Due to the difference in periods lengths, we use a ratio Γ^{bf} of this Γ for each periods pair (b, f) , $b \in \{1, \dots, J\}$, $f \in \{b, \dots, J\}$ such that:

$$\Gamma^{bf} = (d_{bf}/d_{1J}).\Gamma \quad (6.1)$$

where:

$$d_{bf} = \max(d_{rt}^{bf}), \quad \forall r \in \{1, \dots, R\}, \forall t \in \{1, \dots, T\} \quad (6.2)$$

In our deterministic model presented in Section 3.4, the two constraints concerned with the uncertainties are constraints (3.4) and (3.14) as follows:

- Constraints (3.4) control the workload of surgeries for each interval of the day and each room.

$$\sum_{i=1}^O \sum_{b=\beta}^{\gamma} \sum_{f=b}^{\gamma} p_i \cdot x_{itr}^{bf} + \sum_{i=1}^O \sum_{b=1}^{\beta-1} \sum_{f=\gamma+1}^J d_{rt}^{\beta\gamma} \cdot x_{itr}^{bf} \leq d_{rt}^{\beta\gamma} \cdot L_{tr} + u_{\gamma} \cdot \varepsilon_{tr}, \quad (3.4)$$

$$\forall \beta \in \{1, \dots, J\}, \forall \gamma \in \{\beta, \dots, J\}, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\}$$

- Constraints (3.14) ensure that all ambulatory surgeries finish before A_{max} .

$$\sum_{i=1}^O \sum_{b=\beta}^2 \sum_{f=b}^2 p_i \cdot x_{itr}^{bf} + \sum_{i=1}^O \sum_{b=\beta}^3 a_i \cdot p_i \cdot x_{itr}^{b3} \leq A_{max} - \sum_{j=1}^{\beta-1} d_{rt}^j, \quad (3.14)$$

$$\forall \beta \in \{1, \dots, 3\}, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\}$$

As explained before, we will force the feasibility in any solution given that at most Γ surgeries take their maximum duration each full day in each room by replacing these constraints respectively by:

$$\sum_{i=1}^O \sum_{b=\beta}^{\gamma} \sum_{f=b}^{\gamma} \bar{p}_i \cdot x_{itr}^{bf} + \hat{p}_{tr}^{\beta\gamma} + \sum_{i=1}^O \sum_{b=1}^{\beta-1} \sum_{f=\gamma+1}^J d_{rt}^{\beta\gamma} \cdot x_{itr}^{bf} \leq d_{rt}^{\beta\gamma} \cdot L_{tr} + u_{\gamma} \cdot \varepsilon_{tr}, \quad (6.3)$$

$$\forall \beta \in \{1, \dots, J\}, \forall \gamma \in \{\beta, \dots, J\}, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\}$$

and:

$$\sum_{i=1}^O \sum_{b=\beta}^2 \sum_{f=b}^2 \bar{p}_i \cdot x_{itr}^{bf} + \hat{p}_{tr}^{\beta 2} + \sum_{i=1}^O \sum_{b=\beta}^3 a_i \cdot \bar{p}_i \cdot x_{itr}^{b3} + \hat{p}_{tr}^{\beta 3} \leq A_{max} - \sum_{j=1}^{\beta-1} d_{rt}^j, \quad (6.4)$$

$$\forall \beta \in \{1, \dots, 3\}, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\}$$

where for each day t , each room r , each starting period β and finishing period γ , $\hat{p}_{tr}^{\beta\gamma}$ is the optimal value of the following linear program ($\Omega_{tr}^{\beta\gamma}$):

$$(\Omega_{tr}^{\beta\gamma}) = \text{maximise} \sum_{i=1}^O \left(\sum_{b=\beta}^{\gamma} \sum_{f=b}^{\gamma} \hat{p}_i \cdot x_{itr}^{bf} \right) \cdot z_i \quad (6.5)$$

$$\sum_{i=1}^O z_i \leq \Gamma^{\beta\gamma} \quad (6.6)$$

$$0 \leq z_i \leq 1, \quad \forall i \in \{1, \dots, O\} \quad (6.7)$$

We then consider the dual of problem ($\Omega_{tr}^{\beta\gamma}$):

$$\text{Minimize} \quad \sum_{i=1}^O \eta_{itr}^{\beta\gamma} + \Gamma^{\beta\gamma} z_{tr}^{\beta\gamma} \quad (6.8)$$

$$\text{Subject to:} \quad \eta_{itr}^{\beta\gamma} + z_{tr}^{\beta\gamma} \geq \sum_{b=\beta}^{\gamma} \sum_{f=b}^{\gamma} \hat{p}_i \cdot x_{itr}^{bf}, \quad \forall i \in \{1, \dots, O\} \quad (6.9)$$

$$\eta_{itr}^{\beta\gamma} \geq 0, \quad \forall i \in \{1, \dots, O\} \quad (6.10)$$

$$z_{tr}^{\beta\gamma} \geq 0 \quad (6.11)$$

Since problem $(\Omega_{tr}^{\beta\gamma})$ is feasible and bounded for all $\Gamma^{\beta\gamma} \in [0, |O|]$, then by strong duality, the dual problem in (6.8) is also feasible and bounded, and their objective values coincide. From this, the complete robust model formulation is obtained from the deterministic model (in Section 3.4) by replacing constraint (3.4) by:

$$\sum_{i=1}^O \sum_{b=\beta}^{\gamma} \sum_{f=b}^{\gamma} \bar{p}_i \cdot x_{itr}^{bf} + \sum_{i=1}^O \eta_{itr}^{\beta\gamma} + \Gamma^{\beta\gamma} z_{tr}^{\beta\gamma} + \sum_{i=1}^O \sum_{b=1}^{\beta-1} \sum_{f=\gamma+1}^J d_{rt}^{\beta\gamma} \cdot x_{itr}^{bf} \leq d_{rt}^{\beta\gamma} \cdot L_{tr} + u_{\gamma} \cdot \varepsilon_{tr}, \quad (6.12)$$

$$\forall \beta \in \{1, \dots, J\}, \forall \gamma \in \{\beta, \dots, J\}, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\}$$

constraint (3.14) by:

$$\sum_{i=1}^O \sum_{b=2}^2 \sum_{f=b}^2 \bar{p}_i \cdot x_{itr}^{bf} + \sum_{i=1}^O \eta_{itr}^{\beta 2} + \Gamma^{\beta 2} z_{tr}^{\beta 2} + \sum_{i=1}^O \sum_{b=\beta}^3 a_i \cdot \bar{p}_i \cdot x_{itr}^{b3} + \sum_{i=1}^O a_i \cdot \eta_{itr}^{\beta 3} + \Gamma^{\beta 3} z_{tr}^{\beta 3}$$

$$\leq A_{max} - \sum_{j=1}^{\beta-1} d_{rt}^j, \quad \forall \beta \in \{1, \dots, 3\}, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\} \quad (6.13)$$

and by adding the following three constraints:

$$\eta_{itr}^{\beta\gamma} + z_{tr}^{\beta\gamma} \geq \sum_{b=\beta}^{\gamma} \sum_{f=b}^{\gamma} \hat{p}_i \cdot x_{itr}^{bf}, \quad (6.14)$$

$$\forall i \in \{1, \dots, O\}, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\}, \forall \beta \in \{1, \dots, J\}, \forall \gamma \in \{\beta, \dots, J\}$$

$$\eta_{itr}^{\beta\gamma} \geq 0, \quad (6.15)$$

$$\forall i \in \{1, \dots, O\}, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\}, \forall \beta \in \{1, \dots, J\}, \forall \gamma \in \{\beta, \dots, J\}$$

$$z_{tr}^{\beta\gamma} \geq 0, \quad (6.16)$$

$$\forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\}, \forall \beta \in \{1, \dots, J\}, \forall \gamma \in \{\beta, \dots, J\}$$

where:

- Γ is the robust measure, representing the number of surgeries that can reach their maximum duration in each room r and each day t over a full length day (8h45).
- \bar{p}_i is the duration of surgery i representing the central value of the normal distribution function $F(\tilde{p}_i)$ applied for all the surgeries from the same type as surgery i .
- \hat{p}_i is the maximum duration that surgery i can be increased by, representing the standard deviation parameter of the normal distribution function $F(\tilde{p}_i)$ applied for all the surgeries from the same type as surgery i .

6.4 Second formulation

In our second robust approach, we will implement a redundancy-based technique based on the one presented in [Hans et al., 2008]. The main idea behind this approach is to introduce sufficient planned time slack into the schedule in order to minimize the risk of overtime and urgent and priority kits.

Unlike our first formulation which protects against Γ number of surgeries that will have the worst impact on the total duration of surgeries in room r at day t , this approach inserts planned slack for each scheduled surgery in each room r on each day t . The size of the slack is calculated based on the standard deviation of the surgery's duration.

Let z_{trbf} represents the amount of slack given for day t at room r for surgeries that start at period b and end at period f . The value of z_{trbf} is calculated as follows:

$$z_{trbf} \geq B \cdot \sum_{i=1}^O (\hat{p}_i \cdot x_{itr}^{bf}) \quad (6.17)$$

$$\forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\}, \forall b \in \{1, \dots, J\}, \forall f \in \{b, \dots, J\}$$

where:

- $B(\geq 0)$ is a parameter to control the probability that a surgery will finish on time.
- \hat{p}_i is the standard deviation of surgery i .

Similar to our first robust approach, the two constraints concerned with the uncertainties in our deterministic model (from Section 3.4) are constraints (3.4) and (3.14) as follows:

- Constraints (3.4) control the workload of surgeries for each interval of the day and each room.

$$\sum_{i=1}^O \sum_{b=\beta}^{\gamma} \sum_{f=b}^{\gamma} p_i \cdot x_{itr}^{bf} + \sum_{i=1}^O \sum_{b=1}^{\beta-1} \sum_{f=\gamma+1}^J d_{rt}^{\beta\gamma} \cdot x_{itr}^{bf} \leq d_{rt}^{\beta\gamma} \cdot L_{tr} + u_{\gamma} \cdot \varepsilon_{tr}, \quad (3.4)$$

$$\forall \beta \in \{1, \dots, J\}, \forall \gamma \in \{\beta, \dots, J\}, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\}$$

- Constraints (3.14) ensure that all ambulatory surgeries finish before A_{max} .

$$\sum_{i=1}^O \sum_{b=\beta}^2 \sum_{f=b}^2 p_i \cdot x_{itr}^{bf} + \sum_{i=1}^O \sum_{b=\beta}^3 a_i \cdot p_i \cdot x_{itr}^{b3} \leq A_{max} - \sum_{j=1}^{\beta-1} d_{rt}^j, \quad (3.14)$$

$$\forall \beta \in \{1, \dots, 3\}, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\}$$

Thus, we will add the required slack represented by the value of z_{trbf} to each of the concerned constraints. Finally, we obtain the full robust model from the deterministic MILP (presented in Section 3.4) by adding constraints (6.17) and replacing constraints (3.4) by:

$$\sum_{i=1}^O \sum_{b=\beta}^{\gamma} \sum_{f=b}^{\gamma} p_i \cdot x_{itr}^{bf} + \sum_{b=\beta}^{\gamma} \sum_{f=b}^{\gamma} z_{trbf} + \sum_{i=1}^O \sum_{b=1}^{\beta-1} \sum_{f=\gamma+1}^J d_{rt}^{\beta\gamma} \cdot x_{itr}^{bf} \leq d_{rt}^{\beta\gamma} \cdot L_{tr} + u_{\gamma} \cdot \varepsilon_{tr} \quad (6.18)$$

$$\forall \beta \in \{1, \dots, J\}, \gamma \in \{\beta, \dots, J\}, t \in \{1, \dots, T\}, r \in \{1, \dots, R\}$$

and constraints (3.14) by:

$$\sum_{i=1}^O \sum_{b=\beta}^2 \sum_{f=b}^2 p_i \cdot x_{itr}^{bf} + \sum_{b=\beta}^2 \sum_{f=b}^2 z_{trbf} + \sum_{i=1}^O \sum_{b=\beta}^3 a_i \cdot p_i \cdot x_{itr}^{b3} + \sum_{b=\beta}^3 z_{trb3} \leq A_{max} - \sum_{j=1}^{\beta-1} d_{rt}^j \quad (6.19)$$

$$\forall \beta \in \{1, \dots, 3\}, t \in \{1, \dots, T\}, r \in \{1, \dots, R\}$$

6.5 Experimental results

In order to test the two proposed NDSSCS models, we ran the experiments on the 10 instances described in Table 1.3. For the first model, we studied the data from the OSU to determine the value of Γ and found that on average approximately 1 surgery takes the worst duration scenario ($\bar{p}_i + \hat{p}_i$) in each room during a full day. We therefore ran our experiments using the value $\Gamma = 1$. For the second model, we tried different values for B ($B \in \{0.25, 0.5, 0.75, 1\}$) and found that the best value that provides robustness to the results without being too conservative is 0.5. We therefore ran our experiments using the value $B = 0.5$.

The time limits for these experiments were set to 3 hours per objective (9 hours per instance) for the first model and 1 hour per objective (3 hours per instance) for the second model. These different time limits were chosen based on the results of our experiments, where we found that the first model needs more time (around 3 hours per objective) to find good results and the second model needs around 1 hour. An example is given in Figure 6.1, where we compare the distance between the reported best solution obtained and the lower bound for each model while solving the first objective for the second instance. By analysing the example, it is clear that the second model obtains good solutions (close to the LB) way faster than the first model. The same behaviour continued over all the instances for all the objectives, and hence the different set time limits.

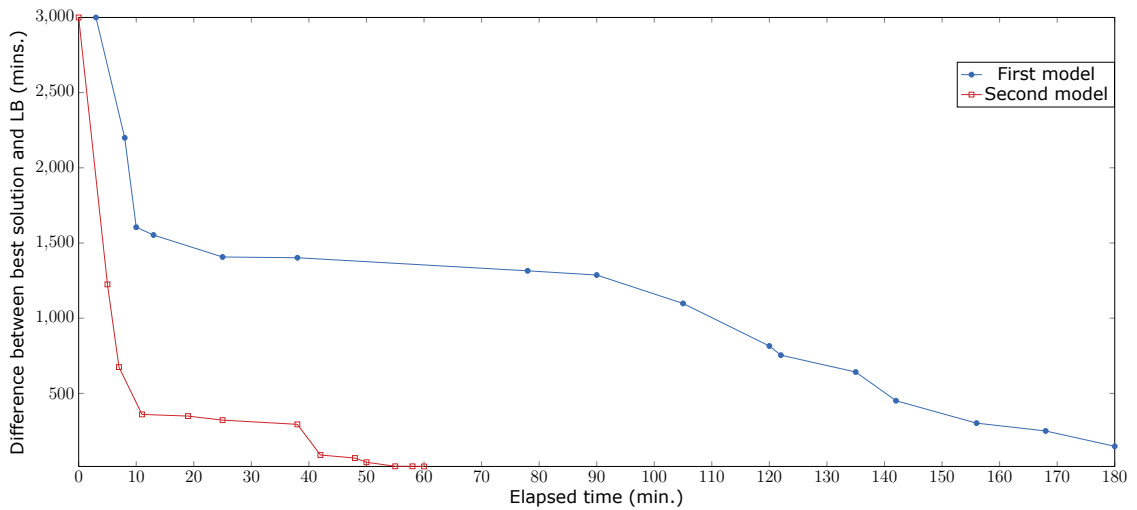


Figure 6.1: Comparison between time needed for both models to solve an objective (first objective for second instance)

6.5.1 Comparison of planned schedules

Following the work process of the OSU, we start our comparison with the planned schedules. Tables 6.1, 6.2 and 6.3 compare the objective functions obtained with the two robust models with our deterministic results (Section 3.6.2) and the planned schedule of the OSU (Section 1.8.1).

Table 6.1 compares the total overtime between the three schedules. In the case of both robust planned schedules, the 'MILP value' column represents the overtime reported by the solver which includes the added time that the MILP uses to ensure the robustness of the solution and the 'overtime' column represents the actual overtime without the added time by the solver. The average overtime in the first robust planned schedules is 286.3 minutes (4 hours and 46 minutes) per month. The average difference between the first robust schedules and those of the OSU is 8 hours 37 minutes per month, representing a decrease of approximately 64.4%, which is less than the deterministic model (9 hours and 48 minutes). On the other hand, the average overtime obtained using the second robust model is 294.2 minutes (4 hours and 54 minutes) per month, representing an increase of 2.6% from the first robust model (+8 minutes).

In addition, the 'max' column shows the maximum overtime in minutes found in a day during the month considered. In 8 instances, the first robust solutions provided lower values than the OSU's solution, while the second robust solutions provided lower values than the OSU in 7 instances, and the deterministic solutions provided lower values than the OSU's solution in 6 instances. The 'LB' column shows the lower bound found by the MILP. We see that the first robust approach did not find any optimal solutions. This is due to the added complexity of the robust MILP which also affects the lower bound calculations as the model needs way more time to report more accurate bounds. On the other hand, the second robust MILP provided a close performance to the deterministic model with 4 optimal solutions (compared to 6 optimal solutions obtained with the deterministic MILP). Finally, the 'CPU' column represents the execution time for the models. The first robust model takes more time than both the deterministic and second robust models to find good solutions. In conclusion for the first objective, the first robust model provides the best results but uses more time than the second model.

Moving to the second objective, table 6.2 shows the number of ORs opened with the deterministic and both robust methods and in the original data, and also compares the room occupancy rates in the four schedules. The conservative nature of both robust models can be seen in the lower number of rooms closed on average each month (1.6 and 1.1 rooms respectively) compared to the deterministic model (1.9 closed rooms on average per month). The 'LB' column shows the lower bound found by the solver concerning the rooms opened. In all instances, the first robust model could not prove optimality for any of the instances, while the second model proved optimality for 3 instances. Again we see that the execution times for the first model are way more than the ones in the second model.

When comparing the occupancy rates, we observe an average occupancy rate of approximately 82.9% in the first robust solutions, 82.1% in the second robust solutions, 84.4% in the deterministic solutions, and 80.9% for the OSU. In addition, the 'Min Occ. rate' column represents the minimum occupancy rate found in each schedule. In the first robust schedules, 7 instances had a minimum occupancy rate greater than 50%, compared with 5 instances in the second robust schedules, 6 instances in the deterministic schedules, and only 1 instance in the OSU's schedules. Moreover, only 1 instance in both the first and second robust schedules

had a minimum occupancy rate below 20%, compared to 4 instances for the OSU and 0 in the deterministic schedules. These values show that both robust solutions have better room load distribution than those of the OSU, but lack behind the deterministic solutions. Therefore, both models provide similar results for the second objective, but the first model takes more time than the second one.

Finally, Table 6.3 compares the total number of obtained urgent and priority kits. In the case of the first robust solution, the average number of urgent kits is 1.1 per month and the average number of priority kits is 5.8 (decrease of 90.03% from the OSU's solution in the total problem kits number), compared to an average of 0.6 urgent kits and 4.5 priority kits per month obtained using the second robust model (decrease of 92.6% over the OSU), and a decrease of 94.65% obtained with the deterministic model. The 'CPU' column represents the execution times for the three models concerning only the urgent and priority kits objective. Following the same behaviour as in the previous objectives, the first robust model takes more time than the deterministic model to find good solutions, while the second robust model performed somewhat similar to the deterministic model and was able to find the optimal solution in 2 instances (compared to 0 optimal solutions using the first robust method and 3 using the deterministic model).

This comparison between the planned schedules shows globally that the results obtained using the first robust model are better in terms of the overtime and number of opened ORs and competitive in terms of the problem kits numbers, but use at least 3 times more execution time than the second robust model.

| Instance | planned schedule | | | | | | | | | | | | | | | |
|----------------|----------------------|-----|--------------------|----------------------|-----|---------|------------------------------------|---------------|----------------------|-----|---------|----------------------------------|------------------|--------------|-----|---------|
| | OSU | | Deterministic MILP | | | | First robust MILP ($\Gamma = 1$) | | | | | Second robust MILP ($B = 0.5$) | | | | |
| | overtime (in min) | max | LB | overtime (in min) | max | CPU (s) | LB | MILP value | overtime (in min) | max | CPU (s) | LB value | MILP (in min) | overtime | max | CPU (s) |
| 1 | 590 | 131 | 0 | 0* | 0 | 16 | 0 | 153 | 87 | 17 | 10800 | 211 | 211* | 141 | 64 | 321 |
| 2 | 714 | 176 | 368 | 384 | 90 | 3600 | 369 | 517 | 313 | 131 | 10800 | 575 | 589 | 416 | 131 | 3600 |
| 3 | 502 | 81 | 89 | 89* | 23 | 1089 | 0 | 412 | 297 | 16 | 10800 | 291 | 295 | 154 | 56 | 3600 |
| 4 | 656 | 77 | 314 | 318 | 94 | 3600 | 0 | 913 | 655 | 151 | 10800 | 490 | 502 | 402 | 101 | 3600 |
| 5 | 1159 | 120 | 312 | 312* | 123 | 58 | 98.9 | 1084 | 701 | 109 | 10800 | 522 | 522* | 375 | 122 | 217 |
| 6 | 1278 | 183 | 347 | 350 | 84 | 3600 | 0 | 491 | 232 | 53 | 10800 | 605 | 613 | 409 | 113 | 3600 |
| 7 | 641 | 109 | 225 | 232 | 130 | 3600 | 0 | 430 | 199 | 48 | 10800 | 554 | 562 | 398 | 117 | 3600 |
| 8 | 561 | 65 | 334 | 334* | 106 | 1619 | 57.3 | 455 | 317 | 67 | 10800 | 499 | 501 | 458 | 45 | 3600 |
| 9 | 870 | 105 | 136 | 136* | 37 | 1031 | 0 | 157 | 62 | 19 | 10800 | 347 | 347* | 189 | 51 | 2196 |
| 10 | 1070 | 266 | 0 | 0* | 0 | 781 | 0 | 19 | 0 | 0 | 10800 | 0 | 0* | 0 | 0 | 934 |
| Average | 804.1 | | | 215.5 | | | | | 286.3 | | | | | 294.2 | | |

Table 6.1: Overtime comparison (planned schedules)

| Instance | planned schedule | | | | | | | | | | | | | | | | | |
|----------------|-------------------|---------------------|----------------------|--------------------|-------------------|---------|---------------------|------------------------------------|------|-------------------|---------|----------------------------------|----------------------|------|-------------------|---------|---------------------|----------------------|
| | OSU | | | Deterministic MILP | | | | First robust MILP ($\Gamma = 1$) | | | | Second robust MILP ($B = 0.5$) | | | | | | |
| | # opened rooms | Min Occ. rate | Avg. Occ. rate | LB | # opened rooms | CPU (s) | Min Occ. rate | Avg. Occ. rate | LB | # opened rooms | CPU (s) | Min Occ. rate | Avg. Occ. rate | LB | # opened rooms | CPU (s) | Min Occ. rate | Avg. Occ. rate |
| 1 | 59 | 27.6% | 80.2% | 55.1 | 58 | 3600 | 48.6% | 81.7% | 48.8 | 58 | 10800 | 64.6% | 81.3% | 56.2 | 58 | 3600 | 54.2% | 80.7% |
| 2 | 59 | 13.8% | 77.5% | 52.3 | 55 | 3600 | 51.6% | 82.4% | 46.3 | 56 | 10800 | 19% | 79.2% | 53.4 | 56 | 3600 | 23.4% | 80.1% |
| 3 | 48 | 18.3% | 80.7% | 44.6 | 46 | 3600 | 55.2% | 85.4% | 39.6 | 46 | 10800 | 43.7% | 83.7% | 44.5 | 46 | 3600 | 49% | 84% |
| 4 | 48 | 42.4% | 86.5% | 46.8 | 48 | 3600 | 40.7% | 87.4% | 42.3 | 48 | 10800 | 75.6% | 86.7% | 45.9 | 48 | 3600 | 42.4% | 85.8% |
| 5 | 59 | 28.8% | 85.1% | 55.2 | 58 | 3600 | 59.8% | 83.9% | 49.4 | 58 | 10800 | 71.9% | 86% | 56.7 | 59 | 3600 | 50.8% | 84.3% |
| 6 | 52 | 51.2% | 83.3% | 52 | 52* | 23 | 51.6% | 83.2% | 44.5 | 52 | 10800 | 55.2% | 82.9% | 52 | 52* | 194 | 32.2% | 82.5% |
| 7 | 59 | 34.3% | 81.5% | 55.3 | 59 | 3600 | 47.9% | 84.1% | 48.8 | 59 | 10800 | 39.6% | 81.6% | 54.9 | 59 | 3600 | 52.7% | 81.9% |
| 8 | 49 | 26.1% | 82.9% | 48 | 48* | 13 | 63.7% | 85.5% | 41.6 | 48 | 10800 | 70.4% | 83.7% | 48 | 48* | 540 | 51.1% | 83.6% |
| 9 | 46 | 11.6% | 79.4% | 43 | 43* | 14 | 58.2% | 86% | 37.6 | 44 | 10800 | 64.1% | 82.6% | 44 | 44* | 322 | 65.1% | 83.1% |
| 10 | 60 | 9.5% | 72.3% | 50.1 | 53 | 3600 | 49.3% | 84.5% | 45.6 | 54 | 10800 | 55.4% | 81.5% | 52.7 | 58 | 3600 | 11.6% | 75.2% |
| Average | 53.9 | | 80.9% | | 52 | | | 84.4% | | 52.3 | | | 82.9% | | 52.8 | | | 82.1% |

Table 6.2: Opened rooms comparison (planned schedules)

| Instance | planned schedule | | | | | | | | | | |
|----------------|------------------|--------------|--------------------|--------------|---------|------------------------------------|--------------|---------|----------------------------------|--------------|---------|
| | OSU | | Deterministic MILP | | | First robust MILP ($\Gamma = 1$) | | | Second robust MILP ($B = 0.5$) | | |
| | # urgent | # priorities | # urgent | # priorities | CPU (s) | # urgent | # priorities | CPU (s) | # urgent | # priorities | CPU (s) |
| 1 | 2 | 51 | 0* | 0* | 1199 | 1 | 7 | 10800 | 0 | 6 | 3600 |
| 2 | 2 | 56 | 0 | 2 | 3600 | 0 | 5 | 10800 | 0 | 4 | 3600 |
| 3 | 8 | 68 | 0 | 7 | 3600 | 3 | 8 | 10800 | 0 | 1 | 3600 |
| 4 | 13 | 82 | 0 | 5 | 3600 | 1 | 6 | 10800 | 1 | 8 | 3600 |
| 5 | 6 | 68 | 0 | 9 | 3600 | 4 | 2 | 10800 | 0 | 10 | 3600 |
| 6 | 2 | 59 | 0 | 11 | 3600 | 1 | 4 | 10800 | 0 | 5 | 3600 |
| 7 | 28 | 98 | 0* | 0* | 1165 | 0 | 11 | 10800 | 1* | 4* | 2833 |
| 8 | 0 | 56 | 0* | 0* | 497 | 0 | 2 | 10800 | 0* | 3* | 1988 |
| 9 | 2 | 53 | 0 | 1 | 3600 | 0 | 12 | 10800 | 2 | 3 | 3600 |
| 10 | 3 | 35 | 0 | 2 | 3600 | 1 | 1 | 10800 | 2 | 1 | 3600 |
| Average | 6.6 | 62.6 | 0 | 3.7 | | 1.1 | 5.8 | | 0.6 | 4.5 | |

Table 6.3: Urgent and priority kits comparison (planned schedules)

6.5.2 Comparison of achieved schedules

We applied our simulation method presented in Section 5.2 (Algorithm 2) to obtain the achieved schedule for both robust models. Table 6.4 presents the achieved schedules obtained using the two robust models in addition to the deterministic method and OSU's achieved schedules. By comparing these results we see that the first robust model decreased the number of late ambulatories by approximately 35.3% compared with the second robust model, 45% compared with the deterministic model and by approximately 78.22% compared with the OSU's solution. Likewise, it managed to decrease the total overtime by approximately 38.3% compared with the second robust model, 45.95% compared with the deterministic model and by approximately 61.26% compared with the OSU's used overtime (savings of approximately 17 hours 32 minutes overtime compared with the 10 hours 39 minutes savings obtained by the second robust model and 8 hours 6 minutes savings obtained by the deterministic model). In addition, the 'max' column shows the maximum overtime found in a day during the month considered. In 5 instances, the first robust model managed to provide lower values than the second robust model. Moreover, in 7 instances, the first robust model managed to provide lower values than the deterministic model and the OSU's solution.

Next, the number of rooms opened does not change since our simulation method only uses rooms opened in the planned solution. Thus, both robust schedules still save the same numbers of ORs (1.6 ORs on average each month by the first robust model, 1.1 ORs by the second robust model, and 1.9 ORs by the deterministic model). Next, the first robust model managed to achieve an average occupancy rate for the rooms opened of approximately 88.4% which is less than the average in the deterministic solution (89.3%) since the deterministic solution uses fewer rooms and the rooms are more packed, but it achieved a better average occupancy rates than the second robust model (86.9%) and than the OSU (78.8%). Similarly, the minimum occupancy rate found in the first robust model schedules in 7 instances is greater than 50% (compared with 4 for the second robust model, 8 for the deterministic model, and 1 for the OSU). Thus, the first robust model managed to provide schedules with better ORs utilization than the second robust model.

Concerning the last criterion, the first robust model used approximately 0.7 urgent and 5 priority kits per month (5.7 problem kits per month), while the second robust model used approximately 0.4 urgent and 8.4 priority kits per month (8.8 problem kits per month). When

comparing these numbers with the achieved schedules of the deterministic model and OSU, we see that the first robust model managed to obtain the best values with a decrease of 90.1% over the problem kits of the OSU (57.3 problem kits per month), a decrease of 70.8% over the problem kits of the deterministic model (19.5 kits per month), and a decrease of 35.2% over the problem kits of the second robust model (8.8 kits per month). Hence, both robust models provided lower numbers of problem kits than the deterministic and OSU's solutions, with the first model outperforming the second one.

| Instance | Violated constraints | | Overtime | | ORs | | | Kits | | |
|------------------------------------|----------------------|----------------|---------------|------|-------------|---------------|----------------|------------|-------------|----|
| | #late amb | #violated kits | total | max | #opened | Min Occ. rate | Avg. Occ. rate | #urgent | #priorities | |
| OSU | 1 | 14 | 5 | 1382 | 190 | 59 | 29.9% | 79.2% | 3 | 39 |
| | 2 | 14 | 5 | 1692 | 227 | 59 | 12.9% | 76% | 3 | 44 |
| | 3 | 6 | 10 | 1322 | 113 | 48 | 19% | 79.3% | 8 | 49 |
| | 4 | 10 | 8 | 1800 | 98 | 48 | 44% | 82.9% | 14 | 64 |
| | 5 | 14 | 5 | 2370 | 130 | 59 | 29.7% | 81.9% | 6 | 58 |
| | 6 | 8 | 4 | 2140 | 201 | 52 | 50.8% | 79% | 1 | 53 |
| | 7 | 6 | 13 | 1454 | 170 | 59 | 33.1% | 80.6% | 28 | 72 |
| | 8 | 9 | 2 | 1584 | 100 | 49 | 23.7% | 81.5% | 4 | 46 |
| | 9 | 10 | 7 | 1500 | 126 | 46 | 12.3% | 77.3% | 8 | 42 |
| | 10 | 10 | 3 | 1930 | 326 | 60 | 9.9% | 70.7% | 4 | 27 |
| Average | 10.1 | 6.2 | 1717.4 | | 53.9 | | 78.8% | 7.9 | 49.4 | |
| Deterministic MILP | 1 | 4 | 0 | 979 | 135 | 58 | 65.1% | 87.8% | 0 | 11 |
| | 2 | 4 | 0 | 1268 | 189 | 55 | 31.4% | 89.7% | 0 | 24 |
| | 3 | 3 | 0 | 1305 | 156 | 46 | 55.0% | 89.8% | 0 | 14 |
| | 4 | 9 | 0 | 1745 | 130 | 48 | 53.9% | 90.7% | 1 | 17 |
| | 5 | 5 | 0 | 1990 | 200 | 58 | 68.3% | 89.1% | 0 | 44 |
| | 6 | 3 | 0 | 1034 | 206 | 52 | 71.4% | 89.4% | 0 | 34 |
| | 7 | 1 | 0 | 965 | 125 | 59 | 48.1% | 89.5% | 5 | 31 |
| | 8 | 3 | 0 | 1014 | 103 | 48 | 66.3% | 91.4% | 0 | 1 |
| | 9 | 6 | 0 | 950 | 137 | 43 | 75.7% | 88.8% | 0 | 7 |
| | 10 | 2 | 0 | 1060 | 240 | 53 | 54.3% | 87.0% | 0 | 6 |
| Average | 4 | 0 | 1231 | | 52 | | 89.3% | 0.6 | 18.9 | |
| First robust MILP ($\Gamma = 1$) | 1 | 1 | 0 | 418 | 59 | 58 | 58.3% | 88.0% | 1 | 5 |
| | 2 | 1 | 0 | 690 | 189 | 56 | 15.4% | 86.2% | 1 | 2 |
| | 3 | 3 | 0 | 777 | 75 | 46 | 54.9% | 89.9% | 2 | 9 |
| | 4 | 2 | 0 | 1041 | 133 | 48 | 64.3% | 90.9% | 0 | 4 |
| | 5 | 7 | 0 | 1265 | 137 | 58 | 40.8% | 88.8% | 3 | 3 |
| | 6 | 1 | 0 | 523 | 86 | 52 | 59.7% | 89.4% | 0 | 5 |
| | 7 | 0 | 0 | 546 | 98 | 59 | 57.2% | 89.7% | 0 | 8 |
| | 8 | 3 | 0 | 671 | 143 | 48 | 66.1% | 90.5% | 0 | 3 |
| | 9 | 3 | 0 | 494 | 98 | 44 | 58.2% | 87.0% | 0 | 9 |
| | 10 | 1 | 0 | 228 | 62 | 54 | 38.5% | 83.1% | 0 | 2 |
| Average | 2.2 | 0 | 665.3 | | 52.3 | | 88.4% | 0.7 | 5 | |
| Second robust MILP ($B = 0.5$) | 1 | 4 | 0 | 1407 | 112 | 58 | 49% | 88.1% | 0 | 12 |
| | 2 | 1 | 0 | 1015 | 189 | 56 | 32.7% | 87.8% | 0 | 12 |
| | 3 | 3 | 0 | 1787 | 141 | 46 | 57.1% | 88.6% | 0 | 11 |
| | 4 | 5 | 0 | 1301 | 213 | 48 | 46.1% | 89.8% | 0 | 8 |
| | 5 | 5 | 0 | 1509 | 155 | 59 | 32.0% | 86.7% | 0 | 15 |
| | 6 | 2 | 0 | 637 | 70 | 52 | 44.2% | 88.5% | 2 | 6 |
| | 7 | 4 | 0 | 954 | 103 | 59 | 54.3% | 87.9% | 0 | 4 |
| | 8 | 4 | 0 | 839 | 120 | 48 | 55.2% | 91.1% | 1 | 4 |
| | 9 | 4 | 0 | 1026 | 85 | 44 | 70.4% | 88.3% | 1 | 5 |
| | 10 | 2 | 0 | 307 | 45 | 58 | 10.6% | 72.4% | 0 | 7 |
| Average | 3.4 | 0 | 1078.2 | | 52.8 | | 86.9% | 0.4 | 8.4 | |

Table 6.4: Achieved schedules comparison

6.5.3 Results degradation

In order to analyse the performance of both robust models, we will analyse the degradation that is found in the achieved schedules. Table 6.5 compares the different objectives degradation

between both robust models, the deterministic one, and the OSU’s solution.

By analysing these results, we can see that both robust models provide more robust solutions with less degradation in all of the objective values when compared with the deterministic model. In addition, the solutions obtained using the first robust model had less degradation than the ones obtained with the second robust model in all criteria.

The degradation values in terms of overtime for both robust models are more than the ones found in the solution of the OSU, but the obtained overtime using the two models stays significantly better than the ones of the OSU. The same can also be said about the degradation values in terms of problem kits.

| | schedule | #late amb s | #not allowed kits | overtime | #problem kits (urgent + priorities) |
|------------------------------|-----------------|--------------------|--------------------------|-----------------|--|
| OSU | Planned | 7.6 | 10.2 | 804.1 | 69.2 |
| | Achieved | 10.1 | 6.2 | 1717.4 | 57.3 |
| | Degradation | +32.9% | -39.2% | +113.6% | -17.2% |
| NDSSCS (first model) | Planned | 0 | 0 | 286.3 | 6.9 |
| | Achieved | 2.2 | 0 | 665.3 | 5.7 |
| | Degradation | NA | - | +132.4% | -17.4% |
| NDSSCS (second model) | Planned | 0 | 0 | 294.2 | 5.1 |
| | Achieved | 3.4 | 0 | 1078.2 | 8.8 |
| | Degradation | NA | - | +266.5% | +72.5% |
| DSSCS | Planned | 0 | 0 | 215.5 | 3.7 |
| | Achieved | 4 | 0 | 1231 | 19.5 |
| | Degradation | NA | - | +471.2% | +427% |

Table 6.5: NDSSCS degradation analysis

6.6 Conclusion

In this chapter, we tackled the non-deterministic version of the static SCS problem of the CHU. We started by providing a technical review for the non-deterministic optimization field. Then, we proposed two mathematical models to solve the problem. Finally, we showed the numerical experiments for both models and compared the results with our deterministic approach and the original schedules of the OSU.

In our experiments, we showed that the first robust formulation provides results that are better in every aspect than the second model and the deterministic one, and shows more robustness against the uncertainties, in exchange for more computational time. But the second robust model, which needs way less time to find good solutions, may be attractive when computational time is critical and sought.

In the next chapter, we will tackle the non-deterministic dynamic version of the SCS problem.

NON-DETERMINISTIC DYNAMIC SURGICAL CASE SCHEDULING PROBLEM

7.1 Introduction

In Chapter 4, we solved the dynamic version of the deterministic SCS problem to overcome the uncertainty in patients arrivals. We proposed a new work flow that replaces the one currently implemented at the OSU by indicating only the due date of the surgery at the consultation step instead of the actual date of the surgery. We then implemented a rolling horizon approach to solve the dynamic scheduling problem, rendering our method applicable for the OSU. Despite the good results that were obtained using this method, there was a big degradation in the solutions qualities when applying surgeries real durations due to the big differences between surgeries estimated and real durations. To overcome this, we will tackle in this chapter the Non-Deterministic Dynamic Surgical Case Scheduling problem (NDDSCS) by adapting our two robust formulations presented in Chapter 6 to our dynamic method presented in Chapter 4.

In this chapter, we start by providing a brief technical review on the methods used to solve the non-deterministic dynamic scheduling problems in the literature. Next, we present our two robust models used in a rolling horizon scheme and compare the numerical results of both models with the deterministic model and the original data. Finally, we analyse the difference between the two models and conclude with this analysis.

7.2 Technical review

In the literature, the non-deterministic dynamic scheduling problem is solved using an approach composed of two phases: in a first phase a base-line predictive schedule that takes into account the uncertain aspects of some of the data is built using a static (off-line) algorithm; then at the second phase, this schedule is adapted at the execution moment to fit the state of the system using dynamic (on-line) algorithm. Depending on the generated base-line schedule and/or the method used to adapt the schedule to the random events, three main approaches can be found in the literature.

In the first method, a set of static schedules are generated instead of a complete base-line schedule, where it is easy to switch from one to another in response to random events that occur. This can be achieved by dividing tasks (jobs) into permutable groups for each one of the resources. In other words, all of the tasks from the same group are completely permutable without negatively affecting the desired performance. Thus, the result of this approach is a set of schedules obtained by enumerating all possible permutations within each group of tasks, among which the decision-maker can choose in real time the one he wishes to set up according to his preferences or in response to randomly occurring events. This method was used by [Wu et al., 1999] to minimize the weighted sum of delays in a Job-Shop problem. The authors present a Branch-and-Bound method for calculating a sort of "crucial subset" of scheduling decisions that gives a global view of the system. This subset is then completed at the moment of execution by decisions taken dynamically according to the disturbances that may occur. A similar problem was studied in [Artigues et al., 2005], where the authors propose a polynomial time dynamic programming algorithm for minimizing the number of groups and for maximizing the number of characterized sequences in order to maximise the solution flexibility. Next, the authors show the impact of grouping operations on the solution makespan value by showing computational results on Job-Shop benchmarks.

In the second method, a complete robust base-line schedule is generated at the first phase, and a "schedule repair" strategy is used to react to the random events. An example of this method is found in [Artigues et al., 2003], where the authors proposed a polynomial algorithm that inserts a new activity inside an existing solution represented by an activity-on-node (AON)-flow network. This algorithm extends to the RCPSP the concept of dominant insertion positions. The principle of their approach is that resources are considered as flows and that any resource unit used during the realization of an activity is transferred to another activity. Given a graph representing a set of schedules and an activity to be included in this graph according to the project pre-defined succession constraints and resources usage, the polynomial algorithm searches for an insertion position for this activity in the graph by minimizing the total duration of the project without modifying existing flows. In order to not disturb the pre-calculated solution too much, the authors require their algorithm to maintain the present activities sequence and try to insert the new activity while minimizing the total duration of the project. Finally, their algorithm applies a serial method to make the obtained schedule the active one. In their experiments, the authors showed that their algorithm in general produces more robust solutions than using rescheduling.

Finally in the last method, a complete robust base-line schedule is generated at the first phase, but a "complete rescheduling" strategy is used instead to react to the random events. An example of this method is found in [Addis et al., 2016], where the authors developed an ILP

to generate a robust base-line schedule and integrated their model in a rolling horizon method to perform the rescheduling. This rescheduling happens because of the semi-elective surgeries and cancelled surgeries (no time to perform).

In our problem, the first method is not suitable as it is impossible to guarantee that among the generated sequences from the first phase, a suitable solution will be found that is coherent with the disturbances that will be present in the second phase. Moreover, the CHU is only interested in having a fixed schedule for 1 month in advance, where this fixed schedule can not be modified. Given this requirement, the second method is unsuitable for our problem, as a schedule repair strategy means modifying the already fixed schedule. Hence, we propose to use the third method for our problem. To achieve this, we will implement an approach similar to the one used to solve the deterministic dynamic version of the problem (Chapter 4), except that at each iteration, we will solve a robust counterpart of the deterministic mathematical model (from Section 4.4). We present two robust formulations using the same methods in Chapter 6.

7.3 First formulation

The first robust formulation follows the one of Bertsimas and Sim ([Bertsimas and Sim, 2004]). We obtain the complete model from the one in Section 4.4 by introducing the following parameters:

- Γ the robust measure, representing the number of surgeries that can reach their maximum duration in each room r and each day t over a full length day (8h45).
- \bar{p}_i the duration of surgery i representing the central value of the normal distribution function $F(\tilde{p}_i)$ applied for all the surgeries from the same type as surgery i
- \hat{p}_i the maximum duration that surgery i can be increased by, representing the standard deviation parameter of the normal distribution function $F(\tilde{p}_i)$ applied for all the surgeries from the same type as surgery i

and by replacing constraint (4.4) by:

$$\sum_{i=1}^O \sum_{b=\beta}^{\gamma} \sum_{f=b}^{\gamma} \bar{p}_i \cdot x_{itr}^{bf} + \sum_{i=1}^O \eta_{itr}^{\beta\gamma} + \Gamma^{\beta\gamma} z_{tr}^{\beta\gamma} + \sum_{i=1}^O \sum_{b=1}^{\beta-1} \sum_{f=\gamma+1}^J d_{rt}^{\beta\gamma} \cdot x_{itr}^{bf} \leq d_{rt}^{\beta\gamma} + u_{\gamma} \cdot \varepsilon_{tr}, \quad (7.1)$$

$$\forall \beta \in \{1, \dots, J\}, \forall \gamma \in \{\beta, \dots, J\}, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\}$$

constraint (4.13) by:

$$\sum_{i=1}^O \sum_{b=\beta}^2 \sum_{f=b}^2 \bar{p}_i \cdot x_{itr}^{bf} + \sum_{i=1}^O \eta_{itr}^{\beta 2} + \Gamma^{\beta 2} z_{tr}^{\beta 2} + \sum_{i=1}^O \sum_{b=\beta}^3 a_i \cdot \bar{p}_i \cdot x_{itr}^{b3} + \sum_{i=1}^O a_i \cdot \eta_{itr}^{\beta 3} + \Gamma^{\beta 3} z_{tr}^{\beta 3}$$

$$\leq A_{max} - \sum_{j=1}^{\beta-1} d_{rt}^j, \quad \forall \beta \in \{1, \dots, 3\}, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\} \quad (7.2)$$

and by adding the following three constraints:

$$\eta_{itr}^{\beta\gamma} + z_{tr}^{\beta\gamma} \geq \sum_{b=\beta}^{\gamma} \sum_{f=b}^{\gamma} \hat{p}_i \cdot x_{itr}^{bf}, \quad (7.3)$$

$$\forall i \in \{1, \dots, O\}, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\}, \forall \beta \in \{1, \dots, J\}, \forall \gamma \in \{\beta, \dots, J\}$$

$$\eta_{itr}^{\beta\gamma} \geq 0, \quad (7.4)$$

$$\forall i \in \{1, \dots, O\}, \forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\}, \forall \beta \in \{1, \dots, J\}, \forall \gamma \in \{\beta, \dots, J\}$$

$$z_{tr}^{\beta\gamma} \geq 0, \quad (7.5)$$

$$\forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\}, \forall \beta \in \{1, \dots, J\}, \forall \gamma \in \{\beta, \dots, J\}$$

where:

$$\Gamma^{bf} = (d_{bf}/d_{1J}).\Gamma \quad (7.6)$$

$$d_{bf} = \max(d_{rt}^{bf}), \quad \forall r \in \{1, \dots, R\}, \forall t \in \{1, \dots, T\} \quad (7.7)$$

7.4 Second formulation

The second robust formulation follows the one presented in [Hans et al., 2008]. We obtain the complete model from the one in Section 4.4 by introducing the following parameters:

B a parameter to control the probability that a surgery will finish on time ($B \geq 0$).

\hat{p}_i the standard deviation of surgery i

and the following decision variables:

z_{trbf} non-integer variable representing the amount of slack given for day t at room r for surgeries that start at period b and end at period f .

and by replacing constraints (4.4) by:

$$\sum_{i=1}^O \sum_{b=\beta}^{\gamma} \sum_{f=b}^{\gamma} p_i \cdot x_{itr}^{bf} + \sum_{b=\beta}^{\gamma} \sum_{f=b}^{\gamma} z_{trbf} + \sum_{i=1}^O \sum_{b=1}^{\beta-1} \sum_{f=\gamma+1}^J d_{rt}^{\beta\gamma} \cdot x_{itr}^{bf} \leq d_{rt}^{\beta\gamma} + u_{\gamma} \cdot \varepsilon_{tr} \quad (7.8)$$

$$\forall \beta \in \{1, \dots, J\}, \gamma \in \{\beta, \dots, J\}, t \in \{1, \dots, T\}, r \in \{1, \dots, R\}$$

and constraints (4.13) by:

$$\sum_{i=1}^O \sum_{b=\beta}^2 \sum_{f=b}^2 p_i \cdot x_{itr}^{bf} + \sum_{b=\beta}^2 \sum_{f=b}^2 z_{trbf} + \sum_{i=1}^O \sum_{b=\beta}^3 a_i \cdot p_i \cdot x_{itr}^{b3} + \sum_{b=\beta}^3 z_{trb3} \leq A_{max} - \sum_{j=1}^{\beta-1} d_{rt}^j \quad (7.9)$$

$$\forall \beta \in \{1, \dots, 3\}, t \in \{1, \dots, T\}, r \in \{1, \dots, R\}$$

and finally by adding the following constraints to calculate the value of z_{trbf} :

$$B \cdot \sum_{i=1}^O (\hat{p}_i \cdot x_{itr}^{bf}) \geq z_{trbf} \quad (7.10)$$

$$\forall t \in \{1, \dots, T\}, \forall r \in \{1, \dots, R\}, \forall b \in \{1, \dots, J\}, \forall f \in \{b, \dots, J\}$$

7.5 Experimental results

In order to test the two proposed NDDSCS models, we applied each model in the rolling horizon scheme presented in Section 4.3. We then ran the experiments on the data described in Section 1.8 as a single instance representing all the surgeries performed at the OSU between 1/9/2014 and 30/6/2015, while considering surgeries due dates to be their actual dates.

7.5.1 First robust model results

We performed 3 experiments on the first robust model using different numbers of extra shifts opened for each surgeon ($h_s \in \{4, 6, 8\}$). The time limit was set to 12 hours per iteration (3 hours per objective). In addition, we set the maximum allowed over time in each room each day to the 3 hours allowed by the CHU ($\varepsilon_{max} = 180$) and the robustness measure $\Gamma = 1$.

During our experiments, we noticed that this model needs too much time to find a starting solution. After 6 iterations (construction of the schedule over 6 weeks), it was clear that the quality of the acquired solutions at the end of the time limit is worse than the ones obtained using the deterministic formulation. In particular, the total number of scheduled surgeries is around 75% of the ones obtained using the deterministic method. Table 7.1 compares the results obtained after 6 iterations from this method with the ones obtained using the deterministic model (from Table 4.1). Note that we consider only the runs with $\varepsilon_{max} = 180$ from the deterministic results. From these results, it is clear that the first robust model does not perform even closely to the deterministic one due to the high complexity.

| Method | h_s | # Scheduled |
|--------|-------|-------------|
| NDDSCS | 4 | 192 |
| NDDSCS | 6 | 201 |
| NDDSCS | 8 | 211 |
| DDSCS | 2 | 263 |
| DDSCS | 4 | 271 |

Table 7.1: NDDSCS first model planned schedule comparison (6 iterations)

In addition to these tests, we performed more experiments using different parameters combinations ($h_s \in \{2, 4, 6, 8\}$ and $\Gamma \in \{0.25, 0.5, 1\}$), but the results always suffered from the same low number of scheduled surgeries problem in comparison with the solutions of the OSU and deterministic method. Thus, it becomes evident that this model is not suitable as the main criteria for the CHU management is to reduce the costs while keeping the same level of quality of service represented by the total number of operated patients. Increasing the time limit could improve the solutions qualities, but this can not be done as 12 hours is the maximum allowed time by the CHU management.

7.5.2 Second robust model results

We performed 4 experiments on the second robust model using the following parameters:

- **Run 1:** $B = 0.5$, $h_s = 2$
- **Run 2:** $B = 0.5$, $h_s = 4$

- **Run 3:** $B = 0.25$, $h_s = 2$
- **Run 4:** $B = 0.25$, $h_s = 4$

Where B is a parameter to control the size of the added slack, and h_s represents the number of the extra shifts opened for each surgeon.

In addition, we set the time limit for the model to be 4 hours per iteration (1 hour per objective), and the maximum allowed over time in each room each day to the 3 hours allowed by the CHU ($\varepsilon_{max} = 180$).

The choice of the parameter values came after multiple experiments with different combinations ($B \in \{0.25, 0.5, 0.75, 1\}$ and $h_s \in \{0, 2, 4, 6, 8\}$), as choosing a value of B greater than 0.5 makes the model too conservative and yields worse results. Similarly, choosing 0 as a value for h_s gives the model too little room for improvements and less surgeries are scheduled, and choosing a value greater than 4 makes surgeries scattered in the horizon with less surgeries fixed at each iteration, since our rolling horizon method only fixes surgeries that are scheduled in the first week of the horizon. Similarly, we found that the model performs well in the given time limit and increasing it yields little to no improvements.

Comparison of planned schedules

Following the work process of the OSU, we start our comparison with the planned schedules, where Table 7.2 compares the obtained planned schedules with the ones of the OSU (Section 1.8.1) and the ones from the deterministic method (Section 4.5).

Starting with the first objective, our robust method managed to schedule around 97.4% of the total 2000 surgeries among the 4 runs with run 4 managing to schedule the most with 1962 scheduled surgeries (98.1%). These values lack behind the deterministic method but is expected due to the conservative nature of the robust method.

Moving forward, column '*Overtime*' represents the total overtime found in each schedule. All of the 4 robust runs managed to use lower overtime than the OSU, but lacked behind the deterministic method. In addition, using a lower value for B as in run 3 and 4 yields the lowest overtime used in all of the robust schedules with run 4 decreasing the used overtime by approximately 40.2% over the OSU. Similarly, as shown in column '*Max. overtime*' all of our runs has less maximum overtime than the OSU since it is a hard constraint.

Regarding the total number of opened ORs, our robust results show less total number of opened ORs than the OSU with an average of approximately 10 ORs closed per run over the OSU. It is important to note that the solver was not able to use these ORs to schedule more surgeries due to other contradicting constraints (ambulatory and kits constraints) and the 3 minimum week delay between the consultation and surgery date that we are posing in our method. The similar behaviour between the deterministic and non-deterministic methods continues as we can clearly see that increasing the flexibility margin given to the solver represented by the value of h_s improves other criteria, but increases the number of opened ORs, as run 2 has more opened ORs than run 1 and run 4 has also more ORs than run 3. Moreover, column '*Min. Occ. rate*' shows the minimum occupancy rate found for a room in each schedule. By comparing these results, we can clearly see that using a value of $h_s = 4$ for each B configuration yields the best minimum occupancy rates as runs 2 and 4 has the highest. The same also can be said about the average occupancy rates shown in column '*Avg. Occ. rate*' with runs 2 and 4 having the highest among the rest of the robust planned results. By

contrast, using $B = 0.5$ resulted in lower average occupancy rates than the ones of the OSU and the deterministic method, due to the higher conservation level.

Next, column '*# Urgent*' shows the total number of urgent kits and column '*# Priorities*' shows the total number of priority kits found in each schedule. Similar to previous criteria, using a value of $h_s = 4$ yields the best results as run 4 managed a decrease of 92.1% for the total number of problem kits (urgent + priorities) over the schedule of the OSU, while run 2 managed the best results among the other robust results with a decrease of 95.4% over the OSU. Despite that run 2 has higher B value ($B = 0.5$) and thus higher conservation level, this decrease is due to the lower total number of scheduled surgeries compared to run 4 with $B = 0.25$. The same can also be said when comparing the deterministic and robust results as the robust model was able to get similar results despite the conservation nature due to the lower number of scheduled surgeries compared to the deterministic method.

Finally, we compare the tardiness of surgeries from their due dates. Again we can see that runs 2 and 4 have the best robust results when comparing the total tardiness, maximum tardiness (column '*Max T_i* ') and average tardiness (column '*Avg T_i* '), with run 4 having the lowest values for the three criteria over all of our runs. One can note that the robust model performed worse than the deterministic one in all three tardiness criteria, which is again due to the conservative nature of the robust model.

| Schedule | | | | | | | | | | | | | |
|---|-------------------|-------|-------------|----------|---------------|--------------|----------------|----------------|----------|--------------|------------|-----------|-----------|
| Method | ϵ_{\max} | h_s | # Scheduled | Overtime | Max. overtime | # opened ORs | Min. Occ. rate | Avg. Occ. rate | # Urgent | # Priorities | $\sum T_i$ | Max T_i | Avg T_i |
| NDDSCS Run 1 ($B = 0.5$) | 180 | 2 | 1933 | 6378 | 161 | 529 | 19.8% | 79.9% | 9 | 98 | 12524 | 105 | 11 |
| NDDSCS Run 2 ($B = 0.5$) | 180 | 4 | 1937 | 6156 | 157 | 531 | 21.6% | 80.2% | 3 | 29 | 9243 | 89 | 8 |
| NDDSCS Run 3 ($B = 0.25$) | 180 | 2 | 1961 | 5274 | 173 | 528 | 23.4% | 81.1% | 8 | 117 | 11946 | 145 | 12 |
| NDDSCS Run 4 ($B = 0.25$) | 180 | 4 | 1962 | 4810 | 173 | 529 | 28% | 81.9% | 3 | 52 | 8124 | 77 | 10 |
| DDSCS | 180 | 2 | 1973 | 5829 | 180 | 527 | 29.7% | 81.5% | 10 | 126 | 13221 | 121 | 12 |
| DDSCS | 180 | 4 | 1981 | 3181 | 177 | 530 | 32% | 82.5% | 2 | 34 | 7279 | 77 | 9 |
| DDSCS | 90 | 2 | 1966 | 2322 | 90 | 528 | 22.8% | 81.6% | 9 | 108 | 10874 | 145 | 11 |
| DDSCS | 90 | 4 | 1974 | 1547 | 90 | 530 | 43.5% | 82.2% | 0 | 30 | 7516 | 121 | 9 |
| OSU | | | 2000 | 8041 | 266 | 539 | 9.5% | 80.9% | 66 | 626 | | | |

Table 7.2: NDDSCS planned schedules comparison

Comparison of achieved schedules

Moving forward, we applied our simulation method presented in Section 5.2 (Algorithm 2) to obtain the achieved schedule for the second robust model. Table 7.3 presents the achieved schedules obtained using the second robust model in addition to the deterministic method and OSU's achieved schedules.

We start by comparing the numbers of late ambulatory surgeries shown in column '*# late ambbs*', where all 4 robust runs had better results than the ones of the OSU and deterministic method, with an average of 7 late ambulatory surgeries among all robust runs and an average decrease of around 93.1% over the OSU and 41.7% over the deterministic method (average of 12 late ambulatory surgeries).

Next, the number of scheduled surgeries doesn't change here from the planned schedules since our simulation method doesn't add or remove surgeries from the planned schedule. Thus, run 4 still provides the best results with 1962 scheduled surgeries (around 98.1% of the total 2000 surgeries).

Regarding the overtime, we can clearly see the advantage of using the robust method in the 163 hours and 42 minutes saved over the OSU's solution (decrease of 57.2%), in comparison to the 128 hours and 51 minutes saved on average from the OSU's solution using the deterministic method (decrease of approximately 45%). Even limiting the maximum overtime ε_{max} per room per day to a lower value (90 minutes instead of 180) in the deterministic method gave worse results than those obtained with the robust method. In addition, we can see that providing the solver with a larger flexibility margin represented by h_s gives better overtime when fixing B , as shown with run 2 having better results than run 1 and run 4 having better results than run 3. Moreover, the maximum overtime found in any room any day (column '*Max. overtime*') is lower in all the robust schedules than both the deterministic and OSU's schedules, with run 2 having the best value of 198 minutes.

Next, column '*Min. Occ. rate*' shows the minimum occupancy rate found in a room in each schedule. Similar to the planned schedules comparison, using a value of $h_s = 4$ yields the best minimum occupancy rates as runs 2 and 4 has the highest values, as well as a better average occupancy rates as shown in column '*Avg. Occ. rate*'. In addition, all of our robust schedules provided an average occupancy rate greater than the 80% goal fixed by the CHU, which is not the case for the schedule of the OSU. On the contrary, all of our robust schedules provided lower values than the ones obtained using the deterministic method, which is due to the lower number of scheduled surgeries found in the robust schedules over the deterministic ones.

Finally, column '*# Urgent*' shows the total number of urgent kits and column '*# Priorities*' shows the total number of priority kits found in each schedule. Another advantage for using the robust method can be found here, as the robust schedules decreased the number of problem kits (urgent + priorities) on average by 498 kits over the OSU's solutions (decrease of approximately 86.9%), compared to an average savings of 370 kits obtained using the deterministic method over the OSU's solution (decrease of approximately 64.6%). Moreover, the same behaviour continues as using a higher value of $h_s = 4$ yields the best results as run 2 had the best result with a total number of problem kits of 51 kits with a decrease of approximately 91.1% over the 573 kits obtained by the schedule of the OSU.

This comparison between the achieved schedules shows globally that the results obtained using the second robust model are better in terms of the overtime and number of problem kits

than the ones obtained using the non-deterministic formulation and the ones of the OSU, with the only downside to the conservative nature of the robust model is the slight decrease in the total number of scheduled surgeries. In addition, we found that using increasing the flexibility margin ($h_s = 4$) for the solver yields better results.

| Schedule | | | | | | | | | | | |
|---|-------------------|-------|------------|-------------|----------|---------------|--------------|----------------|----------------|----------|--------------|
| Method | ϵ_{\max} | h_s | # Late amb | # Scheduled | Overtime | Max. overtime | # opened ORs | Min. Occ. rate | Avg. Occ. rate | # Urgent | # Priorities |
| NDDSCS Run 1 ($B = 0.5$) | 180 | 2 | 6 | 1933 | 6992 | 211 | 529 | 25.9% | 82.7% | 6 | 57 |
| NDDSCS Run 2 ($B = 0.5$) | 180 | 4 | 4 | 1937 | 6501 | 198 | 531 | 34.2% | 83.8% | 3 | 48 |
| NDDSCS Run 3 ($B = 0.25$) | 180 | 2 | 9 | 1961 | 8024 | 246 | 528 | 28.5% | 85.1% | 8 | 94 |
| NDDSCS Run 4 ($B = 0.25$) | 180 | 4 | 8 | 1962 | 7893 | 238 | 529 | 38.7% | 86.4% | 4 | 79 |
| DDSCS | 180 | 2 | 13 | 1973 | 13304 | 371 | 527 | 28.7% | 86.5% | 17 | 213 |
| DDSCS | 180 | 4 | 12 | 1981 | 11346 | 408 | 530 | 39.3% | 88.5% | 2 | 181 |
| DDSCS | 90 | 2 | 11 | 1966 | 10822 | 257 | 528 | 32.8% | 85.9% | 3 | 236 |
| DDSCS | 90 | 4 | 12 | 1974 | 9443 | 258 | 530 | 45.4% | 87.8% | 7 | 152 |
| OSU | | | 101 | 2000 | 17174 | 326 | 539 | 9.9% | 78.8% | 79 | 494 |

Table 7.3: NDDSCS achieved schedules comparison

7.6 Models performance analysis

In the previous sections, we adapted both robust models presented in Chapter 6 to our dynamic method presented in Chapter 4. Despite that the first robust formulation performed better than the second one in the static version of the SCS problem, the same couldn't be replicated in the dynamic version. The first robust model was not able to provide good solutions at the end of the 12 hours time limit, and increasing this time limit is not an option as discussed before due to the restrictions of the CHU's management. We tried different parameters combinations, but the same behaviour continued as this model needs too much time to find a starting solution and more to find improvements. Although this model was able to perform well in the described time limit frame in the static version, the same couldn't be said in the dynamic problem due to the higher number of surgeries considered at each iteration (an average of 400 instead of 200 for the static version) and the increased number of variables representing the horizon days (around 150 days considered at each iteration instead of an average of 26 days in the static version).

On the other hand, the second robust model performed better than the deterministic one and the OSU, despite the drop of 1.3% on average in total number scheduled surgeries compared to the deterministic model (1948 surgeries were scheduled on average among the 4 runs, compared to an average of 1973 surgeries among all 4 deterministic runs). For the rest of the criteria, the second robust model provided substantial gains.

Regarding the "robustness" of the second model, Table 7.4 compares the degradation in solution qualities between the planned and achieved schedules between the 4 runs of the second robust model, the 4 runs of the deterministic model, and the OSU schedules.

Starting with the overtime, we can see that all 4 runs had more robust solutions compared to the deterministic and OSU's solutions. Moreover, we can see that using a higher value of B increases the robustness of the model as both run 1 and 2 had less degradation than run 3 and 4, and using a higher value of h_s yields better results. Hence, run 2 had the most robust solution.

Similarly, all 4 runs show more robustness than both the deterministic and OSU's solutions when comparing the total number of urgent and priority kits. On the contrary to the behaviour pattern that we were seeing before ($h_s = 2$ provides worst results than $h_s = 4$), run 3 shows more robustness than run 2 represented by a decrease in the problem kits degradation values. This difference in behaviour can be justified as run 3 has double the number of problem kits of run 2 in the achieved schedule.

Finally, it is clear that increasing the value of B increases the robustness of the method and decreases the degradation, but at the cost of less number of scheduled surgeries. In addition, increasing the value of h_s gives the model more flexibility and thus yields better results as shown in both runs 2 and 4.

| | schedule | #late amb | #not allowed kits | overtime | #problem kits (urgent + priorities) |
|---|-------------|-----------|-------------------|----------|--|
| OSU | Planned | 76 | 102 | 8041 | 692 |
| | Achieved | 101 | 62 | 17174 | 573 |
| | Degradation | +32.9% | -39.2% | +113.6% | -17.2% |
| NDDSCS Run 1 ($\epsilon_{\max} = 180, h_s = 2, B = 0.5$) | Planned | 0 | 0 | 6378 | 107 |
| | Achieved | 6 | 0 | 6992 | 63 |
| | Degradation | NA | - | +9.6% | -41.1% |
| NDDSCS Run 2 ($\epsilon_{\max} = 180, h_s = 4, B = 0.5$) | Planned | 0 | 0 | 6156 | 32 |
| | Achieved | 4 | 0 | 6501 | 51 |
| | Degradation | NA | - | +5.6% | +59.4% |
| NDDSCS Run 3 ($\epsilon_{\max} = 180, h_s = 2, B = 0.25$) | Planned | 0 | 0 | 5274 | 125 |
| | Achieved | 9 | 0 | 8024 | 102 |
| | Degradation | NA | - | +52.1% | -18.4% |
| NDDSCS Run 4 ($\epsilon_{\max} = 180, h_s = 4, B = 0.25$) | Planned | 0 | 0 | 4810 | 55 |
| | Achieved | 8 | 0 | 7893 | 83 |
| | Degradation | NA | - | +64.1% | +50.9% |
| DDSCS ($\epsilon_{\max} = 180, h_s = 2$) | Planned | 0 | 0 | 5829 | 136 |
| | Achieved | 13 | 0 | 13304 | 230 |
| | Degradation | NA | - | +128.2% | +69.1% |
| DDSCS ($\epsilon_{\max} = 180, h_s = 4$) | Planned | 0 | 0 | 3181 | 36 |
| | Achieved | 12 | 0 | 11346 | 183 |
| | Degradation | NA | - | +256.7% | +408.3% |
| DDSCS ($\epsilon_{\max} = 90, h_s = 2$) | Planned | 0 | 0 | 2322 | 117 |
| | Achieved | 11 | 0 | 10822 | 239 |
| | Degradation | NA | - | +366.1% | +104.3% |
| DDSCS ($\epsilon_{\max} = 90, h_s = 4$) | Planned | 0 | 0 | 1547 | 30 |
| | Achieved | 12 | 0 | 9443 | 159 |
| | Degradation | NA | - | +510.4% | +430% |

Table 7.4: NDDSCS second robust model degradation analysis

7.7 Conclusion

In this chapter, we solved the non-deterministic version of the dynamic surgical case scheduling problem of the CHU. We integrated our two robust formulations presented in Chapter 6 in a rolling horizon approach. We used our testing methodology used in Chapter 4 and ran the numerical experiments on both models. Starting with the first robust model, we showed that this model was not able to perform well in the fixed time limit that was set by the CHU management. We showed both the planned and achieved schedules for this model after 6 iterations, which showed that the results lacked behind the deterministic ones in terms of the total number of scheduled surgeries, as it was only able to schedule around 75% of the total surgeries. This made the first model an unattractive option for the CHU, as their objective is to decrease the operational costs while maintaining the same level of service quality represented by the total number of operated patients.

Next, we presented our numerical experiments for the second robust model. Again, we used the same testing method as in the first robust model. We then compared both the planned and achieved schedules of this model with the ones obtained using the deterministic model and the ones of the OSU. In the results, we showed that this method provided substantial gains represented by a decrease of over 62.1% in the total overtime and 91% in the total number of urgent and priority kits. In addition, we showed that the second robust model outperformed the deterministic and OSU schedules in all criteria, with the only downside for its somehow conservative nature being the 1.3% drop in total number of scheduled surgeries compared to the deterministic method.

Finally, we showed that the solutions obtained using the second robust model provided more robustness as we compared the degradation between these schedules and the ones of the deterministic method and OSU.

GENERAL CONCLUSION AND PERSPECTIVES

This research has explored the surgical case scheduling problem with sterilising activities constraints of the Centre Hospitalier Universitaire d'Angers (CHU).

We started by explaining the SCS problem of the CHU and the main concerned elements, namely the OSU, the SU and the surgical kits. We then moved to describe the work process currently implemented at the OSU. From this description, we analysed the main problems imposed by using such a work flow and found that such problems are mainly a result of the lack of global planning. Next, we analysed the real data that we received from the CHU. This was achieved by exploring the planned schedules, which are the schedules generated using surgeries estimated durations, and the achieved schedules that use the real durations of surgeries and represent what actually happened at the OSU after performing the surgeries. By comparing these schedules, we showed that there is quite a big degradation in solution qualities when applying surgeries real durations in place of the estimated ones. Such degradation is a result of the stochastic nature of surgeries durations, which is showed in details in the last section by comparing surgeries estimated and real durations.

Second, we reviewed the literature on the operating rooms planning and scheduling problem. In the review, we observed that this problem is usually viewed as being made up of three phases corresponding to three decision levels. Thus, we visited the 3 decision levels while focusing on the operational level as it is where the SCS problem resides. Moreover, we found that researchers usually separate the SCS problem into two sub-problems, namely the AdvSP and AllocSP. We explored the literature on both sub-problems while classifying the literature on each sub-problem into four categories based on the availability of data (static or dynamic) and the quality of data (deterministic or non-deterministic). We then explained that decomposing the problem and solving each sub-problem individually negatively affects the quality of obtained solutions. For this, we explored the literature for the approaches that solve both sub-problems in a single step (AASP) following the same four categories classification we used before. During our review, we found that the SCS problem varies heavily in terms of the considered resources constraints and the solved objectives due to the different requirements of hospitals management. In addition, we noticed a lack of researches that take into account the medical instruments sterilisation step, despite its importance. Finally, we presented a summary and synthesis for the presented literature, where we showed that there are no existing researches in the literature that cover the needs of the CHU, while considering the medical instruments sterilisation step. Two of the main contributions of this thesis are the consideration of the sterilisation activities constraints while solving the SCS problem and the introduction of urgent and priority kits as performance measures for the problem.

Next, we proposed to solve the 4 versions of the SCS problem. Starting with the non-deterministic static version, we presented a formulation of the problem and proved that this problem is NP-hard. Next, we proposed a mixed integer linear programming formulation which

is solved in a lexicographic fashion. In addition, we proposed a constructive heuristic method that uses the presented MILP to solve the problem. According to the CHU, the objective we considered in this stage were the minimisation of the overtime, the number of opened ORs and the number of medical instruments treated as urgent or priority at the SU. To test our methods, we solved the problem with both the proposed MILP and heuristic method in order to create the planned schedules using surgeries estimated durations. Both our results significantly improve those of the CHU in terms of overtime and urgent kits at the SU. Moreover, the MILP provides better results than the heuristic, but needs much longer execution times.

Moving forward, we tackled the deterministic dynamic SCS problem. We started by providing a technical background for the dynamic scheduling problem and rolling horizon method. Next, we presented a formal description of the problem, during which we explained the dynamic nature of the scheduling problem at the OSU and proposed a new work flow for the OSU. In this new work flow, the surgeon only indicates the due date for the surgeries at the consultation instead of the actual surgery date. Then, the consulted surgeries are added to a waiting list to be scheduled later. Then, we presented a MILP formulation for the problem, where the objectives are first to maximise the number of scheduled patients, then minimise the total overtime at the OSU, then minimise the total cost of urgent and priority kits processed at the sterilizing unit, and finally minimise the total tardiness of surgeries from their due dates. Finally, we presented the experimental results following the same test methodology we used before. The results showed that our proposed method managed to schedule around 99% of the total number of surgeries and provide better results than the current method applied at the OSU in terms of overtime, numbers of opened ORs, occupancy rates at the ORs and numbers of urgent and priority kits.

Indeed, both deterministic static and dynamic solutions presented before were using surgeries estimated durations. For this, we developed an algorithm that simulates the process at the OSU and generates the corresponding achieved schedule for each planned one. When we compared the generated achieved schedules with the ones of the OSU, we found that our results in both problems outperformed the ones of the OSU in every criteria. Despite such big lead over the OSU, we noticed a certain amount of degradation in solution qualities when comparing each of our planned schedules to its corresponding achieved schedule due to the big differences between surgeries estimated and real durations.

To deal with such degradation, we tackled next the non-deterministic versions of the problem. Starting with the non-deterministic static SCS problem, we provided a technical review for the non-deterministic optimization field. Then, we proposed two robust mathematical models to solve the problem. During our experiments, we showed that both robust models provided better results than OSU and the deterministic method. In addition, the first robust model yields the best results and the least degradation, but uses way more time than the second robust model which needs less time to find good solutions.

Lastly, we solved the non-deterministic dynamic SCS problem. To achieve this, we used a similar approach to the one used for the deterministic dynamic problem and integrated our two robust formulations in the rolling horizon approach. Following the same testing methodology, we compared our results with the ones obtained using the deterministic method and the solutions of the OSU. We found that the first robust model was not able to schedule more than 75% of the surgeries achieved using the deterministic method in the fixed time limit that was set by the CHU management. This made the first model an unattractive option for the CHU, as the

main objective is to reduce the operational costs while maintaining the same level of service quality represented by the total number of operated patients. For the second robust model, we found that this method provided substantial gains represented by a decrease of over 62.1% in the total overtime and 91% in the total number of urgent and priority kits. In addition, we found that the second robust model outperformed the deterministic and OSU schedules in all criteria while also showing the most robustness against the data uncertainties, with the only downside for its somehow conservative nature being the 1.3% drop in total number of scheduled surgeries compared to the deterministic method.

The quality of the obtained results makes our non-deterministic dynamic method a very suitable and attractive decision aid tool for the CHU as it covers all the requirements of the OSU and provides significant savings compared to their actual implemented method, while requiring little modifications to their current work process to be fully implemented.

We identify several perspectives to further extend this thesis work.

First, to consider the semi-elective surgeries that are performed at the OSU in the dynamic version of the problem (surgeries with less than 3 weeks between the consultation and due date), we could implement a rescheduling step that is performed whenever such surgery is consulted. Indeed, this might lead to the disturbance of the fixed schedule, but we could deal with this by studying the probability of the arrival of such surgeries and insert empty slacks in the schedule to host such surgeries. In addition, we could also force the objective of such rescheduling step to minimise the changes done to the fixed schedule.

Another interesting avenue for future research is to jointly address the MSS and SCS problems. As we explained before, the CHU uses a block-scheduling strategy, where each surgeon is assigned blocks of OR time to schedule his/her surgeries, and this MSS is predefined 6 months in advance. Having such a fixed long period schedule decreases the flexibility of the method as in 6 months many things can change including the workload of each surgeon. Thus, it would be interesting to decrease such period by allowing the model to create the best suitable MSS for a shorter period (e.g. 1 month) in advance. Of course, this would require the addition of surgeons' preferences constraints in the model such as the working days, hours, consecutive days, vacations ...etc.

Finally, we could extend the definition of the problem by including more stakeholders' preferences. In the case of patients, such preferences could include but are not limited to: the hour of the surgery as patients who live outside the city prefer to be operated at the middle of the day to have enough time to come to the hospital and leave (if it is the case) at the same day, the day of the surgery as patients normally have other obligations (e.g. work, school, ..etc). For the surgeons, one interesting preference to add is the choice of surgeries. This is due to the fact that many surgeons have some specific type of surgeries that they do not prefer to perform more than once a day (i.e. requires too much time, effort, ...etc).

BIBLIOGRAPHY

- [Addis et al., 2016] Addis, B., Carello, G., Grosso, A., and Tànfani, E. (2016). Operating room scheduling and rescheduling: a rolling horizon approach. *Flexible Services and Manufacturing Journal*, 28(1-2):206–232.
- [Addis et al., 2014] Addis, B., Carello, G., and Tànfani, E. (2014). A robust optimization approach for the operating room planning problem with uncertain surgery duration. In *Proceedings of the international conference on health care systems engineering*, pages 175–189. Springer.
- [Artigues et al., 2005] Artigues, C., Billaut, J.-C., and Esswein, C. (2005). Maximization of solution flexibility for robust shop scheduling. *European Journal of Operational Research*, 165(2):314–328.
- [Artigues et al., 2003] Artigues, C., Michelon, P., and Reusser, S. (2003). Insertion techniques for static and dynamic resource-constrained project scheduling. *European Journal of Operational Research*, 149(2):249–267.
- [Association et al., 2003] Association, H. F. M. et al. (2003). Achieving operating room efficiency through process integration. *Healthcare financial management: journal of the Healthcare Financial Management Association*, 57(3):suppl-1.
- [Augusto et al., 2010] Augusto, V., Xie, X., and Perdomo, V. (2010). Operating theatre scheduling with patient recovery in both operating rooms and recovery beds. *Computers & Industrial Engineering*, 58(2):231–238.
- [Aytug et al., 2005] Aytug, H., Lawley, M. A., McKay, K., Mohan, S., and Uzsoy, R. (2005). Executing production schedules in the face of uncertainties: A review and some future directions. *European Journal of Operational Research*, 161(1):86–110.
- [Baker, 1977] Baker, K. R. (1977). An experimental study of the effectiveness of rolling schedules in production planning. *Decision Sciences*, 8(1):19–27.
- [Beliën and Demeulemeester, 2007] Beliën, J. and Demeulemeester, E. (2007). Building cyclic master surgery schedules with leveled resulting bed occupancy. *European Journal of Operational Research*, 176(2):1185–1204.
- [Beliën et al., 2006] Beliën, J., Demeulemeester, E., and Cardoen, B. (2006). Visualizing the demand for various resources as a function of the master surgery schedule: A case study. *Journal of medical systems*, 30(5):343.
- [Beliën et al., 2009] Beliën, J., Demeulemeester, E., and Cardoen, B. (2009). A decision support system for cyclic master surgery scheduling with multiple objectives. *Journal of scheduling*, 12(2):147.
- [Ben-Tal et al., 2009] Ben-Tal, A., El Ghaoui, L., and Nemirovski, A. (2009). *Robust optimization*, volume 28. Princeton University Press.

- [Ben-Tal and Nemirovski, 2000] Ben-Tal, A. and Nemirovski, A. (2000). Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical programming*, 88(3):411–424.
- [Beroule et al., 2016] Beroule, B., Grunder, O., Barakat, O., Aujoulat, O., and Lustig, H. (2016). Operating room scheduling including medical devices sterilization: towards a transverse logistic. *IFAC-PapersOnLine*, 49(12):1146–1151.
- [Bertsimas et al., 2011] Bertsimas, D., Brown, D. B., and Caramanis, C. (2011). Theory and applications of robust optimization. *SIAM review*, 53(3):464–501.
- [Bertsimas and Sim, 2004] Bertsimas, D. and Sim, M. (2004). The price of robustness. *Operations research*, 52(1):35–53.
- [Blake et al., 2002] Blake, J. T., Dexter, F., and Donald, J. (2002). Operating room managers? use of integer programming for assigning block time to surgical groups: a case study. *Anesthesia & Analgesia*, 94(1):143–148.
- [Blake and Donald, 2002] Blake, J. T. and Donald, J. (2002). Mount sinai hospital uses integer programming to allocate operating room time. *Interfaces*, 32(2):63–73.
- [Bowers and Mould, 2004] Bowers, J. and Mould, G. (2004). Managing uncertainty in orthopaedic trauma theatres. *European Journal of Operational Research*, 154(3):599–608.
- [Cardoen et al., 2009a] Cardoen, B., Demeulemeester, E., and Beliën, J. (2009a). Optimizing a multiple objective surgical case sequencing problem. *International Journal of Production Economics*, 119(2):354–366.
- [Cardoen et al., 2009b] Cardoen, B., Demeulemeester, E., and Beliën, J. (2009b). Sequencing surgical cases in a day-care environment: an exact branch-and-price approach. *Computers & Operations Research*, 36(9):2660–2669.
- [Cardoen et al., 2010] Cardoen, B., Demeulemeester, E., and Beliën, J. (2010). Operating room planning and scheduling: A literature review. *European Journal of Operational Research*, 201(3):921–932.
- [Choi and Wilhelm, 2014] Choi, S. and Wilhelm, W. E. (2014). On capacity allocation for operating rooms. *Computers & Operations Research*, 44:174–184.
- [Church and Uzsoy, 1992] Church, L. K. and Uzsoy, R. (1992). Analysis of periodic and event-driven rescheduling policies in dynamic shops. *International Journal of Computer Integrated Manufacturing*, 5(3):153–163.
- [Cowling and Johansson, 2002] Cowling, P. and Johansson, M. (2002). Using real time information for effective dynamic scheduling. *European journal of operational research*, 139(2):230–244.
- [Creemers et al., 2012] Creemers, S., Beliën, J., and Lambrecht, M. (2012). The optimal allocation of server time slots over different classes of patients. *European Journal of Operational Research*, 219(3):508–521.

- [Denton et al., 2007] Denton, B., Viapiano, J., and Vogl, A. (2007). Optimization of surgery sequencing and scheduling decisions under uncertainty. *Health care management science*, 10(1):13–24.
- [Denton et al., 2010] Denton, B. T., Miller, A. J., Balasubramanian, H. J., and Huschka, T. R. (2010). Optimal allocation of surgery blocks to operating rooms under uncertainty. *Operations research*, 58(4-part-1):802–816.
- [Dexter et al., 2002] Dexter, F., Blake, J. T., Penning, D. H., Sloan, B., Chung, P., and Lubarsky, D. A. (2002). Use of linear programming to estimate impact of changes in a hospital's operating room time allocation on perioperative variable costs. *The Journal of the American Society of Anesthesiologists*, 96(3):718–724.
- [Dexter et al., 2000] Dexter, F., Macario, A., and O'neill, L. (2000). Scheduling surgical cases into overflow block time—computer simulation of the effects of scheduling strategies on operating room labor costs. *Anesthesia & Analgesia*, 90(4):980–988.
- [Dexter et al., 1999] Dexter, F., Macario, A., Qian, F., and Traub, R. D. (1999). Forecasting surgical groups' total hours of elective cases for allocation of block time application of time series analysis to operating room management. *The Journal of the American Society of Anesthesiologists*, 91(5):1501–1501.
- [Dexter et al., 2003] Dexter, F., Traub, R. D., and Macario, A. (2003). How to release allocated operating room time to increase efficiency: predicting which surgical service will have the most underutilized operating room time. *Anesthesia & Analgesia*, 96(2):507–512.
- [Dimitriadis et al., 1997] Dimitriadis, A., Shah, N., and Pantelides, C. (1997). Rtn-based rolling horizon algorithms for medium term scheduling of multipurpose plants. *Computers & Chemical Engineering*, 21:S1061–S1066.
- [Dios et al., 2015] Dios, M., Molina-Pariente, J. M., Fernandez-Viagas, V., Andrade-Pineda, J. L., and Framinan, J. M. (2015). A decision support system for operating room scheduling. *Computers & Industrial Engineering*, 88:430–443.
- [Everett, 2002] Everett, J. E. (2002). A decision support simulation model for the management of an elective surgery waiting system. *Health care management science*, 5(2):89–95.
- [Fei et al., 2009] Fei, H., Chu, C., and Meskens, N. (2009). Solving a tactical operating room planning problem by a column-generation-based heuristic procedure with four criteria. *Annals of Operations Research*, 166(1):91.
- [Fei et al., 2008] Fei, H., Chu, C., Meskens, N., and Artiba, A. (2008). Solving surgical cases assignment problem by a branch-and-price approach. *International Journal of Production Economics*, 112(1):96–108.
- [Fei et al., 2006] Fei, H., Meskens, N., and Chu, C. (2006). An operating theatre planning and scheduling problem in the case of a "block scheduling" strategy. In *Service Systems and Service Management, 2006 International Conference on*, volume 1, pages 422–428. IEEE.
- [Fei et al., 2010] Fei, H., Meskens, N., and Chu, C. (2010). A planning and scheduling problem for an operating theatre using an open scheduling strategy. *Computers & Industrial Engineering*, 58(2):221–230.

- [Ghosh et al., 1995] Ghosh, S., Melhem, R., and Mosse, D. (1995). Enhancing real-time schedules to tolerate transient faults. In *Real-Time Systems Symposium, 1995. Proceedings., 16th IEEE*, pages 120–129. IEEE.
- [Ghosh and Mosse, 1996] Ghosh, S. and Mosse, D. (1996). *Guaranteeing fault tolerance through scheduling in real-time systems*. PhD thesis, University of Pittsburgh.
- [Glouberman and Mintzberg, 2001] Glouberman, S. and Mintzberg, H. (2001). Managing the care of health and the cure of disease—part i: Differentiation. *Health care management review*, 26(1):56–69.
- [Guerriero and Guido, 2011] Guerriero, F. and Guido, R. (2011). Operational research in the management of the operating theatre: a survey. *Health care management science*, 14(1):89–114.
- [Guido and Conforti, 2017] Guido, R. and Conforti, D. (2017). A hybrid genetic approach for solving an integrated multi-objective operating room planning and scheduling problem. *Computers & Operations Research*, 87:270–282.
- [Guimaraesa et al., 2015] Guimaraesa, L., Klabjanb, D., and Almada-Loboa, B. (2015). Rolling horizon formulations for short-to-medium term production planning.
- [Guinet and Chaabane, 2003] Guinet, A. and Chaabane, S. (2003). Operating theatre planning. *International Journal of Production Economics*, 85(1):69–81.
- [Hans et al., 2008] Hans, E., Wullink, G., Van Houdenhoven, M., and Kazemier, G. (2008). Robust surgery loading. *European Journal of Operational Research*, 185(3):1038–1050.
- [Herring and Herrmann, 2012] Herring, W. L. and Herrmann, J. W. (2012). The single-day surgery scheduling problem: sequential decision-making and threshold-based heuristics. *OR spectrum*, 34(2):429–459.
- [Herroelen and Leus, 2005] Herroelen, W. and Leus, R. (2005). Project scheduling under uncertainty: Survey and research potentials. *European journal of operational research*, 165(2):289–306.
- [Hsu et al., 2003] Hsu, V. N., De Matta, R., and Lee, C.-Y. (2003). Scheduling patients in an ambulatory surgical center. *Naval Research Logistics (NRL)*, 50(3):218–238.
- [Hulshof et al., 2012] Hulshof, P. J., Kortbeek, N., Boucherie, R. J., Hans, E. W., and Bakker, P. J. (2012). Taxonomic classification of planning decisions in health care: a structured review of the state of the art in or/ms. *Health systems*, 1(2):129–175.
- [Jackson, 2002] Jackson, R. (2002). The business of surgery. *Health Management Technology*, 23(7).
- [Jebali et al., 2006] Jebali, A., Alouane, A. B. H., and Ladet, P. (2006). Operating rooms scheduling. *International Journal of Production Economics*, 99(1-2):52–62.
- [Jensen, 2001] Jensen, M. T. (2001). Improving robustness and flexibility of tardiness and total flow-time job shops using robustness measures. *Applied Soft Computing*, 1(1):35–52.

- [Kharraja et al., 2006] Kharraja, S., Albert, P., and Chaabane, S. (2006). Block scheduling: Toward a master surgical schedule. In *Service Systems and Service Management, 2006 International Conference on*, volume 1, pages 429–435. IEEE.
- [Kirca and Kökten, 1994] Kirca, Ö. and Kökten, M. (1994). A new heuristic approach for the multi-item dynamic lot sizing problem. *European Journal of Operational Research*, 75(2):332–341.
- [Koppka et al., 2018] Koppka, L., Wiesche, L., Schacht, M., and Werners, B. (2018). Optimal distribution of operating hours over operating rooms using probabilities. *European Journal of Operational Research*, 267(3):1156–1171.
- [Kuo et al., 2003] Kuo, P. C., Schroeder, R. A., Mahaffey, S., and Bollinger, R. R. (2003). Optimization of operating room allocation using linear programming techniques. *Journal of the American College of Surgeons*, 197(6):889–895.
- [Lamiri et al., 2008a] Lamiri, M., Xie, X., Dolgui, A., and Grimaud, F. (2008a). A stochastic model for operating room planning with elective and emergency demand for surgery. *European Journal of Operational Research*, 185(3):1026–1037.
- [Lamiri et al., 2008b] Lamiri, M., Xie, X., and Zhang, S. (2008b). Column generation approach to operating theater planning with elective and emergency patients. *Iie Transactions*, 40(9):838–852.
- [Latorre-Núñez et al., 2016] Latorre-Núñez, G., Lüer-Villagra, A., Marianov, V., Obreque, C., Ramis, F., and Neriz, L. (2016). Scheduling operating rooms with consideration of all resources, post anesthesia beds and emergency surgeries. *Computers & Industrial Engineering*, 97:248–257.
- [Lehtonen et al., 2013] Lehtonen, J.-M., Torkki, P., Peltokorpi, A., and Moilanen, T. (2013). Increasing operating room productivity by duration categories and a newsvendor model. *International journal of health care quality assurance*, 26(2):80–92.
- [Luo et al., 2016] Luo, L., Luo, Y., You, Y., Cheng, Y., Shi, Y., and Gong, R. (2016). A mip model for rolling horizon surgery scheduling. *Journal of medical systems*, 40(5):127.
- [Ma and Demeulemeester, 2013] Ma, G. and Demeulemeester, E. (2013). A multilevel integrative approach to hospital case mix and capacity planning. *Computers & Operations Research*, 40(9):2198–2207.
- [Macario et al., 1995] Macario, A., Vitez, T., Dunn, B., and McDonald, T. (1995). Where are the costs in perioperative care?: Analysis of hospital costs and charges for inpatient surgical care. *The Journal of the American Society of Anesthesiologists*, 83(6):1138–1144.
- [Magerlein and Martin, 1978] Magerlein, J. M. and Martin, J. B. (1978). Surgical demand scheduling: a review. *Health services research*, 13(4):418.
- [Main, 1995] Main, O. (1995). What makes a well-oiled scheduling system? *OR Manager*.
- [Marcon and Dexter, 2006] Marcon, E. and Dexter, F. (2006). Impact of surgical sequencing on post anesthesia care unit staffing. *Health Care Management Science*, 9(1):87–98.

- [Marques and Captivo, 2017] Marques, I. and Captivo, M. E. (2017). Different stakeholders' perspectives for a surgical case assignment problem: Deterministic and robust approaches. *European Journal of Operational Research*, 261(1):260–278.
- [Marques et al., 2012] Marques, I., Captivo, M. E., and Pato, M. V. (2012). An integer programming approach to elective surgery scheduling. *OR spectrum*, 34(2):407–427.
- [Min and Yih, 2010a] Min, D. and Yih, Y. (2010a). An elective surgery scheduling problem considering patient priority. *Computers & Operations Research*, 37(6):1091–1099.
- [Min and Yih, 2010b] Min, D. and Yih, Y. (2010b). Scheduling elective surgery under uncertainty and downstream capacity constraints. *European Journal of Operational Research*, 206(3):642–652.
- [Molina and Framinan, 2009] Molina, J. M. and Framinan, J. M. (2009). Testing planning policies for solving the elective case scheduling phase: a real application. In *35th International Conference on Operational Research Applied to Health Services*.
- [Neyshabouri and Berg, 2017] Neyshabouri, S. and Berg, B. P. (2017). Two-stage robust optimization approach to elective surgery and downstream capacity planning. *European Journal of Operational Research*, 260(1):21–40.
- [Niu et al., 2011] Niu, Q., Peng, Q., El Mekkawy, T., Tan, Y. Y., Bruant, H., and Bernaerdt, L. (2011). Performance analysis of the operating room using simulation. *Proceedings of the Canadian Engineering Education Association*.
- [Ogulata and Erol, 2003] Ogulata, S. N. and Erol, R. (2003). A hierarchical multiple criteria mathematical programming approach for scheduling general surgery operations in large hospitals. *Journal of Medical Systems*, 27(3):259–270.
- [Ouelhadj and Petrovic, 2009] Ouelhadj, D. and Petrovic, S. (2009). A survey of dynamic scheduling in manufacturing systems. *Journal of scheduling*, 12(4):417–431.
- [Ozkarahan, 2000] Ozkarahan, I. (2000). Allocation of surgeries to operating rooms by goal programming. *Journal of Medical Systems*, 24(6):339–378.
- [Penn et al., 2017] Penn, M., Potts, C. N., and Harper, P. R. (2017). Multiple criteria mixed-integer programming for incorporating multiple factors into the development of master operating theatre timetables. *European Journal of Operational Research*, 262(1):194–206.
- [Persson and Persson, 2009] Persson, M. and Persson, J. A. (2009). Health economic modeling to support surgery management at a swedish hospital. *Omega*, 37(4):853–863.
- [Rachuba and Werners, 2017] Rachuba, S. and Werners, B. (2017). A fuzzy multi-criteria approach for robust operating room schedules. *Annals of Operations Research*, 251(1-2):325–350.
- [Riise and Burke, 2011] Riise, A. and Burke, E. K. (2011). Local search for the surgery admission planning problem. *Journal of Heuristics*, 17(4):389–414.
- [Roland et al., 2006] Roland, B., Di Martinelly, C., and Riane, F. (2006). Operating theatre optimization: A resource-constrained based solving approach. In *Service Systems and Service Management, 2006 International Conference on*, volume 1, pages 443–448. IEEE.

- [Roland et al., 2010] Roland, B., Di Martinelly, C., Riane, F., and Pochet, Y. (2010). Scheduling an operating theatre under human resource constraints. *Computers & Industrial Engineering*, 58(2):212–220.
- [Roshanaei et al., 2017] Roshanaei, V., Luong, C., Aleman, D. M., and Urbach, D. (2017). Propagating logic-based benders' decomposition approaches for distributed operating room scheduling. *European Journal of Operational Research*, 257(2):439–455.
- [Saadouli et al., 2015] Saadouli, H., Jerbi, B., Dammak, A., Masmoudi, L., and Bouaziz, A. (2015). A stochastic optimization and simulation approach for scheduling operating rooms and recovery beds in an orthopedic surgery department. *Computers & Industrial Engineering*, 80:72–79.
- [Sabuncuoglu and Bayız, 2000] Sabuncuoglu, I. and Bayız, M. (2000). Analysis of reactive scheduling problems in a job shop environment. *European Journal of operational research*, 126(3):567–586.
- [Sier et al., 1997] Sier, D., Tobin, P., and McGurk, C. (1997). Scheduling surgical procedures. *Journal of the Operational Research Society*, 48(9):884–891.
- [Soyster, 1973] Soyster, A. L. (1973). Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations research*, 21(5):1154–1157.
- [Sun and Xue, 2001] Sun, J. and Xue, D. (2001). A dynamic reactive scheduling mechanism for responding to changes of production orders and manufacturing resources. *Computers in Industry*, 46(2):189–207.
- [Testi et al., 2007] Testi, A., Tanfani, E., and Torre, G. (2007). A three-phase approach for operating theatre schedules. *Health Care Management Science*, 10(2):163–172.
- [Vali Siar et al., 2017] Vali Siar, M. M., Gholami, S., and Ramezani, R. (2017). Multi-period and multi-resource operating room scheduling and rescheduling using a rolling horizon approach: a case study. *Journal of Industrial and Systems Engineering*, 10:97–115.
- [Van Huele and Vanhoucke, 2014] Van Huele, C. and Vanhoucke, M. (2014). Analysis of the integration of the physician rostering problem and the surgery scheduling problem. *Journal of medical systems*, 38(6):43.
- [van Oostrum et al., 2008] van Oostrum, J. M., Van Houdenhoven, M., Hurink, J. L., Hans, E. W., Wullink, G., and Kazemier, G. (2008). A master surgical scheduling approach for cyclic scheduling in operating room departments. *OR spectrum*, 30(2):355–374.
- [VanBerkel and Blake, 2007] VanBerkel, P. T. and Blake, J. T. (2007). A comprehensive simulation for wait time reduction and capacity planning applied in general surgery. *Health care management Science*, 10(4):373–385.
- [Velásquez et al., 2008] Velásquez, R., Melo, T., and Küfer, K.-H. (2008). Tactical operating theatre scheduling: Efficient appointment assignment. In *Operations Research Proceedings 2007*, pages 303–308. Springer.
- [Vieira et al., 2000a] Vieira, G. E., Herrmann, J. W., and Lin, E. (2000a). Analytical models to predict the performance of a single-machine system under periodic and event-driven rescheduling strategies. *International journal of production research*, 38(8):1899–1915.

- [Vieira et al., 2000b] Vieira, G. E., Herrmann, J. W., and Lin, E. (2000b). Predicting the performance of rescheduling strategies for parallel machine systems. *Journal of Manufacturing Systems*, 19(4):256–266.
- [Vieira et al., 2003] Vieira, G. E., Herrmann, J. W., and Lin, E. (2003). Rescheduling manufacturing systems: a framework of strategies, policies, and methods. *Journal of scheduling*, 6(1):39–62.
- [Wachtel and Dexter, 2008] Wachtel, R. E. and Dexter, F. (2008). Tactical increases in operating room block time for capacity planning should not be based on utilization. *Anesthesia & Analgesia*, 106(1):215–226.
- [Wang et al., 2014] Wang, Y., Tang, J., and Fung, R. Y. (2014). A column-generation-based heuristic algorithm for solving operating theater planning problem under stochastic demand and surgery cancellation risk. *International Journal of Production Economics*, 158:28–36.
- [Wu et al., 1999] Wu, S. D., Byeon, E.-S., and Storer, R. H. (1999). A graph-theoretic decomposition of the job shop scheduling problem to achieve scheduling robustness. *Operations Research*, 47(1):113–124.
- [Zhang et al., 2009] Zhang, B., Murali, P., Dessouky, M., and Belson, D. (2009). A mixed integer programming approach for allocating operating room capacity. *Journal of the Operational Research Society*, 60(5):663–673.
- [Zhu et al., 2018] Zhu, S., Fan, W., Yang, S., Pei, J., and Pardalos, P. M. (2018). Operating room planning and surgical case scheduling: a review of literature. *Journal of Combinatorial Optimization*, pages 1–49.

NOTATIONS

To help the reader, we summarize below the main notations used in Part II and Part III.

Time horizon

- T : total number of days in the horizon
- J : number of periods in each day

Surgery

- O : total number of surgeries
- p_i : estimated duration of surgery i
- d_i : due date for surgery i
- \bar{p}_i : nominal duration of surgery i
- \tilde{p}_i : stochastic duration of surgery i
- \hat{p}_i : standard deviation of the duration distribution function $F(\tilde{p}_i)$
- a_i : $\begin{cases} 1 & \text{if surgery } i \text{ is ambulatory} \\ 0 & \text{otherwise} \end{cases}$
- A_{max} : latest time for ambulatory surgeries to be performed at

Operating rooms

- R : total number of operating rooms
- d_{rt}^j : $\begin{cases} \text{duration of period } j \text{ of day } t \text{ for room } r \\ 0 & \text{if the room is closed at day } t \end{cases}$
- d_{rt}^{bf} : duration from period b to period f ($\geq b$) of day t for room r
- ε_{max} : maximum allowed overtime for any room on any given day

Surgeons

- S : total number of surgeons

- J : number of periods in each day
- λ_s : set of surgeries for surgeon s
- δ_{srt} : $\begin{cases} 1 & \text{if surgeon } s \text{ can use room } r \text{ on day } t \\ 0 & \text{otherwise} \end{cases}$

Kits

- K : total number of kit types
- q_{ik} : required quantity of kits of type k for operation i
- Q_k : total quantity of kits of type k owned by the block
- cu : urgent kit penalty
- cp : priority kit penalty

LIST OF FIGURES

| | |
|--|-----|
| 1.1 The 2 patient classes | 16 |
| 1.2 Surgical kit example. Downloaded from http://jkinstrumente.com in June 2018 | 17 |
| 1.3 Washing machine type 1. | 18 |
| 1.4 Washing machine type 2. | 18 |
| 1.5 Wrapped kits example. Downloaded from https://shadeguide.com.au/ in June 2018 | 19 |
| 1.6 Kits sterilization 4 stages at the SU. | 19 |
| 1.7 The work process at the OSU. | 20 |
| 1.8 OSU kits movement at the CHU. | 21 |
| 1.9 Normal kit example. | 22 |
| 1.10 Priority kit (type 1) example. | 22 |
| 1.11 Priority kit (type 2) example. | 23 |
| 1.12 Urgent kit example. | 23 |
| 1.13 Not allowed kit example 1. | 24 |
| 1.14 Not allowed kit example 2. | 24 |
| 1.15 Number of surgeries for the different types. | 31 |
| 1.16 Difference between surgeries planned and real durations. | 32 |
| | |
| 3.1 The four periods of an operating room. | 60 |
| | |
| 4.1 Rolling horizon algorithm as in [Dimitriadis et al., 1997] | 79 |
| 4.2 DDSCS scheduling framework | 80 |
| 4.3 DDSCS rolling horizon scheme | 81 |
| 4.4 Consulted surgeries per week (initial waiting list at 0) | 86 |
| 4.5 DDSCS planned cumulative overtime comparison | 88 |
| 4.6 DDSCS planned cumulative urgent and priority kits comparison | 88 |
| | |
| 6.1 Comparison between time needed for both models to solve an objective (first objective for second instance) | 112 |

LIST OF TABLES

| | | |
|------|--|-----|
| 1.1 | Operating rooms working hours | 16 |
| 1.2 | SU's pickups and deliveries from the OSB | 21 |
| 1.3 | OSU's provided data characteristics. | 26 |
| 1.4 | Violated constraints in the OSU's planned schedule. | 26 |
| 1.5 | OSU's planned schedule overtime. | 27 |
| 1.6 | OSU's planned schedule opened rooms. | 27 |
| 1.7 | OSU's planned schedule urgent and priority kits. | 28 |
| 1.8 | Violated constraints in the OSU's achieved schedule. | 28 |
| 1.9 | OSU's achieved schedule overtime. | 29 |
| 1.10 | OSU's achieved schedule opened rooms. | 29 |
| 1.11 | OSU's achieved schedule expected and achieved idle times in minutes. | 30 |
| 1.12 | OSU's achieved schedule urgent and priority kits. | 30 |
| | | |
| 2.1 | Classification of the AdvSP bibliographical references | 51 |
| 2.2 | Classification of the AllocSP bibliographical references | 52 |
| 2.3 | Classification of the AASP bibliographical references | 52 |
| 2.4 | Thesis chapters position | 55 |
| | | |
| 3.1 | Reusing conditions for the kits. | 61 |
| 3.2 | Periods duration for the ORs. | 69 |
| 3.3 | Violated constraints in the OSU's planned schedule. | 69 |
| 3.4 | DSSCS planned overtime comparison (Objective 1). | 70 |
| 3.5 | DSSCS planned opened rooms comparison (Objective 2). | 71 |
| 3.6 | DSSCS planned urgent and priority kits comparison (Objective 3). | 72 |
| | | |
| 4.1 | DDSCS planned schedule comparison | 87 |
| | | |
| 5.1 | OSU achieved schedule. | 93 |
| 5.2 | DSSCS MILP achieved schedule. | 93 |
| 5.3 | DSSCS heuristic achieved schedule. | 94 |
| 5.4 | DSSCS degradation analysis | 95 |
| 5.5 | Achieved schedule directly from MILP | 96 |
| 5.6 | DDSCS achieved schedule comparison | 96 |
| 5.7 | DDSCS degradation analysis | 98 |
| | | |
| 6.1 | Overtime comparison (planned schedules) | 115 |
| 6.2 | Opened rooms comparison (planned schedules) | 115 |
| 6.3 | Urgent and priority kits comparison (planned schedules) | 116 |
| 6.4 | Achieved schedules comparison | 118 |
| 6.5 | NDSSCS degradation analysis | 119 |
| | | |
| 7.1 | NDDSCS first model planned schedule comparison (6 iterations) | 125 |

| | |
|---|-----|
| 7.2 NDDSCS planned schedules comparison | 128 |
| 7.3 NDDSCS achieved schedules comparison | 131 |
| 7.4 NDDSCS second robust model degradation analysis | 133 |

ACRONYMS

- AASP** Advanced and Allocation Scheduling Problem. 39, 48, 49
- AdvSP** Advanced Scheduling Problem. 33–37, 39–41, 48, 49
- AllocSP** Allocation Scheduling Problem. 33, 37, 39–41, 48, 49
- CG** Column Generation. 33, 35, 41, 47
- CGBH** Column-Generation-Based Heuristic. 40, 42, 45, 47
- CHU** Centre Hospitalier Universitaire d’Angers. 7, 8, 10, 12, 16, 19–22, 24, 26, 28, 48, 49, 52, 61, 63, 66–68, 70, 82–86, 89, 94, 95, 111, 115, 117, 118, 121, 124, 126
- GDP** Gross Domestic Product. 6
- ICU** Intensive Care Unit. 11, 32, 42, 43, 48
- ILP** Integer Linear Programming. 31, 33, 37, 45
- MILP** Mixed Integer Linear Programming. 8, 36, 37, 39, 41, 45–47, 56, 61, 63–66, 71–73, 75, 82, 83, 85–87, 102, 104, 115, 116
- MSS** Master Surgical Schedule. 29, 32, 33
- OR** Operating Room. 6, 12, 20, 24–26, 29, 31–43, 48, 55, 61, 63, 64, 67, 68, 73, 75, 79, 82, 86, 89, 98, 99, 104, 105, 107, 118, 119
- OSU** Orthopaedic Surgery Unit. 10–12, 14–25, 28, 52–54, 58, 62–65, 67, 72, 77, 79–91, 94, 95, 103–105, 107, 108, 110, 111, 113, 117–119, 121, 122, 124, 126
- PACU** Post-Anaesthesia Care Unit. 32, 38, 48
- RO** Robust Optimisation. 95
- SCS** Surgical Case Scheduling. 7, 8, 29, 30, 34, 39, 48, 49, 55, 56, 60, 66–68, 83, 85, 88, 90, 91, 95, 98, 111, 113, 115, 116, 124
- SP** Stochastic Programming. 95
- SU** Sterilization Unit. 10, 12–21, 28, 53, 55, 61, 63, 66, 72

Titre : Ordonnancement de blocs opératoires avec prise en compte des contraintes de stérilisation des instruments chirurgicaux.

Mots clés : Recherche opérationnelle, ordonnancement, gestion des blocs opératoires, programmation linéaire à variables mixtes, horizon roulant, optimisation robuste.

Résumé : Les blocs opératoires sont l'un des principaux postes de dépenses du système hospitalier, rationaliser et optimiser leur gestion permet donc une réduction des coûts pour la structure. S'aidant de l'unité de chirurgie orthopédique du CHU d'Angers, nous proposons donc des outils d'aide à la planification des interventions chirurgicales prenant aussi en compte les contraintes liées à la stérilisation d'instruments médicaux tels que les kits d'intervention. Le but de ces outils est de baisser les coûts de fonctionnement des blocs opératoires, optimiser le recours aux heures supplémentaires et les stérilisations de matériels en urgence, etc. Nous considérons premièrement que toutes les données sont connues et nous proposons un modèle de type MILP et une heuristique de construction de solutions dont les résultats obtenus améliorent la

planification du CHU. Nous adaptons ensuite une approche permettant d'assimiler l'arrivée dynamique des patients et montrons, résultats à l'appui, que cette technique permettrait d'améliorer le processus de prévision des opérations du bloc, si les durées opératoires sont connues.

Cette dernière hypothèse ne tenant pas dans le cas réel, nous suggérons de la lever en proposant de robustifier tout d'abord notre approche statique de deux façons que nous adaptons au cas dynamique. A l'issue de ces travaux, une amélioration de 54% est constatée du processus de planification en termes d'heures supplémentaires tout comme une réduction du nombre de stérilisations à effectuer dans l'urgence (90%) et d'une hausse significative du taux d'occupation des blocs opératoires (5.7%).

Title : Surgical case scheduling with medical instruments sterilizing activities constraints.

Keywords : Operations research, scheduling, operating room management, mixed integer linear programming, rolling horizon, robust optimization.

Abstract : The operating theater is considered as the most expensive and important resource in hospitals as it counts as the main source of income and expenses. This critical rule and the increase in costs urge hospitals to organize their processes more efficiently and effectively.

In this thesis, we will be working with the Centre Hospitalier Universitaire d'Angers (CHU) of Angers in France. We focus on the surgery scheduling problem at the orthopedic surgery unit. The main contribution of this work is the consideration of the activities of the sterilizing unit as a hard constraint and a performance measure for the problem.

In the first part of this work, we present a multidimensional classification of the current literature on the surgical case scheduling problem.

In the second part, we solve the deterministic version of the problem. Starting with the static problem, we

propose a MILP and a constructive heuristic and show that the obtained results significantly improve over the ones of the CHU.

Next, we solved the deterministic dynamic version by implementing our MILP in a rolling horizon approach. Again, the results were superior to the CHU ones.

We then showed that a non-deterministic approach is a must due to the big degradations caused by surgeries duration uncertainties.

In the third part, we tackled the non-deterministic version of the problem. Similarly, we started with the static problem and proposed two robust models.

Finally, we implement both robust models in a rolling horizon method to solve the dynamic scheduling problem. The results of the both non-deterministic versions show much more robustness compared to the deterministic ones and better values overall.