Optimizing Resource Allocation in Surgery Delivery Systems

by

Maya Bam

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Industrial and Operations Engineering)
in The University of Michigan
2017

Doctoral Committee:

Professor Brian T. Denton, Co-Chair
Professor Mark P. Van Oyen, Co-Chair
Professor Mark S. Daskin
Associate Professor Richard E. Hughes

Maya Bam

mbam@umich.edu

ORCID iD: 0000-0002-8757-6285

To my Lord and Savior, Jesus Christ.

Without Him I am nothing, with Him I can do all things.

ACKNOWLEDGEMENTS

There are so many people I owe thanks to for helping me throughout this journey. I would like to thank the teachers who planted and nourished my thirst for knowledge over the years. I owe special thanks to my two advisers and mentors, Brian Denton and Mark Van Oyen. I have learned so much from you through the years, you have taught me how to learn, how to teach, how to be a good colleague and collaborator. Thank you for your encouragements, patience, and support. I cannot imagine anyone else guiding me through the winding roads of my graduate studies. Furthermore, I gratefully acknowledge the support that I received through the Rackham Merit Fellowship and from the Department of Industrial and Operations Engineering at the University of Michigan, and from the National Science Foundation under grant numbers CMMI 0844511 (Denton), and CMMI 1233095 and 1548201 (Van Oyen).

I would also like to thank the remaining members of my committee for their input and guidance: Mark Daskin, thank you for asking me to teach with you, and for always being supportive of my ideas; Richard Hughes, thank you for telling me about the graduate student path, and advising me on how to make it through. I am also very grateful to my collaborators, Mark Cowen and Mary Duck for their guidance and help throughout my studies, and for providing a realistic setting for my research.

I also owe gratitude to my professors at Gordon College, who steered me towards mathematics, when I was on such a different path. Michael Veatch, thank you for

introducing me to the field of operations research, and for starting me down the path that lead me here. Richard Stout, thank you for never kicking me out of your office, despite never coming during office hours. Jonathan Senning, thank you for making even grading fun by telling me which homework exercises to grade for your class as solutions to puzzles you created for me. Karl-Dieter Crisman, thank you for showing me that I did not have to choose between my love of music and mathematics. Your care and support played an essential part in helping me get to this point.

My thanks goes out to my friends and heavenly family at Cornerstone Church in Beverly, MA, and Christ Church Ann Arbor. Your love and prayers have sustained me through the transition into this country with its many hardships, and the challenges of higher education. You have taught me a great deal about the world and myself, and I have grown so much in your community. You have been great blessings in my life.

Thank you to my classmates, officemates, next door neighbors, and many friends in the department and beyond, too many to list all by name, for always being there for me, and listening to me when I needed to talk, and talking when I needed to listen. I am especially grateful to Kayse Maass, Pooyan Kazemian, Victor Wu, Armando Bernal, and Abdullah Alshelahi. You have encouraged me throughout the years, and you made me look forward to coming into the office every day. I owe special gratitude to Zheng Zhang, who has been such a great help during my studies, for showing me what a fruitful collaboration among students is like.

Thank you to my classmate and roommate, Emily Speakman, with whom I shared the grad school experience from its beginning to its end. Thank you for singing with me on bad days, and celebrating with me on good ones. Thank you for helping me

in my indecisive moments, and for always being a positive influence.

I owe gratitude to my family for their unwavering support throughout the years. Thank you to my aunt and uncles, Tania Bam, Shlomo Bam, and Albert Doka, for believing that I can achieve what I set out to do. Thank you to my father, Michael Bam, and his family, Lilia, Dina, and Adam, for bringing me to this country, and helping me through that hard transition with such a warm welcome. You have given me one of the greatest gifts possible by providing a college education for me. Words cannot express my gratitude for that.

Thank you to my grandmother, Magdalena Doka, for taking me to my lessons after school, cooking me my favorite meals, and creating many more fond memories that I will treasure. I wish you were here, and I could share this achievement with you.

I am especially grateful you to my mother, Magdalena Bam, who has raised me to be the person I am today. You are my inspiration, I admire you so much. I have never known anyone who is a more worthy example, and I strive to follow in your footsteps. Without your love and support I would have never come this far. Thank you for being there for me every step of the way.

Above all, I owe my deepest gratitude to my Lord and Savior, Jesus Christ, for putting all these people in my life, and giving me the endurance and strength to arrive at this milestone.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

ABSTRACT

Optimizing Resource Allocation in Surgery Delivery Systems

by

Maya Bam

Co-Chairs: Brian T. Denton and Mark P. Van Oyen

This thesis focuses on developing mathematical models to optimize processes related to surgery delivery systems. Surgical services account for a large portion of hospital revenue and expenses; moreover, increased demand is expected in the future due in part to the aging population in many countries. Achieving high efficiency in this system is challenging due to the uncertain service durations, the interaction of different stages of the system (e.g., surgery, recovery), and competing criteria (e.g., patient wait time, employee satisfaction, the availability and utilization of healthcare professionals, operating rooms (ORs), and recovery beds). Moreover, solutions must overcome an enormous barrier of computational complexity.

Considering the complexity of the problem, and the numerous resources involved in delivering surgical care, this thesis focuses on three aspects of surgery delivery systems: short term scheduling (operational level decisions, e.g., daily sequencing of surgeries), service group team design and staff allocation (strategic level team design decisions on the order of years, and tactical level shift allocation decisions, e.g., monthly), and OR capacity reservation (strategic level decisions, e.g., what OR ca-

pacity reservation policy to use in the following years).

To optimize scheduling policies on an operational level, we developed a 2-phase approximation method, where the first phase determines the number of ORs to open for the day, and assigns surgeons to ORs. The second phase performs surgical case sequencing considering recovery resource availability. For both phases of the approximation, we provide provable worst-case performance guarantees; furthermore, we use numerical experiments to show the methods also have excellent average case performance. We further developed a mixed integer programming (MIP) model for comparison to the approximation method. We evaluated the performance of the approximation compared to the MIP model in deterministic and stochastic settings, using a discrete even simulation (DES) for the latter.

On the strategic and tactical levels, we focus on staffing decisions for surgical nurses. These decisions present a challenge due to nurse availability, skill requirements, hospital regulations, and stochastic surgical demand. We present a MIP to group services into teams, and achieve fairness in training time and overnight surgical volume, and balance size across teams. Once teams are created, we use a MIP-based heuristic to assign shifts to services and teams to ensure coverage of surgical demand. We analyze the performance of the heuristic, and present results that provide insight into optimal surgical nurse staff planning decisions. We show that the newly designed teams are more balanced with respect to the performance metrics, and coverage of surgical demand can be improved.

Finally, on the strategic level, we use DES to evaluate OR capacity reservation heuristics. OR capacity reservation is a challenging problem due to uncertain demand for surgery and surgery durations. Using our DES model, we evaluate two categories

of approximation methods to gain insights into the problem: first come, first served based heuristics, which are used as benchmarks, and appointment slot reservation heuristics, similar to those used in outpatient clinics. We compare the heuristics based on the mean percent of patients that exceed a predefined surgery access target, mean patient wait time, and mean OR utilization.

This research was conducted in collaboration with hospitals, and the problems considered are common to many hospitals. Based on data from these hospitals, we provide evidence that significant improvements could be achieved in the three major decision making levels.

CHAPTER I

Introduction

1.1 Motivation

Hospital surgical services are sources of both great revenue and high expenses for human and physical resources. Within hospitals, 68% of revenues are directly related to surgery, while 20-40% of costs are associated with operating rooms (ORs) (*Jackson* (2002)). Due to this large financial contribution, there is a very high cost associated with inefficient planning and scheduling of ORs. Moreover, studies suggest that demand for surgery will increase by 14—47% by 2020, where the wide range is due to differences in specialty (*Etzioni et al.* (2003)). Furthermore, "aggregate surgical expenditures are expected to grow from $574 billion in 2005 (4.6% of US GDP) to $912 billion (2005 dollars) in the year 2025 (7.3% of US GDP)" (*Muñoz et al.* (2010)). If these predictions are correct, and surgical volume increases in the future, inefficient use of ORs, staff overtime costs, and poor scheduling of ORs will have an increasing financial impact on hospitals. Therefore increased efficiency will become even more important in years to come.

The surgery delivery system is a complex system with many constraints and much inherent uncertainty. To appreciate the complexities and nuances of this system, we start by describing some of the general background on surgery delivery systems,

including different types of surgeries, different types of resources, and patient flow through the system. Next we give a chapter-by-chapter summary of the remainder of this thesis.

1.2 Background on Surgery Delivery Systems

Hospitals generally distinguish at least two patient urgency levels: elective and emergent. *Elective patients* are patients who are scheduled to have surgery weeks, or in some cases months in advance. Since these patients are scheduled well in advance, they comprise the more reliable portion of the daily *surgical schedule*, which includes the list and sequence of surgeries, and their allocation to OR on any given day. *Emergent patients*, on the other hand, generally have life threatening conditions, that require immediate surgical intervention. These types of patients arise on short notice, and are hard to anticipate in advance. Therefore these patients cause disruptions in the surgical schedule, and can cause the rescheduling, or in some cases the canceling of elective patients. In some cases hospitals dedicate OR capacity to emergent cases to avoid the need for rescheduling and cancellations. Some hospitals also consider a third urgency level, *urgent patients*. Similar to emergent patients, urgent patients also arise on short notice with severe conditions, but urgent patients are able to tolerate some wait time for surgery (usually about one or two days).

A natural question is how elective surgeries, surgeries that are arranged well in advance are scheduled. In most hospitals, a surgeon can only schedule a surgery if they or their service has *block time* allocated to them, or if there is open OR time available. The notion of block time is associated with block scheduling, an approach commonly used in many hospitals. The basic idea of block scheduling is that either a surgeon or a service is guaranteed the use of a set of ORs for either the entire day or a fraction of a day, and this reservation is known well in advance. Block scheduling

places a limit on the number of cases a surgeon can perform and on the choices for assigning surgeons to ORs when there are limited number of ORs available. However, most hospitals have methods by which unused block time is released and reallocated as the start of the surgical day approaches, typically a few days before the day of surgery. Thus, the actual day-of scheduling may break the constraints of the block schedule.

There are many resources in the surgery delivery system that support surgery before, on, and after the day of surgery. Elective patients are usually first seen by their surgeon during the surgeon's clinic hours. During this appointment, if the surgeon and patient agree that surgery is the best course of action, the surgery gets added to a surgical listing for a future date on which block time is available. This clinic appointment often creates a patient-surgeon assignment, i.e., the patient agreed with that specific surgeon that they will perform their surgery, and this agreement has to be respected. The patient might immediately receive a surgery date and time; however, many services prefer to inform patients of surgery arrival time closer to the date, when more information is available about the surgery schedule for that day.

On the day of surgery, patients who are coming from home go through a check-in process that includes administrative paperwork in preparation for surgery, after which they are taken to the pre-operative unit ("preop" for short). Patients already in the hospital are taken directly to preop. Patients are seen by multiple members of the surgical team in preop: surgical nurse(s), anesthesiologist(s), surgeon(s). Once all confirm the procedure, the surgical team is ready for the surgery, and the OR is ready, the patient is taken to the OR, where the surgical procedure is performed. After the procedure, most patients are transfered to the post-anesthesia care unit (PACU), where patients recover from the effects of anesthesia and are monitored to

ensure they are stable. Recovery in the PACU usually takes a few hours, after which patients are transfered to an inpatient hospital bed that corresponds to their surgical service (e.g., orthopedic bed, general bed). More severe patients can be transfered to the intensive care unit (ICU) after the procedure, where they can remain for days, until their condition improves.

Following the procedure, patients stay at the hospital until it is safe for them to return home. For some elective cases patients return home on the same day. In the weeks and months after the surgery, patients have follow-up appointments with their surgeons, and possibly other healthcare providers, to ensure that the procedure was successful, and their recovery proceeds as expected.

In this complex system surrounding surgery there are many opportunities for improvements. This thesis focuses on developing mathematical models to improve three aspects of this system: surgery sequencing (operational level decisions), team design and staff shift allocation (strategic and tactical level decisions), and OR capacity reservation (strategic level decisions). The goal of each project is to develop and test fast approximation methods that provide good solutions to the problems considered, provide insight into the problem, and aid implementation. All of the methods we propose are evaluated using test cases developed using real data from hospitals, to demonstrate the potential for impact. In the next sections we summarize each of the remaining chapters.

1.3 Chapter II: Surgery Scheduling with Recovery Resources

In Chapter II we address the operational level decisions associated with sequencing of elective surgery patients on the day of surgery, considering not only resources that are directly related to surgery (e.g., surgeon, OR), but also resources indirectly

related to surgery (e.g., PACU). These are decisions that hospitals face daily, and often have to be made close to the day of surgery, and sometimes on the day of surgery if rescheduling of cases is necessary. Sequencing decisions are usually made based on OR and surgeon availability, without the consideration of supporting resources, such as the PACU. As we will show, consideration of this additional resource in the decision making process greatly increases the complexity of the problem.

Despite the challenges presented by considering the PACU, it is an important resource to consider, as its resource shortages can cause wasting of OR time. To illustrate this point, consider the following example. Suppose that a patient's surgery is finished in the OR, and the patient is ready to move to the PACU, but there are no PACU beds available for the patient. Then the patient has to start the recovery process in the OR. This phenomenon is called *OR boarding*. It is a disadvantage for the hospital, as the OR, which is an extremely expensive resource in the system, is not being used for its designed purpose, and thus OR time is wasted. Moreover, OR boarding can cause delays for subsequent patients. Therefore it is important for both the hospital and patients to strive to avoid OR boarding.

We designed our methods with the goal in mind to improve resource utilization of surgeons and ORs, with an emphasis on avoiding OR boarding. We developed a new deterministic mixed integer program (MIP) formulation for the elective surgery scheduling problem, that considers surgeons, ORs, and the PACU, which allowed us to analyze how shortages of one resource can affect the others. The objective of this model is to minimize the weighted sum of the fixed cost of opening ORs, the variable cost of OR overtime, and the variable cost of surgeon elapsed time, which is taken from the time when a surgeon starts their first surgery until the completion of their last surgery (thus including both working and idle time). Constraints include

ensuring there is no OR boarding, respecting patient-surgeon assignments, and ensuring that each surgeon performs all their surgeries consecutively to mimic a block schedule. The optimization model considers deterministic surgeon and case specific surgery and recovery durations. These deterministic durations are carefully chosen from the duration distributions as a percentile to mitigate the effect of uncertainty on the surgery schedule, and increase its reliability.

Due to the complexity of the system, realistic problem instances are challenging to solve with the optimization model. To overcome this challenge, we developed a fast 2-phase heuristic that separates the problem into 2 phases. First, the number of ORs to open is determined, and surgeons are assigned to the opened ORs without considering PACU resources. The objective of this phase is to minimize the fixed cost of opening the ORs, and the variable cost of OR overtime. Once this surgeon-to-OR assignment is set, now considering the PACU, patients assigned to the same surgeon are sequenced, and surgeons assigned to the same OR are sequenced, creating a complete surgery schedule. The objective of the second phase is to minimize surgeon elapsed time. Similar to the optimization model, this is also a deterministic model. We provide tight worst-case performance guarantees for both phases, and show that the heuristics perform extremely well in the deterministic setting.

We compare the 2-phase heuristic to a decomposed version of the MIP formulation, where the first stage corresponds to decisions made in the first phase of the heuristic, and the second stage corresponds to decisions made in the second phase of the heuristic. Schedules obtained from this decomposition heuristic are used as a benchmark against the 2-phase heuristic schedules.

To understand how the created schedules perform in the presence of uncertainty in

surgery and recovery durations, we also developed a discrete event simulation model that is used to evaluate both the schedules generated by the 2-phase heuristic and the decomposition heuristic. We show that the 2-phase heuristic performs extremely well in the stochastic setting, when compared to optimization based surgery schedules, and we also provide evidence that hospital performance can be improved using our methodology, through a case study that compares hospital schedules to 2-phase heuristic schedules.

1.4 Chapter III: Planning Models for Skills-sensitive Surgical Nurse Staffing

In this chapter we expand the time horizon considered in Chapter II, and look at the problem of surgical nurse staffing. As opposed to the approach of Chapter II, where methods were developed to make operational level short term decisions, we look at this problem on the strategic and tactical levels to design service group teams that remain unchanged for years, and nurse shift staffing schedules that are usually updated monthly.

Surgical nurses are essential parts of the surgery delivery system. Nurses see patients on the day of surgery to confirm their procedure, and they also play an essential part during the surgical procedure itself. Hospitals generally distinguish between two surgical nurses: *surgical technicians*, who work within the sterile field, and are responsible for handing instruments to surgeons; and *circulator nurses*, who work outside of the sterile field, and are responsible for getting the OR ready for surgery, charting during surgery, and obtaining additional instruments, if necessary.

Surgical nurses are a highly specialized part of the workforce. It requires months

of training for a surgical nurse to be able to assist in surgeries without supervision. Due to the large number and complexity of surgical services, nurses specialize in a subset of those services to ensure they can maintain high skill levels in their chosen services. To aid the specialization of surgical nurses, hospitals often divide their surgical services into disjoint teams, where nurses would train in a single team, i.e., all services in their team, and would be assigned to work in that team upon finishing their training.

The focus of Chapter III is twofold. First, we present a model to design service group teams of surgical services in a way that balances factors that contribute to fairness to surgical nurses. The first such factor is training time. As mentioned, surgical nurses are trained over an extended period, and each team has the same number of months available for training. But the time needed to adequately learn the necessary skills can significantly differ across current teams. Thus, to be fair to nurses across teams, part of the objective is to balance the training time across teams. The second factor our model balances is overnight surgical volume, to avoid having a single team fulfilling most of the overnight surgical demand, but instead ensure that this task is shared across teams. Finally, we also consider team size, as team size directly relates to the ability to take advantage of skill flexibility within teams. In larger teams, if surgical demand is below the expected level for one service, but above the expected level for another service within the same team, nurses assigned to the first service can be reassigned to the second service for the day, to ensure a sufficient number of nurses per OR for all services. However, to take advantage of such flexibility, teams need to have a sufficient number of services assigned to them (e.g., a team with one service has no opportunity for such flexibility). Through a case study we compare our optimized teams to current teams at our partner hospital, and show the potential for improvement in terms of the three factors above.

The second focus of Chapter III is to design shifts that correspond to the teams. Once a team structure is set, whether that be the current hospital teams or optimized teams, shifts need to be designed and allocated to services and teams to ensure that there is a sufficient number of nurses available for the surgeries that need to be performed on any given day. Nurses can be assigned to a service, in which case we call that nurse a *fixed nurse*, and this nurse will mostly work in their assigned service, unless they are reassigned to another service for a short period when surgical demand deviates from expected values. Nurses can also be assigned to a team, in which case we call that nurse a *float nurse*. Float nurses spend their time floating across services within their team, i.e., within their team they can be assigned to a different service every day.

To address the problem of shift design and allocation of surgical nurses, we formulated a MIP with decision variables for which shifts to staff from the allowable shifts used at the hospital, and the number of shifts needed to ensure sufficient number of nurses are available to assist with surgeries. The chosen shifts correspond to a weekly schedule (we assume shift schedules remain constant across many weeks unless significant fluctuations warrant the resolving of the model). The objective of the model is to balance the number of nurses per number of ORs across teams, and minimize the number of undesirable shifts. Nurse managers, the nurses responsible for assigning specific nurses to shifts, define undesirable shift characteristics in terms of duration, e.g., 12-hour shifts, or in terms of shift start or end time, e.g., shifts that end at 5 PM.

As in the surgery scheduling work of Chapter II, nurse shift design and allocation is also a computationally challenging problem to solve. For this reason we designed a decomposition heuristic, where each surgical service is considered separately, and

fixed nurses are assigned to each service. Once all services are assigned a sufficient number of fixed nurses, based on their expected demand, the remaining nurses are distributed across teams as float nurses. We show through a numerical study, that the decomposition heuristic performs sufficiently well compared to the optimal solution, and we also show that staffing schedules obtained through the decomposition heuristic outperform hospital staffing schedules. Furthermore, we demonstrate the large-scale problems, such as those encountered in practice, can be solved with reasonable computation times.

1.5 Chapter IV: Capacity Reservation Heuristics to Manage Access Delay in Operating Rooms

In the final technical chapter of this thesis, Chapter IV, we address strategic level decisions faced by hospitals: how they should allocate their OR capacity to meet uncertain demand for surgery and ensure high resource utilization, but also ensure patients are seen in a reasonable time. In Section 1.2, we described the block scheduling scheme that hospitals use, where a certain proportion of OR capacity is reserved for specific services and surgeons to schedule their surgeries in, and only the surgeons that have these reservations can schedule their surgeries in their block time. However, within the block time, there are often no guidelines on how patients should be assigned. For example, if a surgeon has block time on Monday and Tuesday, many hospitals do not prescribe which day an elective patient should be assigned to. Without clear guidelines, patient assignment to days is highly dependent on surgeon and patient preference. A natural intuition is to assign the patient in the example to the first available day, Monday. However, this intuition does not take into account other factors that affect surgery schedules. For example, the current utilization of the two days, and the expected number of urgent and emergent patients that can arise on

that day.

To create guidelines that would define what days to assign patients within the block time of surgeons, we turn to an idea sometimes used in outpatient clinic settings: surgery appointment slot reservations. We propose a heuristic in this vein, where a certain number of surgery slots ("slots" for short), are reserved for patients according to their attributes, and for the most part, patients are only able to use slots that correspond to their attributes. Patient attributes include surgical service (e.g., orthopedic, general), surgery type (e.g., short, long), and urgency level (e.g., emergent, urgent, elective). Moreover, each urgency level is assumed to have a surgery access target that defines maximum allowable waiting time for that level. The collection of the number of slots assigned to each service, type, and urgency level make up a *template* for a specific instance of the surgery slot reservation heuristic.

Consider an example of an orthopedic service with short surgery types, to see what this means in practice. In this setting a template could consist of 10 slots reserved for elective patients, 3 for urgent patients, and 3 for emergent patients, for example. This would mean that a total of 16 patients can be assigned to the orthopedic service with short surgery type on this specific day. However, this strict reservation policy might result in wasted capacity of ORs if demand for surgery is lower than anticipated. To avoid this, we also allow the releasing of unallocated elective reservations to urgent and emergent patients within their service, but across types.

We consider three heuristics that are based on the first come, first served (FCFS) principle. These heuristics serve as benchmarks that are in line with how some hospitals schedule surgeries in practice. The first such heuristic is the classical FCFS heuristic, where patients are assigned to the first day with sufficient capacity. The

second heuristic is a priority based FCFS heuristic, where a proportion of the block time is *carved out*, or reserved, for urgent and emergent patients, i.e., urgent and emergent patients are guaranteed that proportion of the block time to ensure short access to surgery. The third and final benchmark heuristic is a utilization based extension of the previous heuristic, where elective patients are assigned to days considering the current utilization of the day, i.e., how much of the available capacity has been allocated to patients.

Using a discrete event simulation model, we conducted two studies to compare the surgery slot reservation heuristic to the FCFS heuristics based on the following performance metrics: mean OR overtime and undertime, proportion of patients that exceeded their surgery access target, and mean patient wait time. In the first study, we created a stylized system with two identical services, while the second case study is based on hospital data. We show that there are template choices that result in good performance in both cases, and the surgery slot reservation heuristic tends to outperform the benchmarks when the system is highly utilized. The surgery slot reservation heuristic also has the additional benefit of knowing in advance the number and types of patients to expect, which helps hospitals in planning for supporting resources for surgery.

1.6 Chapter V: Conclusions and Future Research

The work presented in Chapters II-IV makes contributions to three important areas of surgery scheduling and planning that affect the following operational, tactical, and strategic decisions: surgery sequencing, service group team design and surgical nurse staffing, and OR capacity reservation. In Chapter V, we summarize some of our most important contributions. We also highlight areas of future research that could expand on this work.

CHAPTER II

Surgery Scheduling with Recovery Resources

2.1 Introduction

Achieving efficiency in surgery delivery systems is vital due to the fact that they greatly contribute to hospital costs and revenues. One of the challenges to achieving greater efficiency in elective surgery scheduling is that surgical cases that complete in an OR must quickly move to the recovery stage (i.e., the post-anesthesia care unit or PACU). Without effective planning and scheduling, the coupling of these stages can cause delays in the surgical schedule, overtime, and employee dissatisfaction. Inherent randomness in surgery and recovery durations makes scheduling challenging. Randomness in surgery durations occurs due to natural variation and unforeseen complications that can arise. Similarly, recovery duration is random, as patients can vary in their physiological response to the surgical procedure and anesthetic agents received. This chapter develops deterministic models; however, we discuss methods for making judicious choices of input parameters that can mitigate the impact of uncertainty, leading to an approach that we show is both tractable and effective in the stochastic setting.

There are several resource assignment challenges as well. In most cases, patient-surgeon assignments have to be respected and surgeons should perform all their surg-

eries consecutively to avoid large gaps in their schedule. Physical resources, such as PACU beds and ORs, can only be used by one patient at a time. Because the PACU is less expensive to operate, we focus on the key drivers of performance for the ORs, including minimizing overtime and surgeon elapsed time (the time between when the surgeon starts their first case and finishes their last case), which is equivalent to minimizing surgeon idle time.

This chapter emphasizes deterministic models; however, we discuss methods for making judicious choices of input parameters that can mitigate the impact of uncertainty, leading to an approach that we show is both tractable and effective in the stochastic setting. We propose fast heuristics that we show have attractive worst-case performance guarantees and average case performance. Moreover, we test the methods we propose using a discrete event simulation model based on data from a partner hospital.

2.2 Background and Literature Review

The scope of this chapter includes the main ORs of a hospital, and methods to generate elective surgery schedules for a single day. Once a patient and surgeon agree that surgery is necessary, the office of the surgeon typically calls a scheduling office to check for OR availability. Our partner hospital uses block scheduling, i.e., surgical services and surgeons have OR capacity reserved for them, and only they are allowed to schedule surgeries in their reserved time. In the problem we solve, these block scheduling rules are assumed to be in place and have already informed the list of surgeries to be performed by each surgeon. It is fairly common practice in hospitals to have ORs dedicated to emergent surgeries, and this is also the case at our partner hospital, therefore we only consider elective surgeries in this chapter.

Figure 2.1 shows the stages of the surgery delivery system at our partner hospital, and this system is common to many hospitals. First, on the day of surgery, if the patient has already been admitted to the hospital, they are transferred to the preoperative unit. If the patient is just arriving to the hospital, they have to go to a check-in area before they can go to the preoperative unit. In the preoperative unit they are seen by a nurse, an anesthesiologist, and their surgeon, each of whom confirms the procedure with the patient to avoid errors. When the patient, the surgical team, and the OR are all available and ready for surgery, the procedure can start. After surgery, most patients are transferred to the PACU to start recovery, if there is a bed available for them, and a nurse to monitor the recovery. Otherwise, the patient will start the recovery process in the OR causing delays in the consecutive cases scheduled in that OR, and potentially compromising patient safety. This phenomenon is called *OR boarding*. As this scenario is disadvantageous to all, the hospital tries very hard to avoid it, if possible. After recovery the patient can go to their desired ward, an alternate ward if the desired ward is full, or can be discharged.

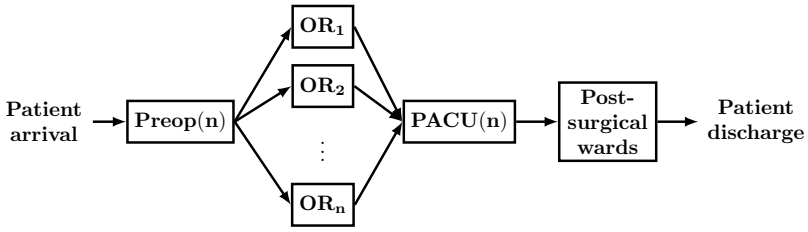


Figure 2.1: Stages of the surgery delivery system for elective surgeries with n preoperative bays, n ORs, and n PACU beds.

There is a substantial literature on surgery planning and scheduling. In our review, we focus on the most relevant literature that considers the PACU in addition to the ORs. For more general and comprehensive recent literature reviews see *Erdogan and Denton* (2010), *Guerriero and Guido* (2011), or *Cardoen et al.* (2010). Unlike the approach of this chapter, an alternate approach is to generate schedules considering the ORs only, and then study the effect of the schedule on the interaction between the ORs and the PACU. In this vein, *Marcon and Dexter* (2006) considered seven

sequencing rules and found the one that reduces the peak in the number of patients in the PACU. Using discrete event simulation they found that using simple sequencing rules hospitals can achieve significant reduction in the percentage of days with at least one PACU delay. *Saadouli et al.* (2015) used mathematical programming to decide which cases to perform, and in which ORs to perform the cases, but without accounting for PACU resources. They also used a discrete event simulation model to measure the impact of uncertainty on PACU resources.

Like this chapter, some authors have considered the PACU in the schedule generating phase. *Gul et al.* (2011) used a discrete event simulation for an outpatient procedure center to evaluate sequencing rules and methods to mitigate the effect of uncertainty with respect to the competing criteria of expected patient wait time and expected OR overtime, where they account for intake, preoperative care (or "preop" for short), surgery and recovery. Then they used a genetic algorithm to improve on the heuristic solutions. They assumed that a single surgeon has an OR for the entire day, an assumption that we relax to better model the behavior of many hospitals. We also allow for multiple surgeons in an OR with the constraint that each surgeon performs all their cases consecutively.

Jebali et al. (2006) proposed a two step method for daily OR scheduling. In step 1 they selected cases to perform from a wait list, and assigned them to ORs considering intensive care unit (ICU) bed availability and special OR equipment constraints, while minimizing the cost of keeping patients in the hospital waiting for surgery, the cost of OR overtime and OR undertime. In step 2 they sequenced the cases assigned to each OR with the possibility of reconsidering patient-OR assignments and also considering recovery constraints, while minimizing OR overtime. In this step they allowed for OR boarding. They considered surgeon availability, but consecutive surgeries for sur-

geons are not guaranteed, while our approach ensures consecutive surgeries for each surgeon. They used two disjoint mixed integer programs (MIPs) in the two steps, and assumed that all durations are deterministic. They found that their models work well on small examples with three ORs, four surgeons, four PACU beds, and 11-15 surgeries, however, unlike our chapter, they did not demonstrate their approach could scale to problems encountered by larger hospitals.

Fei et al. (2010) developed a two-stage heuristic approach, where in the first phase they assigned dates to surgeries using a column generation based heuristic to solve their set-partitioning IP model. They modeled the second phase as a flexible flow shop problem, where they assigned surgeries to ORs and sequenced them using a hybrid genetic algorithm. Their models respect patient-surgeon assignments, but unlike our chapter, a surgeon might not perform all their cases consecutively. They accounted for recovery time and allowed for OR boarding assuming deterministic surgery and recovery durations. Our approach yields an intuitive and computationally lightweight method.

Wang et al. (2014) considered a particle swarm optimization algorithm for the surgery scheduling problem with post-anesthesia resources. They formulated the problem as a deterministic MIP, and proposed a discrete particle swarm optimization algorithm combined with heuristic rules, where they found the number of ORs to open and the number of PACU beds needed. They found that their method performs well when compared to optimal solutions. However, they do not consider surgeon blocks or uncertainty. *Cardoen et al.* (2009a) used 6 objectives, including minimizing PACU overtime and the peak number of PACU beds used, to optimize case sequencing in an outpatient procedure center, but also considering factors like patient travel time to the procedure center and infection occurrence. They showed that the surgical case

sequencing optimization problem is NP-hard and developed optimization based exact and heuristic solution approaches for their formulated MIP. *Cardoen et al.* (2009b) elaborated on this approach by proposing an exact branch-and-price approach.

Augusto et al. (2010) investigated the logistical benefit of OR boarding when PACU workload is greater than OR workload. They considered surgery scheduling as a four stage deterministic flexible flow shop machine scheduling problem with the following stages: transfer from ward to OR, surgery and recovery, OR turnover, and finally transfer from OR to ward. They used a Lagrangian relaxation-based method to solve their deterministic mathematical program with the objective of minimizing the sum of a function of the surgery completion times. They showed that allowing recovery in the ORs can improve efficiency, which is intuitive. Their tested instances had 10-30 surgeries, 2-6 ORs, 1-4 PACU beds, and 1-2 transporter teams. Depending on the algorithm they used to build a feasible schedule, their worst-case duality gap in computational experiments was 16.5% or 31.25%. Our chapter indicates that even when PACU workload is lower on average than surgical workload in total for the day, poor sequencing can cause instances where the PACU is full and causes OR boarding. Our approach also provides insight into the problem, which we can claim due to the accuracy of our heuristic. Moreover, our experience in practice is that recovery in the OR as opposed to the PACU is strongly discouraged, and our approach seeks to avoid it.

2.2.1 Our Contributions to the Literature

This chapter makes new contributions to surgery scheduling arising from our collaboration with a mid-sized hospital. Despite a substantial literature, a number of open questions exist. Most of the existing literature relies on the use of complex models and methods (e.g., optimization, genetic algorithms, particle swarm algorithms,

and Lagrangian based methods) which are not accessible to most healthcare professionals at hospitals. The complexities of this problem also make it computationally infeasible to obtain optimal solutions for the large problem instances that are relevant to hospitals. As seen in the literature review, state-of-the-art approaches grapple with the size and complexity of the models. Our goal is to generate new models, algorithms, and insights for the purpose of improving surgery scheduling in hospitals. The approaches we propose are both intuitive and computationally tractable, and yield good performance when compared to optimization based solutions for small test instances, and when compared to current practice. This strongly suggests that the insights contributed in the reasoning behind the heuristic are sound and offer good intuition. We comprehensively address the relatively complex problem of scheduling surgeries for a single day under limited availability of ORs and PACU beds with a fast, easy to understand, and easy to implement 2-phase heuristic, supported by a combination of theoretical analysis of worst-case performance and computational analysis of average case performance.

2.2.2 Chapter Organization

The remainder of this chapter is organized as follows. To capture how shortages of one resource can affect the others, Section 2.3 presents a new MIP formulation for creating elective surgery schedules that consider resources directly supporting surgery (e.g., surgeon, OR), and also the limited availability of the PACU. This model uses deterministic surgery times and recovery times (both durations are surgeon and case specific) that are carefully selected as percentiles from the duration distributions to mitigate the impact of uncertainty in surgery and recovery durations to increase the reliability of the schedule. These durations, which we refer to as *hedged durations*, are determined through numerical experiments using a discrete event simulation detailed in Section 2.7.2. In our deterministic optimization, we ensure that there is no OR

boarding, and patient-surgeon assignments are respected. The objective is to minimize the fixed cost of opening the ORs, the variable cost of OR overtime, and the variable cost of surgeon elapsed time. In Section 2.4 we propose a fast 2-phase heuristic that exploits the problem structure, where the first phase finds the number of ORs to open and assigns surgeons to ORs, and the second phase sequences cases for each surgeon while considering the PACU. The heuristic is intuitive for healthcare professionals, and is easy to implement. Also in Section 2.4, we propose a decomposition heuristic for the MIP to be used as a benchmark for the 2-phase heuristic, since the overall problem is too computationally challenging to solve to optimality. In Section 2.5 we describe a discrete event simulation model that is used to evaluate the generated schedules under uncertainty. In Section 2.6 we provide worst-case performance guarantees for each of the phases of the 2-phase heuristic, and show that on average the heuristic solutions are very close to the optimal solutions. Section 2.7 presents case studies based on data from our partner hospital that use the simulation as a realistic model that incorporates stochasticity. We evaluate the heuristic schedules and the optimization based heuristic benchmark, and compare their cost to measure performance of the 2-phase heuristic in this more realistic setting.

2.3 Problem Formulation

A common approach for OR scheduling in the presence of uncertain surgery durations is to formulate the problem as a stochastic program (see, for example, *Denton et al.* (2010)). However, due to the addition of the PACU, which results in a large number of decision variables and multiple stages of decision making, this approach would not lead to a model that is solvable in a reasonable time. Indeed, as we show, even the deterministic problem is extremely difficult to solve for typical problem instances. Instead, we begin by formulating a deterministic MIP and then use a discrete event simulation model to evaluate schedules under uncertainty. Moreover, we com-

bine these models to investigate the ideal choice of model parameters in the MIP to mitigate the impact of uncertainty.

Our cost model is designed to match the reality of most ORs in hospitals in the United States and Canada. We assume the objective is to minimize the fixed cost of opening an OR for the day, the variable cost per unit time of OR overtime and the variable cost per unit time of surgeon elapsed time, while accounting for limited availability of ORs, surgeons, and PACU beds. At the surgical stage, we account for OR availability, and require that patient-surgeon assignments be respected, and that each surgeon performs all their cases consecutively. We also include constraints that ensure there is no OR boarding, i.e., recovery in the PACU starts right after surgery. At the recovery stage we assume limited PACU bed availability. Our focus is on the PACU, as opposed to the ICU, for example, because the vast majority of patients have to go to the PACU after surgery, and we are focusing on this majority of services; only a few surgery types require the patient to go to the ICU (e.g., cardiothoracic surgery), and bed availability is carefully managed to make certain a bed is available. Moreover, similar to surgery duration, recovery time in the PACU is on the order of hours, while length of stay in the ICU is on the order of days. Once a schedule is created, we use a discrete event simulation model to evaluate the schedule under uncertainty according to the same criteria as established for the MIP, where surgery durations and recovery durations are randomly generated according to probability distributions based on historical data.

Some hospitals, like our partner hospital, strategically invest in standardized, flexible OR suites to promote operational efficiency. In our MIP model we consider multiple services that do not have special equipment needs, and thus we assume that ORs are interchangeable and can be used by any service; however, the inclusion of

additional constraints for equipment or other requirements is straightforward. We also assume that the surgery duration includes turnover time, as this is the current practice at our partner hospital, where turnover time represents the time after each surgery that is needed to clean the OR, and potentially set up for the next surgery. Moreover, we assume that cancellations are not allowed, since cancellations the day before surgery are rare.

We begin by introducing a MIP model formulation for OR scheduling, which lays the foundations for incorporating PACU constraints into the model. Our formulation approach is to break up time into discrete time slots to easily track the whereabouts of patients and surgeons at any given slot. Thus, every time parameter is given in terms of numbers of time slots, with the horizon including the planned length of the day plus overtime for the day, if applicable. The smaller the length of the time slot, the more accurate the schedule is; however, small length also makes the model more computationally challenging. Therefore the length of a time slot is chosen to be large enough for computational tractability, but small enough to be consistent with hospital needs. In our case studies we used a time slot length of 15 minutes. Decision variables include the number of ORs to be opened, and assignment of surgeries to ORs and time slots to minimize total cost. The model also respects patient-surgeon assignments and makes sure that each surgeon performs all their surgeries one after the other to reflect block scheduling. Our notation is the following.

Indices:

- i index for surgeries (and thus for patients), $i = 1, \ldots, P$, with P being the number of patients to schedule.
- j index for ORs, $j = 1, \ldots, R$, with R being the number of ORs available.
- k index for surgeons, $k = 1, \ldots, K$, with K being the number of surgeons to operate.

t index for time slots, $t = 1, \ldots, T$, with T being the end of the time horizon.

Model parameters:

d_i duration for surgery i, including turnover time.

s_{ik} binary parameter representing if patient i is assigned to surgeon k.

S_j planned session length of OR j.

n number of time slots needed for turnover.

c^f fixed cost of opening an OR for a day.

c^v variable cost per time slot to keep OR j open past time S_j, (i.e., overtime).

c^s variable cost per time slot of surgeon elapsed time.

Decision variables:

x_j binary decision variable indicating whether OR j is opened ($x_j = 1$), or not ($x_j = 0$).

α_{ijt} binary decision variable indicating whether surgery i is allocated to OR j and starts in time slot t ($\alpha_{ijt} = 1$), or not ($\alpha_{ijt} = 0$).

q_{ijt} binary decision variable indicating whether patient i is in OR j in time slot t ($q_{ijt} = 1$), or not ($q_{ijt} = 0$).

u_{ikt} binary decision variable indicating if surgeon k operates on patient i in time slot t ($u_{ikt} = 1$), or not ($u_{ikt} = 0$).

o_j decision variable representing overtime for OR j.

Δ_k decision variable representing the last time slot surgeon k is operating.

δ_k decision variable used to calculate the first time slot surgeon k is operating with $T - \delta_k$ being the first time slot when surgeon k operates.

The following is the MIP formulation for the scheduling of ORs only:

$$\min \sum_{j=1}^{R} \left(c^f x_j + c^v o_j\right) + \sum_{k=1}^{K} c^s(\Delta_k - (T - \delta_k) + 1 - n) \tag{2.1}$$

$$\text{s.t.} \sum_{i=1}^{P} \alpha_{ijt} \leq x_j \qquad \forall j, t \tag{2.2}$$

$$\sum_{i=1}^{P}\sum_{j=1}^{R} q_{ijt} \leq \sum_{j=1}^{R} x_j \qquad \forall t \tag{2.3}$$

$$\sum_{j=1}^{R}\sum_{t=1}^{T} \alpha_{ijt} = 1 \qquad \forall i \tag{2.4}$$

$$\sum_{i=1}^{P} q_{ijt} \leq 1 \qquad \forall j, t \tag{2.5}$$

$$q_{ijt} \geq \alpha_{ijt} \qquad \forall i, j, t \tag{2.6}$$

$$\sum_{t'=t}^{t+d_i-1} q_{ijt'} \geq d_i \alpha_{ijt} \qquad \forall i, j, t = 1, \ldots, T - d_i + 1 \tag{2.7}$$

$$\sum_{j=1}^{R}\sum_{t=1}^{T} q_{ijt} = d_i \qquad \forall i \tag{2.8}$$

$$t q_{ijt} \leq S_j + o_j \qquad \forall i, j, t \tag{2.9}$$

$$\sum_{i=1}^{P} u_{ikt} \leq 1 \qquad \forall k, t \tag{2.10}$$

$$\sum_{t=1}^{T} u_{ikt} = d_i s_{ik} \qquad \forall i, k \tag{2.11}$$

$$\sum_{j=1}^{R} q_{ijt} = \sum_{k=1}^{K} u_{ikt} \qquad \forall i, t \tag{2.12}$$

$$\sum_{i=1}^{P} (T - t) u_{ikt} \leq \delta_k \qquad \forall k, t \tag{2.13}$$

$$\sum_{i=1}^{P} t u_{ikt} \leq \Delta_k \qquad \forall k, t \tag{2.14}$$

$$x_j, \alpha_{ijt}, q_{ijt}, u_{ikt} \in \{0, 1\}; o_j, \delta_k, \Delta_k \geq 0 \qquad \forall i, j, k, t. \tag{2.15}$$

The objective function (2.1) minimizes the fixed cost of opening the ORs, the variable cost per time slot of overtime of all ORs and the variable cost per time slot of surgeon elapsed time (including operating time and idle time, but not including the turnover time after the surgeon's last patient). Constraints (2.2) make sure that ORs are opened if they have patients assigned to them. Constraints (2.3) make sure that at any point in time the number of patients that are being operated on does not

exceed the number of ORs opened. Constraints (2.4) make sure that every patient starts surgery, thus no cancellations are allowed. Constraints (2.5) make sure that at most one patient can occupy an OR in any given time slot. Constraints (2.6) make sure that if a patient starts surgery in a time slot in an OR, the patient occupies that OR in that time slot. Constraints (2.7) make sure that the number of time slots allocated to each patient in the OR after they start surgery is at least the patient's surgery duration. Constraints (2.8) make sure that the number of time slots allocated to each patient in the OR equals the patient's surgery duration. Constraints (2.9) make sure that if a patient is in the OR after the planned session length of the OR, then overtime is used. Constraints (2.10) make sure that each surgeon can operate on at most one patient at any given time. Constraints (2.11) make sure that if a patient is assigned to a surgeon, then that surgeon operates on that patient for the required time, and if the patient is not assigned to that surgeon, then the surgeon does not operate on that patient. Constraints (2.12) make sure that a surgeon operates on the patient when the patient is in the OR. Constraints (2.13)-(2.14) are used to calculate the first and last time slots a surgeon is busy.

To speed up computational time, we can add the following inequalities to fix α_{ijt} variables based on the fact that surgery has to start in time to finish the procedure before the end of the time horizon:

$$\sum_{j=1}^{R} \sum_{t=T-d_i+1}^{T} \alpha_{ijt} = 0 \qquad \forall i. \tag{2.16}$$

We also add additional constraints to eliminate symmetry in the problem, e.g., to make sure ORs are opened in order (*Denton et al.* (2010)).

Next we build on the above model to develop our comprehensive deterministic model, which we call MIP[OR,PACU], to solve the problem of allocating surgeries to ORs, given limited PACU capacity. This formulation augments formulation (2.1)-

(2.15) with additional decision variables and constraints, that ensure that a surgery is only started if there will be a PACU bed available for the patient. Note, that unlike at the OR stage, where patients are assigned to specific ORs, in the PACU they are not assigned to specific beds, as is typically the case in practice. MIP[OR,PACU] focuses on the OR costs, and the prevention of OR boarding, because they outweigh the costs of the PACU. The following is a list of new parameters and decision variables.

Parameters:

r_i recovery time of patient i.

B number of available beds in the PACU.

Decision variables:

β_{it} binary decision variable representing whether patient i starts recovery in time slot t ($\beta_{it} = 1$), or not ($\beta_{it} = 0$).

z_{it} binary decision variable representing whether patient i is in the PACU in time slot t ($z_{it} = 1$), or not ($z_{it} = 0$).

MIP[OR,PACU]: OR and PACU Scheduling Model

$$\min \sum_{j=1}^{R} \left(c^f x_j + c^v o_j\right) + \sum_{k=1}^{K} c^s(\Delta_k - (T - \delta_k) + 1 - n) \tag{2.17}$$

s.t. Constraints (2.2)-(2.14)

$$\beta_{i,t+d_i-n} \leq \sum_{j=1}^{R} \alpha_{ijt} \qquad \forall i, t = 1, \ldots, T - d_i \tag{2.18}$$

$$\sum_{t=1}^{T} \beta_{it} = 1 \qquad \forall i \tag{2.19}$$

$$z_{it} \geq \beta_{it} \qquad \forall i, t \tag{2.20}$$

$$\sum_{t'=t}^{t+r_i-1} z_{it'} \geq r_i \, \beta_{it} \qquad \forall i, t = 1, \ldots, T - r_i + 1 \tag{2.21}$$

$$\sum_{t=1}^{T} z_{it} = r_i \qquad \forall i \tag{2.22}$$

$$\sum_{i=1}^{P} z_{it} \leq B \qquad \forall t \tag{2.23}$$

$$x_j, \alpha_{ijt}, q_{ijt}, u_{ikt}, \beta_{it}, z_{it} \in \{0,1\}; o_j, \delta_k, \Delta_k \geq 0 \qquad \forall i,j,k,t. \tag{2.24}$$

The objective function, equation (2.17), includes as before, the fixed cost of opening the ORs, the variable cost per time slot of OR overtime and the variable cost per time slot of surgeon elapsed time. Constraints (2.18) make sure that recovery can only start in the time slot immediately following surgery. Note that turnover has to be subtracted from surgery duration, since by definition it includes turnover time. Constraints (2.19) make sure that recovery starts exactly once. Constraints (2.20) make sure that if the patient starts recovery in a time slot, then the patient is in the PACU. Constraints (2.21) make sure that the number of time slots allocated to each patient in the PACU after they start recovery is at least the patient's recovery duration. Constraints (2.22) make sure that the number of time slots allocated to each patient in the PACU equals the patient's recovery duration. Constraints (2.23) make sure that the number of patients in the PACU in any given time slot does not exceed the number of beds available.

Note that the objective function and the constraints in this model strive to achieve high utilization; therefore overtime and OR boarding are not counted. To accomplish this, the model picks the number of ORs to open, sets surgeon-to-OR assignments, and sequences patients to avoid OR boarding while minimizing OR idling.

As before, we can add additional constraints to fix α_{ijt} variables, since surgery has to start in time to finish both surgery and recovery before the end of the time horizon. Note that recovery starts parallel to the turnover of the OR, so $r_i + d_i - n$ is the total time that each patient needs to finish both surgery and recovery. Moreover, we can also add constraints to fix β_{it} variables, since we know that recovery cannot

start at the beginning of the time horizon, when surgery could not have finished yet, i.e., the earliest recovery can start is in time slot $d_i - n + 1$.

2.4 Solution Methods

In this section we focus on solution methods for MIP[OR,PACU]. Due to the computationally challenging nature of the problem, we develop a very fast and intuitive 2-phase heuristic, that exploits the problem structure. In the first phase we find the surgeon-to-OR assignments. Note that this also means finding the number of ORs to open. Considering these decisions fixed, sequencing decisions are made in the second phase. Since we cannot compute the optimal solutions to realistic problems due to the computational challenges, we evaluate the performance of the 2-phase heuristic as follows. We propose a decomposition heuristic in Section 2.4.2 that, similar to the 2-phase heuristic, separates the decisions about the number of ORs to open and surgeon-to-OR assignments in a preprocessing step and fixes them before the overall problem with sequencing decisions is solved in the second step. Although this decomposition heuristic does not guarantee optimal solutions, we show that it provides good error bounds; thus, it serves as a benchmark for measuring performance of the 2-phase heuristic. In Section 2.7 we compare the approaches on the basis of computational time and solution quality.

2.4.1 Fast 2-Phase Heuristic

First, we introduce the very intuitive and easy-to-implement 2-phase heuristic for the surgery scheduling problem. We explain each of the two phases of the heuristic in this section.

2.4.1.1 Phase 1: Surgeon-to-OR Assignment Heuristic

In this phase, we first fix the number of ORs, and assign surgeons to ORs using the longest processing time first (LPT) algorithm, then using this method, we find the ideal number of ORs to open through exhaustive search. Some have considered

this problem in the on-line setting, where decisions are made without knowing the durations distributions, for example *Berg and Denton* (2017); however, we consider a different context in which the surgeries to be scheduled are known, and duration distribution information can be used in the scheduling process. To our knowledge, we are the first to prove the result we present for the LPT algorithm, which is an extension to the results of *Dell'Olmo et al.* (1998) where they do not distinguish between cost of regular time and overtime.

Consider each surgeon's block (i.e., all their surgeries they perform for the day) and order the blocks in decreasing order according to their total surgical time duration (including turnover). Given a fixed number of ORs, we take the ordered list of surgery blocks and then perform the assignment of surgeons to ORs by always selecting next the OR with the most available time, breaking ties arbitrarily. When the planned session length is the same for all ORs, this is equivalent to choosing the least utilized OR. (Note that this does not consider the PACU at all; rather, that will be considered in the second phase.) This problem is exactly the extensible bin packing problem, where ORs are the bins, surgeon blocks are the items, and OR overtime means extending the bins. The version of the problem, where surgeon blocks of size no greater than $S/3$ can be preempted is called the semi-preemptive version. Let C^H be the cost of the heuristic solution to the surgeon-to-OR assignment problem, and C^* be the cost of the optimal semi-preemptive solution for the same instance. By extending the results of *Dell'Olmo et al.* (1998) we prove that LPT has the following worst-case performance bound when the number of ORs is fixed.

Theorem 2.1. *For any instance, where the planned session length of each OR is S, we have*

$$\frac{C^H}{C^*} \leq 1 + \frac{Sc^v}{12c^f},$$

where an instance is defined by the list of surgeon blocks and the number of ORs available. Moreover, there exist instances for which this bound is tight.

The proofs and definitions, which are presented in Appendix 2.9.1, closely parallel the proofs in *Dell'Olmo et al.* (1998) and extend them to the case of arbitrary costs c^f and c^v, and planned session length S.

To complete phase 1, we employ exhaustive search in R, i.e., we perform the heuristic with varying the number of ORs available, to easily find the solution with minimal cost, which will also possess the above shown worst-case performance guarantee. Note, that Theorem 2.1 is valid under the assumption that the planned session length of each OR is S, but the approach can be applied to the more general case where the planned session length differs by OR.

Observe, that based on the block scheduling rules in place, the list of surgeries to be performed that feeds into our algorithm is already consistent with the block schedule. We do allow two surgeons from different services to use the same room on the same day. If this is not acceptable in certain hospital contexts, one can restrict attention to each service to enforce the constraint.

2.4.1.2 Phase 2: Sequencing Heuristic

LPT assigns surgeon blocks to ORs, which only requires the total duration of a surgeon block (i.e., the sum of the durations of all surgeries of a surgeon) while recovery information is disregarded. The LPT heuristic is insensitive to the sequence of surgeries within a surgeon's block; any sequence of surgeries will give the same block duration when recovery is ignored. However, the question of sequencing surgeries within a block given limited PACU capacity still remains. This problem is similar to the scheduling problem $F2|block|C_{\max}$, which is a two machine flow shop problem with blocking (i.e., if there is OR boarding, the patient's surgery will be delayed until such time that a PACU bed is available at the end of surgery), where the objective is to minimize overall makespan. However, in our setting the goal is to minimize makespan with respect to the first stage, the ORs (which also minimizes OR overtime). This

goal is justified by the much lower cost of operating the PACU and the objectives of a typical hospital practice. Moreover, if OR boarding occurs (which we allow in the simulation model which has random durations), that means that a job spends some of its machine 2 processing time on machine 1 (i.e., recovering in the OR), and will have a correspondingly smaller processing time on machine 2 as a result. Thus this problem is different from the machine scheduling context. We propose a heuristic for sequencing patients within a single surgeon's block. OR overtime is a non-decreasing function of surgeon elapsed time, thus through minimizing surgeon elapsed time we also minimize OR overtime. Moreover, surgeons also like to avoid the potential idle time induced by patient recovery in the OR. Therefore the objective of the heuristic is to minimize surgeon elapsed time. The heuristic tries to match recovery time of the patient currently in the OR, to the next patient's surgery time to avoid OR idling due to a PACU bed being unavailable and thus minimize surgeon elapsed time and OR overtime.

Let W be a $P \times P$ matrix, with $W_{ij} = r_i - d_j$ for $i \neq j$, and $W_{ii} = \infty$. Let $W^j = \min_i W_{ij}\ \forall j$, and let $p^* = \text{argmax}_j\, W^j$ be the first patient in the sequence. Then the heuristic follows.

for *($a = 1, \ldots, P-2$)* **do**
  if $\min_j W_{p^*j} > 0$ **then**
    $p^*_{\text{new}} = \text{argmin}_j\, W_{p^*j}$
  else
    $p^*_{\text{new}} = \underset{j:W_{p^*j}\leq 0}{\text{argmax}}\, W_{p^*j}$
  end
  add p^*_{new} to the end of the sequence and exclude this patient from further consideration.
  $p^* = p^*_{\text{new}}$
end

The basic idea of the heuristic is to pick the first patient to be the patient that

has the most potential to cause idling if they were to follow another patient. Then always pick the next patient to be a patient whose surgery duration is closest to the current patient's recovery duration. Once the sequence is set, we assign start times to patients, inserting idle time into the OR schedule to avoid OR boarding. Note that as before, recovery and turnover are parallel events. We refer to this as the *difference heuristic* (DH).

We have the following performance bound for the difference heuristic.

Theorem 2.2. *In the difference heuristic setting, where $W_{ij} = r_i - d_j$ for $i \neq j$ and $W_{ii} = \infty$, let*

$$W^i = \max_{j:j\neq i}(W_{ij})^+; \quad \bar{W}^i = \min_i W^i; \quad w^i = \min_{j:j\neq i}(W_{ij})^+; \quad \bar{w}^i = \max_i w^i \quad \forall i,$$

$$W^j = \max_{i:i\neq j}(W_{ij})^+; \quad \bar{W}^j = \min_j W^j; \quad w^j = \min_{i:i\neq j}(W_{ij})^+; \quad \bar{w}^j = \max_j w^j \quad \forall j.$$

Then for any instance we have

$$C^{DH} - C_1^* \leq c^s \cdot \min\left\{ \sum_{i=1}^{P} W^i - \bar{W}^i - \left(\sum_{i=1}^{P} w^i - \bar{w}^i \right), \right. \\ \left. \sum_{j=1}^{P} W^j - \bar{W}^j - \left(\sum_{j=1}^{P} w^j - \bar{w}^j \right) \right\},$$

where C^{DH} is the cost of the schedule given by the difference heuristic, and C_1^* is the cost of the optimal solution. Moreover, there exist instances for which this bound is tight.

It can also be shown that the difference heuristic is optimal in the following case that often happens in practice with long procedures.

Theorem 2.3. *For any instance with a single surgeon, the difference heuristic results in an optimal sequence if the number of cases assigned to the surgeon is two.*

For proofs of these theorems, please refer to Appendix 2.9.2. Note that the idea behind Theorem 2.3 also applies for sequencing two surgeons in the same OR. To see this, considering allowing for each surgeon to have an arbitrary number of patients, and fix the surgery sequence of each surgeon. By associating each surgeon with the surgery duration of their first patient and the recovery duration of their last patient, the argument proving Theorem 2.3 also applies to this problem: the difference heuristic will find the optimal sequence of the two surgeons. From this we can further observe that if two surgeons share an OR with one associated PACU bed, each has at most two surgeries, and if one surgeon follows the other, then the difference heuristic will find an optimal sequence for each surgeon and also an optimal ordering of the surgeons, conditional on the sequence of surgeries for the two surgeons being fixed first.

In some hospitals multiple surgeons may use an OR on a given day. In such cases, once the sequence within each surgeon's block is decided, if for each surgeon block we consider the first patient's surgery duration and the last patient's recovery duration, we can again use the difference heuristic to sequence surgeons that are assigned to the same OR. In the sequel, when referring to the difference heuristic, we mean sequencing patients within each surgeon's block, and then sequencing surgeons that are assigned to the same OR.

2.4.2 MIP Decomposition Heuristic

To evaluate the performance of the 2-phase heuristic, we propose the following decomposition heuristic as a benchmark, which also has two parts, which we will call *steps* to avoid confusion with the phases defined in Section 2.4.1. In step 1 we use a MIP to assign surgeons to ORs in the absence of PACU constraints, then in step 2 we fix the surgeon-to-OR assignments in the MIP[OR, PACU] and sequence surgeries using the restricted instance of MIP[OR, PACU].

We presented a formulation for the OR scheduling problem that assigns surgeons to ORs in Section 2.3. To lay the foundation for incorporating PACU constraints into the model, that formulation was more complex due to accounting for discrete time slots. However, the OR scheduling problem, which is the same as the extensible bin packing problem, can be formulated in a simpler way that we present now. We refer to the following model as MIP[OR] for short. Let $\theta_{jk} = 1$ if surgeon k is assigned to OR j, and $\theta_{jk} = 0$ otherwise. Using the same notation as defined before, the following is the MIP[OR]:

$$\min \sum_{j=1}^{R} \left(c^f x_j + c^v o_j \right) \tag{2.25}$$

$$\text{s.t.} \sum_{k=1}^{K} \left(\theta_{jk} \sum_{i=1}^{P} d_i s_{ik} \right) \leq S_j x_j + o_j \qquad \forall j \tag{2.26}$$

$$\sum_{j=1}^{R} \theta_{jk} = 1 \qquad \forall k \tag{2.27}$$

$$\theta_{jk}, x_j \in \{0,1\}; o_j \geq 0 \qquad \forall j,k. \tag{2.28}$$

The objective function (2.25) minimizes the fixed cost of opening the ORs and the variable cost of OR overtime. Constraints (2.26) make sure that if a surgeon is assigned to an OR it will be open and that overtime is used if necessary. Constraints (2.27) make sure that each surgeon is assigned to exactly one OR. Moreover, symmetry eliminating constraints can be added as before.

Solving MIP[OR] in the first step of the decomposition heuristic generates the surgeon-to-OR assignments. To enforce these surgeon-to-OR assignments in the complete model, we add the following constraint to MIP[OR,PACU]:

$$\sum_{t=1}^{T} q_{ijt} \geq s_{ik} \theta_{jk} \qquad \forall i,j,k. \tag{2.29}$$

Since surgeons are preassigned to ORs, only one patient is allowed to be in an OR

at any given time, and surgeon elapsed time is minimized, there is no need for the variables u_{ikt}, and we can replace constraints (2.10)-(2.14) in MIP[OR,PACU] by the following constraints to reduce the number of decision variables:

$$\sum_{i=1}^{P} t q_{ijt} s_{ik} \leq \Delta_k \qquad \forall j, k, t \tag{2.30}$$

$$\sum_{i=1}^{P} (T-t) q_{ijt} s_{ik} \leq \delta_k \qquad \forall j, k, t. \tag{2.31}$$

This decomposition is not guaranteed to find the overall optimal solution to the problem; however, the following is a lower bound on the overall optimal solution:

$$c^f \sum_{j=1}^{R} x_j^* + c^v \sum_{j=1}^{R} o_j^* + c^s \sum_{i=1}^{P} d_i,$$

where x_j^* and o_j^* is the optimal solution to MIP[OR] for all j. Thus the first two terms represent the fixed cost of opening the ORs and the variable cost of OR overtime when the PACU is ignored. The last term is a lower bound on surgeon elapsed time, and can be calculated from the data. This is a lower bound, since the MIP[OR] is a relaxation of the overall problem with the assumption that the PACU has infinite capacity. We provide some insight into the performance of the decomposition heuristic in Section 2.7.3.

2.5 Simulation Model

Since the previous models assume deterministic surgery and recovery durations, the question arises how the resulting schedules would perform under uncertainty. To account for the stochastic nature of surgery and recovery durations, we have developed a discrete event simulation model to evaluate the daily schedules generated by the decomposition heuristic and the 2-phase heuristic. Figure 2.2 shows the steps of generating and evaluating a schedule. To generate a schedule using the 2-phase heuristic, first we use LPT to get surgeon-to-OR assignments, second the difference heuristic to

sequence patients within a surgeon's block and surgeons that are assigned to the same OR. In the decomposition heuristic setting we first use the MIP[OR] from Section 2.4.2 to get surgeon-to-OR assignments, then use the restricted MIP[OR,PACU] from Section 2.3 to sequence surgeries. Once a schedule is generated, we evaluate it with the discrete event simulation model to find the expected cost of the schedule.

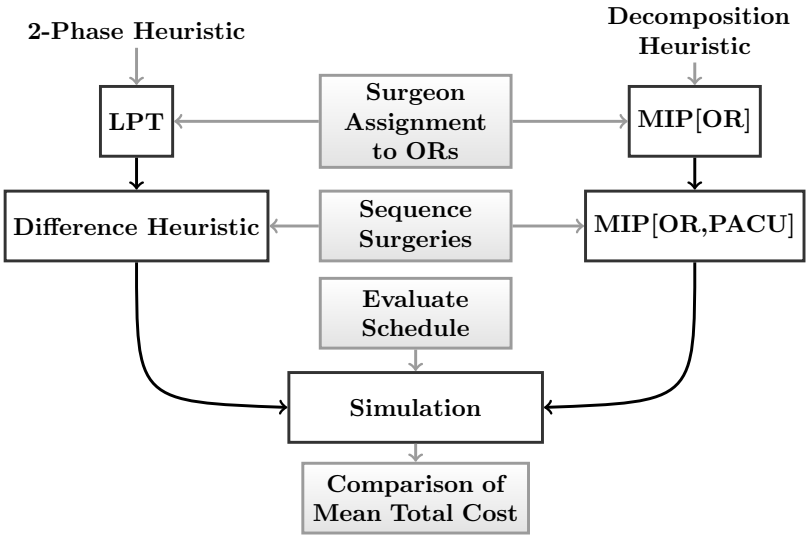


Figure 2.2: The process of schedule generation and evaluation using two two-stage heuristics: the 2-phase heuristic and the decomposition heuristic.

Inputs to the discrete event simulation model include the number of ORs available, the number of PACU beds available, patient-surgeon assignments, surgery start times, surgery and recovery duration distributions, turnover duration, the fixed cost of opening an OR, the variable cost of OR overtime, and the variable cost of surgeon elapsed time. The planned session length of each OR is 8 hours, which is consistent in both heuristics. For both surgery and recovery durations, we assumed lognormal distributions (*May et al.* (2000); *Zhou and Dexter* (1998)). If enough data was available, we considered surgeon and case specific surgery and recovery durations. However, some surgeries are performed often by a surgeon, while others are not. Due to this, not all surgeon-case pairs have enough data points to obtain a distribution to find percentiles. To overcome this challenge, for each surgeon-case pair that did

not have at least 10 samples we used the overall mean and variance for all surgeon samples for the case type.

Patients move to the OR after their surgery start time as soon as their surgeon and an OR is available. A random surgery duration for the patient is generated from the surgery duration distribution based on historical data. Once the surgery is over, the patient moves to the PACU if there is a bed available. Otherwise the patient boards in the OR until a bed becomes available, or their recovery duration is up, which is generated from the recovery duration distribution, based on historical data. As soon as the patient leaves the OR, a 30 minute turnover time starts, after which the OR is ready for the next patient.

Simulation evaluation criteria included cost as defined before: cost of opening the ORs, OR overtime, and surgeon elapsed time. Moreover, in the deterministic setting we make sure that OR boarding does not occur. In the simulation, however, OR boarding can happen if recovery takes longer than expected and there are no beds available in the PACU. This is an additional performance metric measured in the simulation model.

2.6 Numerical Results

The worst-case performance of each phase of the 2-phase heuristic provides an upper bound on the error across all possible model instances; however, the average performance is also a critical metric, because it more closely reflects what can be expected in practice. We demonstrate the performance of the combined phases through a case study in the next section. In this section, for a set of random test cases we compare the numerical performance of the phases of the 2-phase heuristic: LPT and the difference heuristic.

2.6.1 Surgeon-to-OR Assignment: LPT Heuristic

In order to estimate the average performance of phase 1 of the 2-phase heuristic, we tested LPT on 270 randomly generated instances where surgeon block durations were independent and identically distributed uniform random variables between 0 and 1, and an OR day is one unit ($S = 1$). Instances were defined in terms of the number of surgeon blocks and the variable cost of OR overtime, c^v; the fixed cost of opening an OR, c^f, was 1 for all cases, without loss of generality. Each instance was tested on 30 replications. The number of surgeon blocks considered was 10, 15 and 20 and the values considered for c^v were 2, 4, and 8. The choice of $c^v/c^f = 4$ is intended to be representative of a hospital setting with the additional values of 2 and 8 selected. The performance was calculated using the following formula for the optimality gap:

$$\frac{C^{LPT} - C^*_N}{C^*_N} \cdot 100\%,$$

where C^*_N is the optimal solution of the non-preemptive problem.

Overall, the average gap was 0.42%, the worst-case gap was 6.99%, and the optimal solution was found 77.41% of the time. The heuristic is most prone to error when the mean surgeon block duration is around half of the OR day. This is intuitive, since as surgeon block durations tend to zero or to the OR day duration, the heuristic is expected to have zero error (e.g. durations close to zero approach a continuous relaxation, while surgeon block durations close to the OR day duration means there are no alternative arrangements of surgeon blocks within ORs). Moreover, the largest error is associated with the largest ratio of variable cost of OR overtime to fixed cost of opening an OR, which is also intuitive, as there is a high penalty for errors in such cases. Our conclusions hold across the different numbers of surgeon blocks considered.

2.6.2 Surgery Sequencing: Difference Heuristic

In order to estimate the average performance of phase 2 of the 2-phase heuristic, we conducted a numerical analysis for the general, orthopedic and urology surgery

services, that are common to most hospitals. To generate test instances, we randomly sampled days from our data set when surgeries in these specialties were performed. To match the heuristic's setup, days were only considered if each surgeon performed all their cases in the same OR. On the days selected, each OR was considered separately. Each day we took all surgeons and surgeries performed in the same OR and sequenced them using the difference heuristic (sequenced surgeries within each surgeon's block and then sequenced surgeons in the OR) with one PACU bed available. We considered 270 single OR, single PACU bed instances. Then we used the MIP to obtain the optimal solution, and compared the two schedules based on surgeon elapsed time, since in these environments minimizing surgeon elapsed time also minimizes OR overtime. The optimality gap was calculated based on the following formula:

$$\frac{C^{DH} - C_1^*}{C_1^*} \cdot 100\%$$

Overall, the average gap was 0.70%, the worst-case gap was 30.30%, and the optimal solution was found 95.19% of the time. The heuristic tends to perform poorly when the mean recovery duration exceeds mean surgery duration. This is intuitive, since recovery duration tends to have less effect on sequencing decisions when surgeries are long and recovery durations are short.

2.7 Case Study

In this section we present a case study to demonstrate how our algorithms can be used to generate schedules that work well under uncertainty.

2.7.1 Case Study Description

The data we used was provided by our partner hospital, a medium sized teaching hospital. The extensive data set includes information over a span of 14 months about arrival and departure times in the ORs and the PACU, and procedure and surgeon information.

To test our proposed heuristics, we selected three services (orthopedic, general, and urology), that are common to most hospitals. This provided large enough instances for our results to be relevant, and small enough instances to be able to get solutions using the decomposition heuristic. We randomly sampled the data set to capture days that had orthopedic, general, and urology surgeries and there were between 15 to 20 patients of these types of surgeries. On each day there were up to 15 ORs available to open. We compared the two heuristics (2-phase and decomposition) for each instance using the mean cost given by the simulation, which includes the fixed cost of opening the ORs, the variable cost of OR overtime, and the variable cost of surgeon elapsed time.

Based on the assessment of the importance of criteria for the hospital, the following parameters were used. We set $c^f = 20$ and $c^v = 4$, so that about 1.5 hours of overtime would be equivalent to opening a new OR. Moreover, $c^s = 1$ to ensure surgeon waiting is minimized and that each surgeon performs all their cases consecutively. Our time slot length was 15 minutes and OR turnover time was set to 30 minutes. The former was chosen because it provides suitably detailed resolution of surgery schedules and the latter was based on expert opinion at our partner hospital.

2.7.2 Surgery and Recovery Duration Hedging

It is well known in OR scheduling practice that using the mean surgical durations leads to increasing delays as the day progresses. In the authors' experience, hospitals sometimes use the mean or median durations, but often try to hedge against uncertainty by using percentiles from the duration distribution that range between the 60th and 80th percentiles. Planning for cases to take longer than the median helps create more reliable schedules.

Our models require deterministic data input, however, surgery and recovery du-

rations are stochastic. Therefore we need a way to estimate these durations that will result in highly reliable schedules. To achieve this, we performed experiments in which schedules based on various percentile combinations were evaluated with the simulation model. From this we selected a percentile from the surgery and recovery duration distributions to be used as deterministic data inputs. As before, surgery and recovery distributions were surgeon and case specific, if enough data was available, and we assumed a lognormal distribution to find the desired percentile (*May et al.* (2000), *Zhou and Dexter* (1998)).

To determine the best percentile given our system parameters, our approach was to randomly sample days for the practices considered (general, orthopedic and urology) to create a set of test instances. Due to the long tail on surgery and recovery durations, duration mean tends to be significantly higher than the median (typically the mean is closer to the 60th percentile than to the 50th). Durations below the mean are not expected to have good performance because of the very high probability of delays. Therefore we evaluated the 60th, 70th, and 80th percentiles for surgery and recovery durations. For each test instance we used the decomposition heuristic to obtain a schedule using all 9 combinations of percentiles and evaluated the schedule with the simulation model. The large number of runs for each instance and computational challenges limited the size of the test suite. Figure 2.3 shows the cost for 12 instances considered with 18 patients and 8 surgeons on average, as determined by the simulation. Mean simulation costs were calculated with a 95% confidence interval, and the half width of the confidence intervals was less then 0.2% in all instances, indicating high precision. The variation between percentiles for each instance was not large, indicating relative insensitivity due to the fact that the schedules were optimized. In our notation (60,80) means that surgery was considered at the 60th percentile and recovery was considered at the 80th percentile, for example. We calculated how many times each percentile combination achieved the minimum considering all instances. The pairs (60,70) and (60,80) each achieved the minimum in 4 instances, and the average total cost of (60,70) was also less than that of (60,80), so we used (60,70) in

our case study in Section 2.7.3.

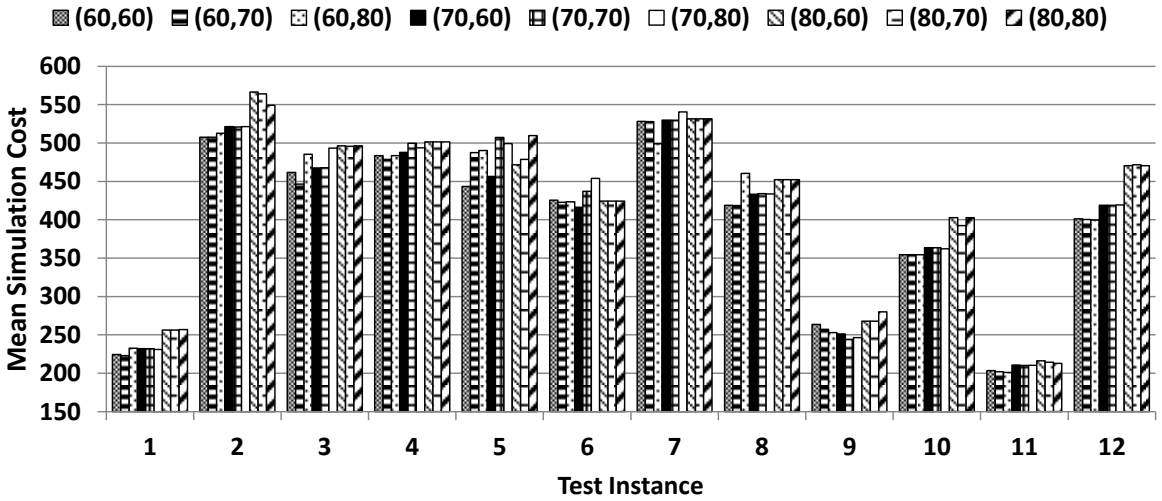


Figure 2.3: Hedging analysis of randomly sampled days with surgeon and case specific surgery and recovery durations under the decomposition heuristic. Nine pairs of surgery and recovery percentiles are compared for each test instance.

We show that modeling the PACU can reduce overtime costs significantly in the following analysis. As the benchmark for schedules that do not attempt to optimize sequencing, we used phase 1 of the 2-phase heuristic, i.e., LPT, to assign surgeons to ORs in a near-optimal manner, and then used a random sequence of surgeon blocks in ORs and a random sequence of surgeries within each surgeon's block. Random sequences were used as the benchmark since there were no discernible patterns based on historical data, and this way the comparison is based on the importance of sequencing, as opposed to surgeon-to-OR assignments. We compared overtime for the optimized and randomized schedules, which are affected by every aspect of the problem (number of ORs opened, case sequencing, surgeon sequencing and OR idling to avoid OR boarding). When we use the (60,70) combination for decomposition, we see that the mean overtime cost for the 12 instances was 88.6 with a standard deviation of 59.8. Using LPT and random sequence with the (60,60) combination, which again was picked by calculating how many times each percentile combination achieved the minimum cost considering all instances, the mean overtime cost was 100.6, with a standard deviation of 55.5. Although the standard deviation was similar, there was

a 12% reduction in mean overtime cost, so we observe that considerable improvements are possible when the limited availability of the PACU is considered through sequencing.

2.7.3 2-Phase Heuristic Performance Case Study Results

We considered 43 randomly sampled days. Statistical information about the data considered and computation times are given in Table 2.1. Observe the dramatic reduction in processing time for the 2-phase heuristic. On average, the decomposition heuristic took $3 \cdot 10^6$ times as much CPU time.

	Minimum	Average	Maximum
Surgery duration (min)	60	166	375
Recovery duration (min)	75	133	210
Number of ORs used	4	6	7
Number of patients	15	18	20
Number of surgeons	6	8	11
2-phase Heuristic CPU time (seconds)	0.000	0.005	0.016
Decomposition Heuristic CPU time (seconds)	149	14954	123520

Table 2.1: Statistics about the hospital data and computational time for the 43 days considered for the case study.

Figure 2.4 shows the mean simulation costs associated with the schedules generated for the 43 instances. As before, schedule cost is the sum of the fixed cost of opening the ORs, the variable cost of OR overtime, and the variable cost of surgeon elapsed time. The figure shows the mean cost obtained from the simulation associated with schedules generated with the 2-phase heuristic and with the decomposition heuristic. Mean simulation costs were calculated with a 95% confidence interval, and the half width of the confidence intervals was less than 1.2% in all instances, indicating high precision. We can see from the figure that the 2-phase heuristic performed well when compared to the decomposition heuristic, sometimes even beating the de-

composition heuristic in part due to the stochastic performance analysis.

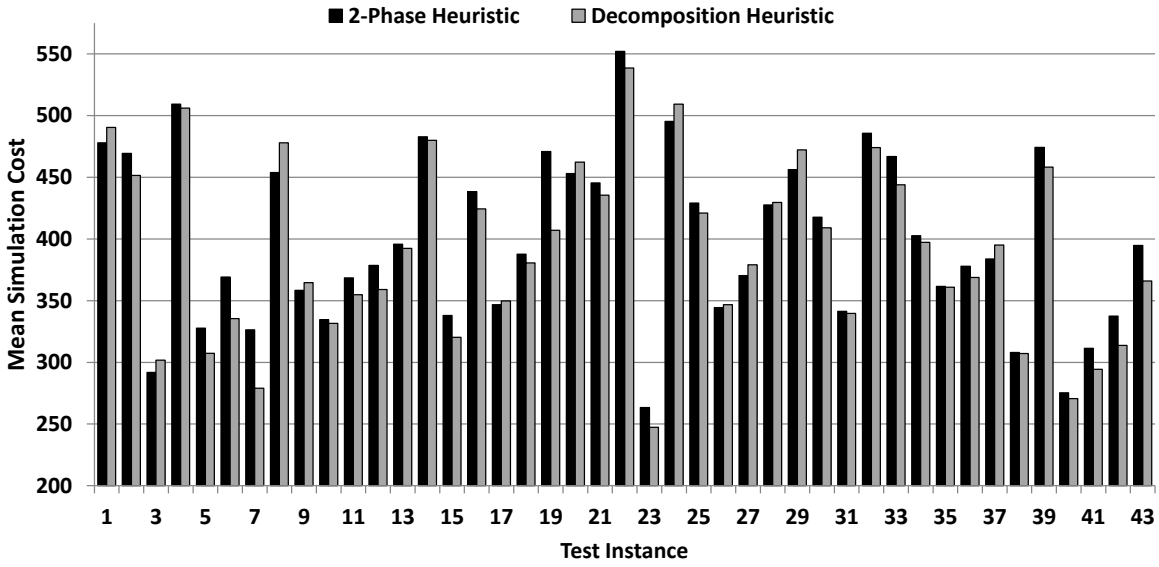


Figure 2.4: Simulation cost comparison between the decomposition and the 2-phase heuristic. The results are equally good when the cost of OR boarding is considered at the same rate as OR overtime cost.

Our computational experiments indicated that MIP[OR,PACU] cannot be solved for all instances in a reasonable time. Therefore in the deterministic setting we compared solutions to the lower bound derived from the decomposition heuristic in Section 2.4.2 to evaluate how often the heuristics found the optimal solution to the overall problem. The 2-phase heuristic found a solution with an objective function value of the lower bound in 26% of the instances, and on average the solutions were 6% away from the lower bound with a maximum of 27%. The decomposition heuristic found a solution with the objective function value equal to the lower bound in 37 out of the 43 cases (86% of the time), and on average the solutions were 0.7% away from the lower bound with a maximum deviation of 9%. These results indicate that the 2-phase heuristic is likely to be very good, thus the additional advantage of using the computationally challenging optimization models is limited.

Overall, solutions generated by the 2-phase heuristic were within 10% of the decomposition heuristic solutions in 93% of the instances considered, and within 5% in

74% of the instances considered, when evaluated using the simulation model. The average difference between the cost achieved by the 2-phase heuristic relative to the decomposition heuristic was 2.38% with a standard deviation of 4.6.

In addition to minimizing cost, our goal is to generate schedules with minimal OR boarding. In the schedules obtained through the 2-phase heuristic, the simulation showed that the average percent of OR time used for boarding was 0.05% with a maximum of 0.34%. For the decomposition heuristic, the average percent of OR time used for boarding was 0.27% with a maximum of 3.16%. Moreover, in 33 our of the 43 cases (77% of the instances) the 2-phase heuristic achieved less boarding than the decomposition heuristic. This is likely due to the stochastic performance analysis.

2.7.4 Hospital Case Study Results

We conducted another case study to compare the partner hospital's performance to the 2-phase heuristic performance. In this case study we considered four cases: the three previously studied services individually (general, orthopedic, and urology services), and case 4 combines the three services together, allowing for multiple services to share an OR. In each case we randomly sampled 25 days from the data set and compared schedules generated with respect to average OR overtime and average surgeon elapsed time across the 25 instances. Note that the 25 instances that combined the services (case 4) were independently sampled.

Our data set included planned surgery start times (i.e., start times estimated before the day of surgery), and realized surgery start times, planned and realized surgery durations, and realized recovery durations. We divided up the data set into two parts. The first part was used to establish surgery and recovery duration distributions. The second part was used to sample test instances for numerical analysis. In the planned schedules of the hospital, planned surgery start time and planned surgery duration were used from the data set, and in the realized schedules of the hospital, realized

surgery start time and realized surgery duration were used. In the planned heuristic schedules we used the 2-phase heuristic with the (60,70) percentile combination from the duration distributions to create the schedule. For realized heuristic schedules we used the realized surgery and recovery durations from the data set and the start times from planned heuristic schedules. If surgery was delayed due to overutilization, the surgery started as soon as the OR and the surgeon were available. We also allowed surgery to start 15 minutes before the scheduled start time if all resources were available to make the comparison fair, as this is common practice at our partner hospital. To give insight, we report overtime (OT) and surgeon elapsed time (SET) separately.

First we compared the averages of the realized values minus the planned values in the heuristic schedules and in the schedules of the hospital in terms of our performance metrics, OR overtime and surgeon elapsed time. This is shown in columns A and B in Table 2.2. The results show that both the 2-phase heuristic and the hospital tend to underestimate OR overtime in all cases, and surgeon elapsed time in case 2 and 3. However, both the heuristic and the hospital overestimate surgeon elapsed time in case 1 and 4. Overall, the heuristic is better.

Second, we looked at the performance metrics in terms of what the hospital planned for minus what the heuristic planned for, shown in column C. The results show that the hospital plans for more overtime in all cases except case 3, and that the hospital plans for more surgeon elapsed time in case 1 and 4. Third, shown in column D, we analyzed the performance metric in terms of what was realized at the hospital minus what would have been realized had the heuristic schedules been used. We find that, similar to the planned schedule comparison, the hospital had more overtime and more surgeon elapsed time in all cases, except case 3. The numbers suggest significant benefit from using the heuristic.

Case #	Surgical service	A		B		C		D		Avg # of ORs	Avg # of patients	Avg # of surgeons
		Heuristic difference: realized-planned (min)		Hospital difference: realized-planned (min)		Planned difference: hospital-heuristic (min)		Realized difference: hospital-heuristic (min)				
		OT	SET	OT	SET	OT	SET	OT	SET			
1	General	18	-8	4	-55	270	115	256	71	3	7	5
2	Orthopedic	22	21	37	46	127	-9	142	14	3	7	4
3	Urology	39	9	25	21	-38	-101	-53	-87	1	6	2
4	Integrated Model	11	-18	1	-46	159	129	149	103	4	7	5

Table 2.2: Comparison of 2-phase heuristic schedules to actual hospital schedules with respect to average OR overtime (OT) and surgeon elapsed time (SET) in minutes.

2.8 Conclusions

This chapter focused on the problem of creating single day elective surgery schedules while considering resources directly supporting surgery (i.e., ORs, surgeons) and resources indirectly supporting surgery (i.e., PACU). We proposed a fast 2-phase heuristic to solve this problem: in the first phase, LPT decides on the number of ORs to open and assigns surgeons to ORs, and in the second phase the difference heuristic sequences cases within each surgeon's block, and also sequences surgeon blocks in ORs. We found that our 2-phase heuristic, which is deterministic in nature, still performed well under uncertainty when evaluated with a discrete event simulation model, achieved high resource utilization and improved schedule predictability when compared to a much more computationally intensive heuristic that achieves near optimal solutions to MIP[OR,PACU]. It also performed well when compared to hospital schedules. Moreover, the 2-phase heuristic is not only fast and performs well, it is also very intuitive and provides researchers with sound insights. Also, it can be easily

implemented and used by healthcare professionals with a simple computational aid such as Excel, and without any difficult computational implementation or the use of a MIP solver. This is extremely important to hospitals, as most do not wish or have the opportunity to invest in and use complex and high-maintenance systems.

In addition to the practical advantages of the 2-phase heuristic, we also proved theoretical worst-case performance guarantees for both phases and showed that the bounds are tight.

In this chapter we focused on using the 2-phase heuristic to create the initial surgery schedule, but it can also be used for rescheduling surgical cases if changes need to be made on the day of surgery. One of the advantages of the 2-phase heuristic is that decisions that cannot be changed for a specific day can easily be fixed within the heuristic. This further aids the implementation of the methodology in settings with limited flexibility in decision making.

2.9 Appendix

2.9.1 Worst-Case Performance Guarantee of the LPT Heuristic

Dell'Olmo et al. (1998) proved that the longest processing time first (LPT) heuristic is a 13/12 approximation algorithm for a special case of the extensible bin packing problem, where the number of bins to be used is fixed. In this algorithm the items are ordered in decreasing length, and they are assigned in this order to the bin with the most available capacity, breaking ties arbitrarily. By a reduction from 3-PARTITION, it can be shown that this problem is strongly NP-hard (*Garey and Johnson* (1979)). Therefore a heuristic with a good worst-case performance ratio is highly desirable for the ability to tackle large instances of this problem. We extended the result of *Dell'Olmo et al.* (1998) to the extensible bin packing problem where there is a different cost associated with using a bin and extending the bin. We present our results in

the surgery scheduling framework, where bins are analogous to ORs, items are analogous to surgeon blocks, and extending a bin is equivalent to OR overtime. Note that this problem is the same as the MIP[OR] we formulated in 2.4.2, with the additional assumption that the planned session length of each OR is the same, S.

We use the notation of *Dell'Olmo et al.* (1998) described in a manner appropriate to our application. Let $\mathcal{A}$ be a set of surgeon blocks of duration p_k, where the number of surgeon blocks is n, and they are ordered in decreasing duration, i.e., $p_1 \geq p_2 \geq \cdots \geq p_n$. The main characteristic of a surgeon block is its duration, thus surgeon block k will be associated with its duration, p_k. In addition, a set of m ORs is given, $R_1, \ldots, R_m$, and each OR will be identified with the set of surgeon blocks it contains. An instance, $\mathcal{I} = (\mathcal{A}, m)$ is formed by the set of ORs and $\mathcal{A}$. For $A \subset \mathcal{A}$, $\ell(A)$ is called the *length*, and it is the sum of all surgeon blocks in A. Furthermore, $\ell(R_j)$ denotes the *load* of OR R_j, which is defined as the length of the surgeon blocks in R_j. Finally, $w(R_j)$ is the *size* of OR R_j, and it is defined to be $\max\{\ell(R_j), S\}$.

Consider a solution given by LPT to MIP[OR]. It is possible that in this solution some of the ORs have overtime, while others do not. If OR R_j has a load that is less than S, we say that R_j is *uncovered*. Otherwise, we say that R_j is *covered*. If R_j is uncovered, the difference between S and the load of R_j is called the *idle space*. If OR R_j was uncovered before surgeon block p_k was assigned to it by LPT, and R_j is covered after p_k is assigned to it, then p_k *covers* R_j. Moreover, we call surgeon blocks that are not bigger than $S/3$ *small* surgeon blocks, and we call surgeon blocks that are bigger than $S/3$ *big* surgeon blocks. In addition, big surgeon blocks with size greater than $2S/3$ are called *very big*, and big surgeon blocks between $S/3$ and $2S/3$ are called *medium*. This naming scheme is used to be consistent with the naming scheme of *Dell'Olmo et al.* (1998).

Now we define the cost of a solution. Consider a relaxation $\mathcal{F}$ of MIP[OR], where we can preempt each small surgeon block, i.e., small surgeon blocks can be broken

up into pieces and the pieces can be assigned to different ORs. The optimal solution of $\mathcal{F}$ is called the optimum semi-preemptive solution, $\mathrm{OPT_S}$. Let $C^H(\mathcal{I})$ be the total cost of the solution given by LPT and let $C^*(\mathcal{I})$ be the total cost of the optimal semi-preemptive solution associated with instance $\mathcal{I}$ (when obvious from the context, we omit the reference to the instance when talking about costs). In this section we show that $C^H(\mathcal{I})/C^*(\mathcal{I}) \leq 1 + \dfrac{Sc^v}{12c^f}$ for any instance.

We develop a worst-case performance guarantee for the LPT heuristic for MIP[OR]. The proofs given are extensions of, and closely parallel the proofs in *Dell'Olmo et al.* (1998). As a first step, we introduce a modified definition of a minimal counterexample.

Definition 2.1. An instance $\mathcal{I} = (\mathcal{A}, m)$ of surgeon blocks and m ORs is said to be a counterexample, if $C^H(\mathcal{I})/C^*(\mathcal{I}) > 1 + \dfrac{Sc^v}{12c^f}$. Moreover, a minimal counterexample also satisfies the following:

(i) there does not exists a counterexample that has a smaller number of ORs, and

(ii) there does not exist a counterexample that has a smaller number of big surgeon blocks.

If there exists a counterexample, it follows, that there exists a minimal counterexample. To further explore properties of minimal counterexamples we reintroduce a definition from *Dell'Olmo et al.* (1998).

Definition 2.2. We say that OR R_j^* of $\mathrm{OPT_S}$ dominates OR $R_i = \{\chi_1, \ldots, \chi_r\}$ of the LPT solution, if there is a partition $P_1^*, \ldots, P_r^*$ of the big surgeon blocks of R_j^* such that $\ell(P_t^*) \geq \chi_t$ for $t = 1, \ldots, r$, where $\{\chi_1, \ldots, \chi_r\}$ represents the set of surgeon blocks when there are r blocks assigned to OR R_i.

Lemma 2.3. *Let B_i be an OR that is covered in the LPT solution to MIP[OR] in a minimal counterexample. Then B_i will not be dominated by any OR B_j^* of OPT_S.*

Proof. This proof is by contradiction. Let us suppose that there exists an OR B_j^* that dominates B_i. Now consider a new instance, call it $\mathcal{I}'$, that we get if we delete OR B_i and every surgeon block in it. The LPT assignment of $\mathcal{I}$ and $\mathcal{I}'$ is exactly the same, the only difference is that we do not have OR B_i in $\mathcal{I}'$. Therefore, $C^H(\mathcal{I}') = C^H(\mathcal{I}) - c^f - (w(B_i) - S)c^v$.

Next, from OPT_S let us create a new assignment for $\mathcal{I}'$. From the OPT_S solution delete surgeon blocks χ_t $(t = 1, \ldots, r)$ that were in B_i, and replace them with the elements that correspond to them in P_t^*, the partition set. Then assign the rest of the surgeon blocks of B_j^* (i.e., the small surgeon blocks) randomly to the other ORs, and delete B_j^*. We know that $l(P_t^*) - \chi_t \geq 0$ for $t = 1, \ldots, r$, and that $\ell(B_i) = w(B_i)$, since B_i is a covered OR. Therefore

$$\begin{aligned} C^*(\mathcal{I}') &\leq C^*(\mathcal{I}) - (w(B_j^*) - S)c^v - c^f + (w(B_j^*) - \ell(B_i))c^v \\ &= C^*(\mathcal{I}) - w(B_j^*)c^v + Sc^v - c^f + w(B_j^*)c^v - \ell(B_i)c^v \\ &= C^*(\mathcal{I}) - c^f - w(B_i)c^v + Sc^v, \end{aligned}$$

where the inequality holds, because $C^*(\mathcal{I}')$ can only be better than taking the optimal solution for instance $\mathcal{I}$, and replacing the surgeon blocks of R_i by their corresponding element from the partition set, and randomly distributing the small surgeon blocks. But

$$\begin{aligned} C^H(\mathcal{I}') = C^H(\mathcal{I}) - c^f - (w(B_i) - S)c^v &> \left(1 + \frac{Sc^v}{12c^f}\right) C^*(\mathcal{I}) - c^f - (w(B_i) - S)c^v \\ &\geq \left(1 + \frac{Sc^v}{12c^f}\right) (C^*(\mathcal{I}') + c^f + w(B_i)c^v - Sc^v) - c^f - w(B_i)c^v + Sc^v \\ &= \left(1 + \frac{Sc^v}{12c^f}\right) C^*(\mathcal{I}') + \frac{Sc^v}{12c^f}(c^f + (w(B_i) - S)c^v), \end{aligned}$$

since B_i is a covered OR, $w(B_i) \geq 0$, and $c^v(w(B_i) - 1) \geq 0$. This contradicts the fact that $\mathcal{I}$ is a minimal counterexample. □

Lemma 2.4. *In a minimal counterexample there is no OR in the LPT solution which contains surgeon blocks a, b $(a \geq b)$ such that $a + b > S$ and $a < 2S/3$.*

Proof. Let us number the ORs so that surgeon block p_k is assigned to OR B_k $(k = 1, \ldots, m)$ by LPT. By the setup of the algorithm, surgeon block p_{m+k} is assigned to the OR with smallest load, beginning with the assignment of surgeon block p_{m+1} to OR B_m. Let OR B_j be the "first" OR, i.e., the OR with the smallest index, such that $B_j = \{a, b\}$ with $a + b > S$ and $a < 2S/3$. Naturally, $a = p_j$. From $a < 2S/3$ and $b > S/3$ we can also conclude that $b = p_{2m-j+1}$.

If in $\mathrm{OPT_S}$ at most two surgeon blocks of $T_1 := \{p_1, \ldots, p_{2m-j+1}\}$ are contained in any OR, then there exists an OR containing two surgeon blocks of T_1 where one of these two surgeon blocks is at least as large as a. If in $\mathrm{OPT_S}$ there is an OR with three surgeon blocks x, y, z of T_1, then $x + y > a$ and $z \geq b$ since $a < 2S/3$ and b is the smallest element of T_1. In either case, we found an OR which dominates the covered OR, B_j, which is a contradiction to Lemma 2.3. □

Lemma 2.5. *Let k_1 be the number of big covering surgeon blocks in an LPT solution, also called critical surgeon blocks. The critical surgeon blocks in a minimal counterexample have the following properties:*

(a) The critical surgeon blocks are exactly the k_1 smallest among the big surgeon blocks, and all critical surgeon blocks are medium surgeon blocks.

(b) There is an optimal semi-preemptive solution in which all the critical surgeon blocks are assigned to covered ORs which contain either a very big surgeon block and a medium surgeon block, or three medium surgeon blocks.

Proof. If $k_1 = 0$, the result is trivial. Therefore, we assume that $k_1 > 0$. There are no more than m very big surgeon blocks. Otherwise, a contradiction to Lemma 2.3 can be found, similar to the argument in the proof of Lemma 2.4. Let r be the number of very big surgeon blocks. Observe the assignments just prior to the time the first

critical surgeon block is assigned by LPT. By Lemma 2.4 there are r ORs with a single very big surgeon block, and $m - r$ ORs with two medium surgeon blocks. At that time the load of each OR is greater than $2S/3$ but smaller than S. Therefore each of the following medium surgeon blocks will become covering surgeon blocks. Claim (a) follows.

If we exchange a very big surgeon block with a medium surgeon block, we can guarantee that there is no OR in $\mathrm{OPT_S}$ with two very big surgeon blocks, because the total size will not be increased. Let r_1 be the number of ORs in $\mathrm{OPT_S}$ that contain a very big surgeon block and a medium surgeon block. Then there are $r - r_1$ ORs with very big surgeon blocks but no other big surgeon blocks. There remain $m - r$ ORs with exactly $2(m-r) + k_1 - r_1$ medium surgeon blocks, where the total number of medium surgeon blocks is $2(m - r) + k_1$. At least $k_1 - r_1$ of them have 3 medium surgeon blocks. Altogether, at least k_1 ORs exist with total length of the big surgeon blocks greater than S. Therefore, we found an optimal semi-preemptive solution with a set of K_1 of at least k_1 covered ORs that have only big surgeon blocks assigned to them, with a minimum of one medium surgeon block per OR. If we exchange any critical surgeon block not assigned to an element of K_1 with a medium surgeon block of an OR B_j in K_1, the load of OR B_j is sill greater than S. Claim (a) ensures that the total size will not increase. Therefore, claim (b) follows. □

Lemma 2.6. *In a minimal counterexample critical surgeon blocks cannot exist.*

Proof. Assume $k_1 > 0$ and let the total length of critical surgeon blocks be $\delta + k_1 S/3$. Obtain a new instance, $\mathcal{I}'$, through replacing all critical surgeon blocks with surgeon blocks that have a length of exactly $S/3$. By Lemma 2.4, the LPT surgeon block to OR assignments do not change, and $C^H(\mathcal{I}') = C^H(\mathcal{I}) - \delta c^v$. According to Lemma 2.5, it is possible to create an optimal semi-preemptive solution where all critical surgeon blocks are assigned to ORs that only have big surgeon blocks assigned to them. After making the critical surgeon blocks smaller, the load of the ORs they are assigned to

are still at least S. Therefore, $C^*(\mathcal{I}') \leq C^*(\mathcal{I}) - \delta c^v$. This is a contradiction to the fact that the counterexample is minimal. □

Corollary 2.7. *The total length of all big surgeon blocks in a minimal counterexample is not greater than m. Furthermore, the big surgeon blocks can be assigned to ORs without covering any OR.*

We are now ready to use the above Lemmas and Corollary 2.7 to prove our main result about the worst case performance of LPT for MIP[OR].

Theorem 2.1. *For any instance $\mathcal{I}$, where the planned session length of each OR is S, we have*

$$\frac{C^H}{C^*} \leq 1 + \frac{Sc^v}{12c^f},$$

where an instance is defined by the list of surgeon blocks and the number of ORs available. Moreover, there exist instances for which this bound is tight.

Proof. As a reminder, we are considering a minimal counterexample, and we can make the assumption that the total length of all surgeon blocks, $\mathcal{L}$, does not exceed mS. Otherwise, let $\mathcal{L} = mS + \delta$ with $\delta > 0$. By the assumption that we can preempt small surgeon blocks, and by Corollary 2.7, we know that $C^* = mc^f + \delta c^v$. If we delete the smallest surgeon blocks such that the total length deleted would be δ, we get a new instance $\mathcal{I}'$. Note: we might have to break one surgeon block, but no more than one. Then, $C^H(\mathcal{I}') \geq C^H(\mathcal{I}) - \delta c^v$. Corollary 2.7 tells us that only small surgeon blocks were deleted, and due to the fact that small surgeon blocks can be preempted, it follows that $C^*(\mathcal{I}') = C^*(\mathcal{I}) - \delta c^v = mc^f$, and thus $\mathcal{I}'$ is a new counterexample that has a worse performance ratio, and the total length of surgeon blocks is mS.

Now take the LPT solution, and reorder the ORs so that the first t ORs would be covered, i.e., ORs $B_1, \ldots, B_t$, and the rest of the ORs are not covered. For each OR B_j with $j = 1, \ldots, t$, let the length of the covering surgeon block of the OR be

$a_j + b_j$, where a_j is the part of the surgeon block that fills the OR, and b_j the part of the surgeon block that is in overtime. Furthermore, in ORs $B_j,\ j = t+1, \ldots, m$, i.e., the uncovered ORs, let c_j be the idle space. Due to the fact that $\mathcal{L} \leq mS$, $\sum_{j=1}^{t} b_j \leq \sum_{j=t+1}^{m} c_j$. Furthermore, $a_i \geq c_j$, for $i = 1 \ldots, t$ and $j = t+1, \ldots, m$, because every surgeon block of $\mathcal{A}$ is assigned by LPT to the OR with the most available time. Therefore,

$$(m-t)\sum_{j=1}^{t} a_j \geq t \sum_{j=t+1}^{m} c_j \geq t \sum_{j=1}^{t} b_j.$$

If we add $(m-t)\sum_{j=1}^{t} b_j$ to both sides, we get

$$(m-t)\sum_{j=1}^{t}(a_j + b_j) \geq m \sum_{j=1}^{t} b_j.$$

Since

$$(m-t)\sum_{j=1}^{t}(a_j + b_j) \leq \frac{S}{3}(m-t)t$$

using Lemma 2.6, and

$$\frac{S}{3}(m-t)t \leq \frac{S}{3}\left(\frac{m}{2}\right)^2$$

it follows, that

$$\sum_{j=1}^{p} b_j \leq \frac{mS}{12}$$

and $C^H \leq mc^f + \frac{1}{12}mSc^v$. Therefore $\dfrac{C^H}{C^*} \leq 1 + \dfrac{Sc^v}{12c^f}$, which is a contradiction to the existence of a minimal counterexample.

To show that this bound is tight, consider the following instance. The number of ORs, m, is even, and there are m surgeon blocks with length $S/2$ and $\frac{3}{2}m$ surgeon blocks of length $S/3$. Then the LPT solution will give a cost of $mc^f + \frac{mS}{12}c^v$, while the optimal solution gives a cost of mc^f. □

This proves the worst-case performance guarantee of the LPT heuristic with different costs associated with regular time and overtime for a given number of ORs. Note that when $c^f = c^v$ and $S = 1$, which is the case considered in *Dell'Olmo et al.* (1998), this result is the same as their result.

Also note that the theorem applies to C^*, which is the optimal solution to the semi-preemptive problem, a relaxation of the non-preemptive problem, MIP[OR]. Therefore it directly follows that the following bound holds: $\frac{C^H}{C_N^*} \leq 1 + \frac{Sc^v}{12c^f}$, where C_N^* is the optimal solution of the non-preemptive problem.

2.9.2 Worst-Case Performance Guarantee of the Difference Heuristic

Theorem 2.2. *In the difference heuristic setting, where $W_{ij} = r_i - d_j$ for $i \neq j$ and $W_{ii} = \infty$, let*

$$W^i = \max_{j:j\neq i}(W_{ij})^+; \quad \bar{W}^i = \min_i W^i; \quad w^i = \min_{j:j\neq i}(W_{ij})^+; \quad \bar{w}^i = \max_i w^i \quad \forall i,$$

$$W^j = \max_{i:i\neq j}(W_{ij})^+; \quad \bar{W}^j = \min_j W^j; \quad w^j = \min_{i:i\neq j}(W_{ij})^+; \quad \bar{w}^j = \max_j w^j \quad \forall j.$$

Then for any instance we have

$$C^{DH} - C_1^* \leq c^s \cdot \min\left\{ \sum_{i=1}^{P} W^i - \bar{W}^i - \left(\sum_{i=1}^{P} w^i - \bar{w}^i \right), \right.$$
$$\left. \sum_{j=1}^{P} W^j - \bar{W}^j - \left(\sum_{j=1}^{P} w^j - \bar{w}^j \right) \right\},$$

where C^{DH} is the cost of the schedule given by the difference heuristic, and C_1^* is the cost of the optimal solution. Moreover, there exist instances for which this bound is tight.

Proof. Surgeon elapsed time is the sum of two components: surgery duration and idle time. Since surgery duration is a fixed constant, minimizing idling will minimize surgeon elapsed time. The entry $(W_{ij})^+$ for $i \neq j$ denotes the idling if patient i is

followed by patient j. Finding a sequence that minimizes idling is equivalent to the traveling salesman problem (TSP), except that we can break the tour between two selected patients i and j, since we require a sequence, not a tour.

The assignment problem is a well known relaxation of the TSP in which binary decisions determine the assignment of prior ("from") nodes to subsequent ("to") nodes. In our context we let y_{ij} denote a binary decision variable that equals 1 if patient i is followed by patient j, and zero otherwise. Then the following is the formulation of the assignment problem, [AP]:

$$\min \sum_{i=1}^{P}\sum_{j=1}^{P}(W_{ij})^{+}y_{ij} \tag{2.32}$$

$$\text{s.t.} \sum_{i=1}^{P} y_{ij} = 1 \qquad \forall j \tag{2.33}$$

$$\sum_{j=1}^{P} y_{ij} = 1 \qquad \forall i \tag{2.34}$$

$$y_{ij} \in \{0,1\} \qquad \forall i,j. \tag{2.35}$$

There are two more ways to relax this assignment problem: we can relax constraints (2.33) or (2.34). Relaxing (2.33) corresponds to the first term in the minimum in the bound in the theorem, and relaxing (2.34) corresponds to the second term. Note that the result from solving this assignment can be reduced by breaking the tour to obtain a sequence, as noted above.

When (2.33) is relaxed and (2.32) is maximized subject to the remaining constraints, the solution of [AP] with a term subtracted from the objective to break the tour represents the worst case idling, which is $\sum_{i=1}^{P} W^i - \bar{W}^i$, i.e., the sum over the maximum idling in each row minus the minimum of those values. When (2.33) is relaxed and (2.32) is minimized subject to the remaining constraints, the solution of [AP] with a term subtracted from the objective to break the tour represents the best

case idling, $\sum_{i=1}^{P} w^i - \bar{w}^i$, i.e., the sum over the minimum idling in each row minus the maximum of those values. Then the first term of the minimum on the right hand side of the inequality in the theorem represents the difference between the worst and best case scenarios, which gives an upper bound on the difference of the heuristic cost and the optimal cost when multiplied by the cost of surgeon elapsed time.

Similarly, the second term of the minimum is used to find an upper bound on the difference in the cost of surgeon elapsed time between the heuristic and optimal solutions by relaxing (2.34) instead of (2.33), and considering the columns of W instead of its rows.

Each of the two terms are valid upper bounds on the difference in idling between the heuristic and optimal solutions. Thus, the minimum of the two upper bounds is an upper bound.

To see that the bound can be tight, consider any instance where the longest recovery duration does not exceed the shortest surgery duration, i.e., $\max_i r_i \leq \min_i d_i$. In this case no sequence of surgeries can cause idling, thus any sequence results in the optimal cost, and the left hand side of the equation will be zero. The right hand side will also be zero, since each non-diagonal entry in the matrix W will be non-positive. This will ensure that both terms in the minimization will be zero. Therefore there exist instances for which the bound is tight. □

Theorem 2.3. *For any instance with a single surgeon, the difference heuristic results in an optimal sequence if the number of cases assigned to the surgeon is two.*

Proof. Suppose we have 2 patients, where surgery and recovery durations for patient 1 are (d_1, r_1) and for patient 2 are (d_2, r_2). Then

$$W = \begin{bmatrix} \infty & r_1 - d_2 \\ r_2 - d_1 & \infty \end{bmatrix}$$

If both $r_1 - d_2$ and $r_2 - d_1$ are non-positive, any schedule is optimal. If at least one of them is positive, we have two cases.

Case 1: $r_2 - d_1 \leq r_1 - d_2$. In this case the heuristic will pick patient 2 to go first and patient 1 to go second, so the idling is $(r_2 - d_1)^+$.

Case 2: $r_2 - d_1 > r_1 - d_2$. In this case the heuristic will pick patient 1 to go first and patient 2 to go second, so the idling is $(r_1 - d_2)^+$.

Thus the sequence picked is such that total idling equals the $\min\{(r_2 - d_1)^+, (r_1 - d_2)^+\}$. □

CHAPTER III

Planning Models for Skills-sensitive Surgical Nurse Staffing

3.1 Introduction

As discussed in Chapter I, due to the fact that a significant proportion of hospital revenues and cost can be attributed to operating rooms (ORs), this makes ORs one of the most important areas in hospitals where improvement efforts can lead to substantially lower cost and increased revenue. Part of the OR cost can be attributed to surgical nurses, which are a highly specialized and critical resource for effective surgical care.

Surgical services have evolved over time with the development of new procedures and numerous subspecialties. The expansion of surgical specialties also made surgical nurse specialization necessary, due to the increased complexity of the services. Most hospitals organize surgical nurses into teams, where each team covers several surgical services, and nurses assigned to a team are trained in all services in that team. Nurses are hired into one of these teams either as a *fixed nurse* (a nurse that would be assigned to a specific service most of the time), or as a *float nurse* (a nurse assigned to a team who would regularly work in multiple services within the team). The main difference between fixed and float nurses is that fixed nurses are always assigned to their specific service, unless demand for surgery on a specific day is low, thus less

nurses are needed in that service, and therefore fixed nurses need to be reassigned for the duration of their current shift. Float nurses, however, do not have a specific service they spend the majority of their time in, they regularly work in different services within their team. Thus the necessity for both fixed and float nurses to be trained in each service belonging to their team, but fixed nurses are more skilled in their assigned service than float nurses or other fixed nurses assigned to a different service. Each surgery requires at least two nurses to be present: a *surgical technician* (which for the purposes of this chapter we consider a type of nurse, even though they might not be registered nurses), who works within the sterile field and hands instruments to the surgeon, and a *circulator nurse* who works outside of the sterile field and assists in setup, charting, and acquiring instruments during surgery, if necessary.

Surgical demand on a case level is highly variable; however, there is some predictability in terms of specialty because of OR block scheduling policies in place at most hospitals. Under a block booking system, services have OR time allocated to them on given days. This block structure is changed infrequently. If the services are unable to schedule enough cases to fill up their allocated time, their unused time becomes available to other services a few days in advance. Some ORs can also be reserved for urgent and emergent cases with an open scheduling policy, meaning that such patients, who have to be operated on within a day or on the same day, are scheduled into open ORs on a first come first served basis. Moreover, due to the uncertainty in surgery durations and emergent cases arising that exceed capacity, surgeries can be rescheduled within the day of surgery.

There is also uncertainty associated with the number of skill-specific nurses needed by hour of day, which presents a challenge for staffing. Because of this variability, surgical services are often understaffed or overstaffed, which requires corrective recourse actions. These may include calling in temporary nurses from an agency, or sending nurses home before completing their shift. This uncertainty can also necessitate the assignment of a nurse to a service in which they do not usually work, and thus are

unfamiliar with the other staff on the procedures. *Pronovost and Freischlag* (2010) highlight that the lack of teamwork is a significant contributor to surgical adverse events, and point out that it has been shown that "improving teamwork was associated with reduced surgical mortality". This is important, because *Anderson and Talsma* (2011) found that surgeries that happen in the afternoon or later in the day, tend to have medical professionals (i.e., surgeons, nurses) working together that do not work together regularly. To address this, an important consideration is to make sure that a sufficient number of fixed nurses are allocated to services.

Based on collaboration with a large academic hospital, we focus on improving surgical nurse staffing to ensure that nurses are assigned to cases that they are qualified to staff, and that staffing levels fit surgical demand. We address the complex problem of surgical nurse staffing in two stages. In the first stage, we group services into surgical teams to optimize the fairness of teams in terms of the time required for cross training, the overnight surgical volume, and the number of assigned ORs for each service. The purpose of the teams is to create groups of surgical services, where a nurse who specializes in one of the services in a team can also cover the other services in the team when necessary, thus maintaining flexibility in the system. In the second stage, given the new team assignments, we determine daily fixed shifts that need to be staffed for services and float shifts that need to be staffed for teams, subject to constraints on the nurse-to-OR ratio required for each service, the number of nurses available, and historical OR usage data for each service. The separation of these stages is necessary, because of the significantly different time line for the decisions made in the two stages. Service group teams are determined for the long term (on the order of years), and are not easy to change once established. On the other hand, shift allocation can be frequently changed (on the order of weeks or months) to ensure that the shifts fit surgical demand. Decomposition into two stages is also necessary to reduce an otherwise computationally difficult problem to one with manageable computation time.

The remainder of this chapter is organized as follows. In Section 3.2 we conduct a review of relevant literature and highlight our contributions to the literature. In Section 3.3.1 we formulate the service group team design problem that allocates services to teams with the objective of balancing training time, overnight surgical volume and the number of ORs across teams, thus creating a team structure that achieves fairness to nurses regardless of their team. In Section 3.3.2 we formulate the shift design and allocation problem that determines the number and type of shifts that are needed to be assigned to services and teams by hour of day and day of week to ensure coverage of surgical demand, i.e., that there are enough nurses working to staff surgical cases. We further present a 2-phase heuristic that is an optimization based decomposition of the shift design and allocation model. In Section 3.4 we study the performance of the 2-phase heuristic through a numerical case study. In Section 3.5 we present a detailed case study based on data provided by our partner hospital. Section 3.5.1 presents results for stage 1, where we present recommendations for team redesign. Section 3.5.2 presents results for the stage 2 problem, where we compare hospital performance to both the optimized teams and the current teams in use. We finish with conclusions in Section 3.6.

3.2 Literature Review

Staffing decisions in general can be broken down into several stages. *Bard* (2010) classified nurse staffing decisions into four stages: long term planning (e.g., size of workforce, shift design), midterm scheduling (e.g., shift assignment to nurses, cross-training), short term scheduling (e.g., OR or case assignment to nurses, vacations, temporary nurses), and real time control (e.g., responding to absenteeism). In his review, *Bard* (2010) identified midterm scheduling and short term scheduling, as the areas that most research has focused on, and he highlighted the need for more research in the area of long term staffing decisions. Our chapter helps to address this gap in the literature.

The problem of personnel staffing and scheduling has been a focus of many research studies across multiple industries, including, but not limited to: emergency response (*Fry et al.* (2006)), service delivery systems (*Ingolfsson et al.* (2002)), call centers (*Aksin et al.* (2007)), and airlines (*Barnhart et al.* (2004)). There has been a significant amount of research in nurse staffing as well, mostly focusing on assigning nurses to days of week and shifts within days, and distributing off days across nurses. More details can be found in reviews such as *Bard* (2010), *Cheang et al.* (2003), and *Ernst et al.* (2004), who also highlight other application areas.

We highlight some related work from other healthcare areas. *Kazemian et al.* (2013), for example, presented an integer programming based method to minimize patient hand-offs in their resident and fellow shift design problem. They considered required constraints, such as national shift guidelines, and desired constraints, such as livability rules, and investigated the affect of desired constraints on the objective. They showed that the desired constraint about shift length had the largest impact. This work focused on a single unit, thus specialization and skills are not considered, which plays an important role in our problem.

Floor nurse scheduling has been well studied as well. *Maass et al.* (2015) presented a stochastic programming model to determine the optimal number of nurses needed by type. They considered three nurse types: unit, pool, and temporary. They also accounted for nurse absenteesm. Using a genetic algorithm, their solutions indicated potential for significant financial savings. This work, however, only made suggestions for the number of nurses to hire, and not for shift assignments to units to account for demand. *Kim and Mehrotra* (2015) considered a single unit integrated staffing and scheduling problem as a two stage stochastic programming problem. In the first stage they picked the number of workers for each time period by choosing from available staffing patterns. In the second stage they made necessary adjustments to the schedule, which are penalized. They showed that using mixed integer rounding inequalities, they can achieve a tight formulation of the second stage, thus

the integrality constraints can be relaxed. They also presented a modified version of the L-shaped method to further improve computational time. They showed that considering future uncertainty can result in significant cost savings. However, their method applies to a single unit, and assumes that skills are the same.

Improvement of staffing has been used to provide better patient access. *Woodall et al.* (2013) used a simulation-optimization approach for this objective at the Duke Cancer Institute in Durham, NC. They used a mixed integer program (MIP) for monthly and weekly planning, with the objective of minimizing the total shortage of nurse hours relative to patient demand. Then they used a simulation-optimization approach to determine daily shift start times that minimize average direct patient wait time. With this method they showed it was possible to decrease patient wait time by as much as 25 minutes in some cases. They also found that part time nurses help mitigate the uncertainty in demand, and that the greatest impact was achieved by changing arrival and departure times for nurses. Our approach is solely based on optimization methods, and we distinguish between different nurse types, i.e., fixed and float.

Closest to this chapter is the work of *Villarreal and Keskinocak* (2016). They considered the long term surgical nurse staffing problem, and took a two phase approach to the problem. In the first phase, they used a linear program to find the staffing budget and the number of full time equivalents (FTEs). Then in the second phase they used a MIP to create a staffing structure, i.e., decide how the previously determined FTEs should be allocated across days and shifts with the objective in minimizing the difference between allocated and required staffing levels. They also based their models on historical OR usage data. Contrary to our approach of determining the teams, they took the team assignments as given. Moreover, they based their shift assignments solely on teams, i.e., float nurses, whereas our solutions propose shift assignments to specific services as well, i.e., fixed nurses. Another main distinction is that our objective focuses on the even distribution of shifts across teams and day

periods, as well as minimizing the usage of undesirable shifts, and that our models ensure that required staffing levels are fulfilled.

Our method optimizes the assignment of nurses to teams, exploiting the resulting workforce flexibility to manage the inherent uncertainty in surgical demand. This general idea has been explored by *Jordan and Graves* (1995) and *Hopp and Van Oyen* (2004) in a general setting, and by *Inman et al.* (2005) in the nurse staffing setting. However, we take a different approach that fits well with current hospital practice: we assign service specific shifts within service group teams, and also assign float shifts to teams to address the uncertainty in surgical demand. Both these shift types belong to full time nurses, i.e., not temporary nurses.

3.2.1 Our Contributions to the Literature

We propose to address the problem of longterm surgical nurse staffing with the following two stage approach. First, we formulate a MIP to determine service group teams, i.e., the partition of services into disjoint subsets, with the objective of balancing the training time, overnight surgical volume and the number of ORs used by each team, motivated by fairness to nurses across teams. Second, based on the teams created in stage 1, we formulate a second MIP to determine which shifts need to be staffed for each service (fixed shifts), to ensure patient safety standards are met, i.e., nurses with necessary experience in that service are available, and then allocate the remaining available shifts to teams (float shifts) to mitigate the uncertainty in surgical demand. We further ensure that a predefined minimum nurse-to-OR ratio is respected. We present a heuristic for the second stage that can be used to speed up computational time, when solutions are needed quickly, and where intuitive understanding of how the decisions are made is necessary for implementation. To our knowledge, both of these two stages have not been considered together before to design planning level staffing models. Moreover, the main focus of the team design stage is to ensure fairness to nurses regardless of their team, which is not the main focus

of previous studies, where fairness means that a nurse can expect approximately the same training time and overnight surgeries, regardless of the team they are assigned to. Based on data from our partner hospital, we showed that using our methodology can improve surgical demand coverage, even with less nurses than current staffing levels.

3.3 Problem Formulation

The objective of the design of surgical nurse staffing levels and schedules is to ensure that there is a sufficient number of nurses skilled in the necessary service for every surgery performed on a given day during the week. To ensure this, we need to consider several factors. First, due to the advances in recent years that has increased the complexity of surgical services, a nurse cannot be expected to master every surgical service. Moreover, as mentioned in Section 3.1, teamwork is an important patient safety factor, and having the same people working together consistently contributes to high quality care. Therefore, we need to determine the subsets of services a nurse should cover, which we call service group teams, or simply teams for short.

Once the teams are determined, shifts need to be designed to cover demand. As we are focusing on a planning level tool, we do not assign specific nurses to days or shifts, rather we determine the necessary shifts that need to be staffed to satisfy demand. To do this, we choose from the most common shift types currently in use at our partner hospital to determine how many of each shift type need to be staffed by day of week. Once the shifts are determined, individual nurses can be assigned by a skilled nurse manager to the needed shifts, based on skills and preferences. It is important to note that throughout the chapter the word nurse does not refer to a specific person, but to an FTE assigned to a collection of shifts within a week, whose total duration equals a typical full time surgical technician or circulator nurse weekly work hours.

Our main assumptions are the following. We only focus on weekdays, because generally hospitals do not offer elective surgery services during the weekend in their inpatient ORs. As our models are for planning purposes, we do not distinguish between surgical technicians and circulator nurses. Moreover, as this consistent with current practice at our partner hospital, we assume that each nurse is able to cover each service in a team, regardless of whether they are a fixed nurse (assigned to a service within a team) or a float nurse (assigned to a team to mitigate the effect of random demand). This allows the hospital to take advantage of skill flexibility to better adapt to the variable nature of surgical volume. However, fixed nurses still spend the majority of their time in their specified service, and are only temporarily assigned to a different service if surgical demand is less than expected in their specified service. We assume a block schedule that allocated OR time to surgical services for elective surgery. We also assume that some ORs are designated *open*, i.e., to be used for urgent and emergent surgeries.

3.3.1 Service Group Team Design

We have established that teams are needed due to the complexity of surgical services, for nurses to be able to maintain their skills, and to better facilitate teamwork. Thus as a first step in improving surgical nurse staffing, we group surgical services together into teams, i.e., the set of surgical services is divided into disjoint subsets. Our partner hospital currently has three teams; however, the current teams were created almost a decade ago considering only overnight call equity, and they have not been updated. Because all else is dramatically impacted by the design on the teams, the first step in improving surgical nurse staffing is to redesign these teams.

The first-stage problem objective consists of three main components that we strive to balance across teams to achieve fairness. The first one is the maximum training time across teams, where the training time of any team is the sum of the training time of the services assigned to the team. This is important because our partner hospital

desires to have the same amount of time allocated to train nurses, thus to ensure fairness to nurses across teams, we need to ensure that training time is balanced. The second component is the maximum percent of overnight surgeries across teams, where the percent of overnight surgeries of any team is the sum of the percent of overnight surgeries of each service assigned to the team. Being on call overnight is undesirable for nurses, thus minimizing the number of on-call nurses is important from the nurses' perspective. However, there are services that require specialized nurses on-call, even if they have low overnight surgical volume, such as transplant. Therefore the available on-call nurses as well as overnight surgical volume need to be taken into consideration when designing teams to ensure fairness to nurses in each team. The third component is the maximum number of ORs by day across teams. The number of ORs will determine the number of nurses needed for the team, therefore by balancing the number of ORs, we approximately balance team size. This is important, because larger teams have more room for flexibility if surgical volume deviates from expected values, while smaller teams have less room for flexibility. Thus by balancing team size, we ensure similar levels of flexibility for each team.

To reduce variation in these performance metrics, the objective is to minimize the weight associated with the maximum training time across teams, the maximum percent of overnight surgical volume for each team, and the maximum number of ORs by day per team. Note that the number of teams is an input to the model, not a decision variable. This input can be fixed to align with hospital needs, or by varying this input and repeatedly solving the model, the best solution can be found.

One of the most important inputs to this model is the training time of a service. To measure this, a group of nurse educators and nurse managers filled out a matrix whose elements denote the number of weeks of training needed to learn a single service j, if a nurse is already skilled in one other service, i. For more details on the survey process conducted by the partner hospital and the data, please see Appendix 3.7.1. The following is our notation.

Indices:

i, j index of services, $i, j = 1, \ldots, S$.

t index of teams, $t = 1, \ldots, T$.

d index of days, $d = 1, \ldots, 5$.

Data:

r_{di} number of ORs that service i has on day d.

w_{ij} amount of training needed in weeks to learn service j if the nurse already knows service i.

z_i indicates if a service needs a specialized on-call group ($z_i = 1$), or not ($z_i = 0.1$).

u_i percent of overnight surgical volume attributed to service i.

A desired number of on-call groups per team.

c_W weight associated with maximum weeks of training across teams.

c_U weight associated with maximum percent of overnight surgical volume across teams.

c_R weight associated with maximum number of ORs across teams.

Decision Variables:

x_{it} binary variable indicating whether service i is assigned to team t ($x_{it} = 1$), or not ($x_{it} = 0$).

y_{ijt} binary variable indicating is services i and j are assigned to the same team t ($y_{ijt} = 1$), or not ($y_{ijt} = 0$).

W maximum number of weeks of training across teams.

U maximum percent of overnight surgical volume across teams.

R maximum number of ORs across teams.

Mathematical Model:

$$\min \; c_W W + c_U U + c_R R \tag{3.1}$$

$$\text{s.t. } y_{ijt} \geq x_{it} + x_{jt} - 1 \qquad \forall i, j, t \tag{3.2}$$

$$\sum_{t=1}^{T} x_{it} = 1 \qquad \forall i \tag{3.3}$$

$$\sum_{i=1}^{S} \sum_{j=1}^{S} y_{ijt} w_{ij} \leq W \qquad \forall t \tag{3.4}$$

$$\sum_{i=1}^{S} u_i x_{it} \leq U \qquad \forall t \tag{3.5}$$

$$\sum_{i=1}^{S} x_{it} r_{di} \leq R \qquad \forall d, t \tag{3.6}$$

$$\sum_{i=1}^{S} z_i x_{it} = A \qquad \forall t \tag{3.7}$$

$$x_{it}, y_{ijt} \in \{0, 1\}; R, W, U \geq 0 \qquad \forall i, j, t \tag{3.8}$$

The objective (3.1) minimizes the weight associated with the maximum training time across teams to balance the skill requirements of teams, the maximum percent of overnight surgical volume across teams to balance on-call requirements, and the maximum number of ORs over days to balance team size. The purpose of this is to attempt to reduce variation in these performance metrics. Constraints (3.2) determine whether two services are in the same team. Constraints (3.3) require each service to be assigned to exactly one team. Constraints (3.4) determine the maximum training time based on the team assignments and how long it takes to learn one service if you already know another. Constraints (3.5) calculate the maximum percent overnight surgical volume across teams. Constraints (3.6) determine the maximum number of ORs by day across teams. Constraints (3.7) ensure that the required number of on-call groups are assigned to each team. In the remainder of this chapter we will refer to this model as MIP[Team].

3.3.2 Shift Design and Allocation

The second component of a nurse staffing plan is deciding which shifts should be used by day of week, and how to best allocate the chosen shifts across services and teams to cover surgical demand. To do this, we choose which shift types to staff from the commonly used shift types at our partner hospital, that are acceptable to the labor union. We break up each day into periods to model this problem, where a new period starts if an available shift type starts or ends at that hour.

The objective in this problem is to distribute the nurses evenly across day periods, i.e., to make sure that the minimum number of nurses per OR in day periods is approximately the same in each team. Note that the minimum number of nurses is calculated considering both fixed and float nurses. During night periods we assign a set number of float shifts to teams that matches the number of ORs open at night. Moreover, we penalize the usage of undesirable shifts that can be defined by start time, end time, or duration.

Note that the first-stage model considered the block schedule as a measure of workload for each service, but in this second-stage model we consider a combination of the block schedule and historical usage data as a measure for workload, to account for services utilizing OR time outside of their block time, e.g., for emergent surgeries. The following is our notation.

Indices and sets:

g	index for nurses, $g = 1, \ldots, G$.
p	index for periods of day, $p = 1, \ldots, 12$.
s	index for shift templates.
$\mathbb{P}, \bar{\mathbb{P}}$	set of day periods and night periods, respectively.
$\mathbb{S}, \bar{\mathbb{S}}$	set of desirable and undesirable shifts, respectively.

Data:

x_{it} indicates if service i is in team t ($x_{it} = 1$), or not ($x_{it} = 0$).

α_s penalty for undesirable shift type s.

n_{dip} number of ORs used by service i in period p of day d.

γ_i minimum ratio of fixed nurses to ORs for service i.

q minimum acceptable value of ratio of all nurses (fixed and float) to ORs over teams, days, and day periods.

η_p total number of nurses needed in night period p.

b_s number of hours in shift template s.

a_{ps} indicates if shift template s covers period p ($a_{ps} = 1$), or not ($a_{ps} = 0$).

h_L lower bound on the number of hours an FTE works per week.

h_U upper bound on the number of hours an FTE works per week.

Decision Variables:

z_{gt} indicates if nurse g is float in team t ($z_{gt} = 1$), or not ($z_{gt} = 0$).

$\bar{z}_{gi}$ indicates if nurse g is fixed in service i ($\bar{z}_{gi} = 1$), or not ($\bar{z}_{gi} = 0$).

λ_{dgts} indicates if nurse g in team t has float shift template s on day d ($\lambda_{dgts} = 1$), or not ($\lambda_{dgts} = 0$).

$\bar{\lambda}_{dgis}$ indicates if nurse g in service i has fixed shift type s on day d ($\bar{\lambda}_{dgis} = 1$), or not ($\bar{\lambda}_{dgis} = 0$).

Q minimum ratio of nurses to ORs over teams, days, and day periods.

Mathematical Model:

$$max\ Q - \alpha_s \sum_{s \in \bar{\mathbb{S}}} \left(\sum_{\forall d,g,t} \lambda_{dgts} + \sum_{\forall d,g,i} \bar{\lambda}_{dgis} \right) \tag{3.9}$$

$$s.t.\ \sum_{\forall g,s} a_{ps} \lambda_{dgts} + \sum_{\forall g,i,s} a_{ps} x_{it} \bar{\lambda}_{dgis} \geq \left(\sum_{\forall i} x_{it} n_{dip} \right) Q \qquad \forall d, t, p \in \mathbb{P} \tag{3.10}$$

$$\sum_{\forall g,s,t} a_{ps} \lambda_{dgts} + \sum_{\forall g,i,s} a_{ps} \bar{\lambda}_{dgis} = \eta_p \qquad \forall d, p \in \bar{\mathbb{P}} \tag{3.11}$$

$$\sum_{\forall g,s} a_{ps} \bar{\lambda}_{dgis} \geq \gamma_i n_{dip} \qquad \forall d, i, p \in \mathbb{P} \tag{3.12}$$

$$\sum_{\forall t} z_{gt} + \sum_{\forall i} \bar{z}_{gi} = 1 \qquad \forall g \qquad (3.13)$$

$$\lambda_{dgts} \leq z_{gt} \qquad \forall d, g, s, t \qquad (3.14)$$

$$\bar{\lambda}_{dgis} \leq \bar{z}_{gi} \qquad \forall d, i, g, t \qquad (3.15)$$

$$\sum_{\forall t,s} \lambda_{dgts} + \sum_{\forall i,s} \bar{\lambda}_{dgis} \leq 1 \qquad \forall d, g \qquad (3.16)$$

$$h_L \leq \sum_{\forall d,s,t} b_s \lambda_{dgts} + \sum_{\forall d,i,s} b_s \bar{\lambda}_{dgis} \leq h_U \qquad \forall g \qquad (3.17)$$

$$z_{gt}, \bar{z}_{gi}, \lambda_{dgts}, \bar{\lambda}_{dgis} \in \{0,1\}; Q \geq q \qquad \forall d, g, i, s, t \qquad (3.18)$$

The objective (3.9) maximizes the minimum ratio of nurses to ORs by day by team by day period, and the second term minimizes the penalty for using undesirable shifts. Constraints (3.10) determine the minimum ratio of nurses to ORs by day by team by day period. The left hand side determines the number of nurses available, while the right hand side is the product of the number of ORs and the minimum ratio. Constraints (3.11) makes sure the total number of nurses in night periods is the required number. Constraints (3.12) makes sure the total number of fixed nurses will satisfy the required nurse-to-OR ratio for each service at each day period. Constraints (3.13) ensure that each nurse can either be fixed or float. Constraints (3.14)-(3.15) make sure that float nurses are assigned float shifts, and fixed nurses are assigned fixed shifts. Note: under these constraints we make sure that a nurse is either always assigned to a service, or always assigned to a team float pool, never both. Constraints (3.16) makes sure each nurse will not have more than one shift on each day. Constraint (3.17) makes sure that each nurse works no more than h_U hours and no less than h_L hours per week. Constraints (3.18) restrict decision variables while $Q \geq q$ means the minimum ratio of nurses to ORs is no less than q across teams, days and day periods. In the remainder of this chapter we refer to this model as MIP[Shift].

To reduce computational effort, we can further add the following symmetry eliminating constraints to MIP[Shift].

$$\sum_{i=1}^{g} \bar{z}_{gi} = 1 \qquad \forall g = 1, \ldots, S \tag{3.19}$$

$$\bar{z}_{g+1,i} + \sum_{j>i} \bar{z}_{gj} + \sum_{t} z_{gt} \leq 1 \qquad \forall i, g = 1, \ldots, G-1 \tag{3.20}$$

$$z_{g+1,t} + \sum_{k>t} z_{gk} \leq 1 \qquad \forall t, g = 1, \ldots, G-1 \tag{3.21}$$

Constraints (3.19) ensure that the first nurse is assigned to the first service, the second nurse is assigned to one of the first two services, the third nurse is assigned to one of the first three services, and so on. Constraints (3.20) make sure that nurses are assigned to services in order (i.e., lower indexed nurses are assigned to lower indexed services, and vice versa), and that lower indexed nurses will be fixed, and higher index nurses will be float. Constraints (3.21) make sure that nurses are assigned to teams in order (i.e., lower indexed nurses are assigned to lower indexed teams, and vice versa).

Figure 3.1 describes the interaction between the two stages of our model. In the first stage teams are designed, creating service-to-team assignments. These assignments are fed into the second stage, where shifts are designed to cover surgical demand.

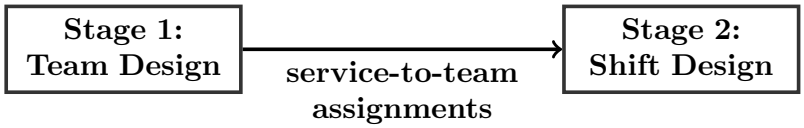


Figure 3.1: Two stage solution approach to service group team design and shift design and allocation.

3.3.3 Approximation Method for the Shift Design and Allocation Problem

The shift design and allocation problem can be computationally challenging for large, realistic instances, even with the addition of the symmetry eliminating constraints, and at hospitals solutions might need to be obtained quickly. To make this problem more tractable, it can be decomposed it into two phases as shown in Figure

3.2. In the first phase, fixed shifts are assigned to each service, making sure that the nurse-to-OR ratio is met, while minimizing the penalty for the usage of undesirable shift types and the number of nurses assigned (to maximize the number of nurses left for the float pools). With these decisions fixed, in the second phase, we allocate the remaining nurses, all of whom will have float shifts, to the teams, while making sure that the allocation is balanced across day hours and minimizing the penalty of undesirable shift types.

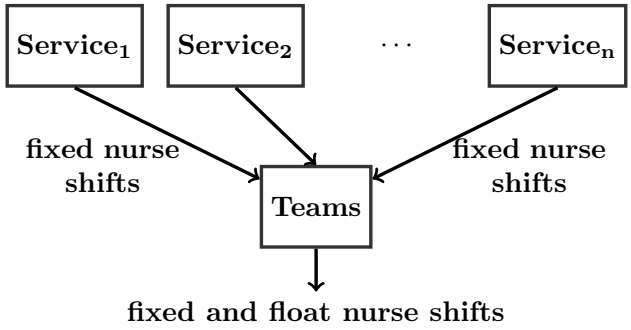


Figure 3.2: Heuristic approach to shift design and allocation: first assign fixed shifts to services, then assign the remaining shifts that will be float shifts to teams.

The following is a detailed description of the approximation method.

3.3.3.1 Shift Design and Allocation Phase 1: Assignment of Fixed Nurses

The first phase considers allocation and assignment decisions of fixed nurses for a given service i. The model can be expressed as follows.

Mathematical Model for Phase 1:

$$min \sum_{\forall g} \bar{z}_{gi} + \alpha \sum_{s \in \bar{\mathbb{S}}} \left(\sum_{\forall g,d} \bar{\lambda}_{dgis} \right) \tag{3.22}$$

$$s.t. \sum_{\forall g,s} a_{ps} \bar{\lambda}_{dgisd} \geq \gamma_i n_{dip} \qquad \forall d, p \in \mathbb{P} \tag{3.23}$$

$$\bar{z}_{gi} \leq 1 - \sum_{j=1}^{i-1} \bar{z}_{gj} \qquad \forall g \tag{3.24}$$

$$\bar{\lambda}_{dgis} \leq \bar{z}_{gi} \qquad \forall d, g, s \tag{3.25}$$

$$\sum_{\forall s} \bar{\lambda}_{dgis} \leq 1 \qquad \forall g, d \qquad (3.26)$$

$$h_L \bar{z}_{gi} \leq \sum_{\forall d,s} b_s \bar{\lambda}_{dgis} \leq h_U \bar{z}_{gi} \qquad \forall g \qquad (3.27)$$

$$\bar{z}_{gi}, \bar{\lambda}_{dgis} \in \{0, 1\} \qquad \forall d, g, s \qquad (3.28)$$

The objective (3.22) minimizes the number of fixed nurses assigned to service i and the use of undesirable shift templates. Constraints (3.23) enforce the nurse-to-OR ratio given the number of ORs needed in each period for the service. Constraints (3.24) make sure that only those nurses are considered for service i that have not been previously assigned to another service. Constraints (3.25) make sure that a nurse cannot have a shift in service i unless they were assigned to that service. Constraints (3.26) make sure that each nurse has a single shift each day, while constraints (3.27) ensure that an assigned nurse work no more than h_U hours, and no less than h_L hours per week.

Model (3.22) can be solved sequentially for one service at each time to find the service allocation and shift assignment of fixed nurses for all services, i.e., for each instance the number of times this model is solved equals the number of services considered. This solution is passed to the second phase as an input. As long as there is a sufficient number of nurses to cover demand, i.e., the model is feasible for both phases, the sequence in which services are considered by model (3.22) has no impact on the solution, as services are considered independently.

3.3.3.2 Shift Design and Allocation Phase 2: Distribution of Float Nurses

The second phase considers team allocation and shift assignment decisions of float nurses in one step, i.e., this model is only solved once for each instance. The following is a description of the model. Note that many of the constraints are identical to those found in MIP[Shift], but the order of the terms was changed to highlight the difference between variables presented on the left hand side, and constants presented

on the right hand side. Moreover, in this phase we only consider the nurses that have not been assigned yet, indexed by g, while the index of nurses already assigned is g'.

Mathematical Model for Phase 2:

$$max\ Q - \alpha \sum_{s \in \bar{\mathbb{S}}} \left(\sum_{\forall g,t,d} \lambda_{dgts} \right) \tag{3.29}$$

$$s.t.\ \left(\sum_{\forall i} x_{it} n_{dip} \right) Q - \sum_{\forall g,s} a_{ps} \lambda_{dgts} \leq \sum_{\forall g',i,s} a_{ps} x_{it} \bar{\lambda}_{dg'is} \qquad \forall d,t,p \in \mathbb{P} \tag{3.30}$$

$$\sum_{\forall g,s,t} a_{ps} \lambda_{dgts} = \eta_p - \sum_{\forall g',i,s} a_{ps} \bar{\lambda}_{dg'is} \qquad \forall d,p \in \bar{\mathbb{P}} \tag{3.31}$$

$$\sum_{\forall t} z_{gt} = 1 \qquad \forall g \tag{3.32}$$

$$\lambda_{dgts} \leq z_{gt} \qquad \forall g,t,s,t \tag{3.33}$$

$$\sum_{\forall s,t} \lambda_{dgts} \leq 1 \qquad \forall d,g \tag{3.34}$$

$$h_L \leq \sum_{\forall d,t,s} b_s \lambda_{dgts} \leq h_U \qquad \forall g \tag{3.35}$$

$$z_{gt}, \lambda_{dgts} \in \{0,1\}; Q \geq q \qquad \forall d,g,s,t \tag{3.36}$$

The objective (3.29) maximizes the minimum ratio of nurses to workload by day by team by day periods, and minimizes the penalty of undesirable shifts. Constraints (3.30) calculate the minimum ratio of nurses to workload. Constraints (3.31) make sure that the required nurses in night periods is fulfilled. Constraints (3.32) designate each nurse that was not assigned to a service a float nurse. Constraints (3.33) make sure that if a nurse had a float shift in a team, they are assigned to that team. Constraints (3.34) ensure that each nurse can have at most one shift each day. Constraints (3.35) ensure that all float nurses work no more than h_U hours and no less than h_L hours per week. Constraints (3.36) restricts decision variables, while $Q \geq q$ means the minimum ratio of nurses to workload is no less than q across teams, days and day periods.

The following is the description of the heuristic.

Number of available nurses to allocate: G

for *($i = 1, \ldots, S$)* **do**
  Solve (3.22)
  g = number of nurses assigned to service i by (3.22)
  $G = G - g$
end

Distribute the remaining G nurses using (3.29)

In the next section we show through a numerical study, that the 2-phase heuristic is an efficient approach to the shift design and allocation problem through a numerical study. We use this method in our hospital data based case study in Section 3.5 to further demonstrate that significant improvements can be achieved using the 2-phase heuristic. In addition to good performance and fast computational time, the 2-phase heuristic has the benefit of being able to address subsets of the staffing problem without affecting the rest of the staffing schedule. For example, if there is significant change in the surgical volume of a service, the first phase can be solved on that single service, and the second phase on the teams, which would not affect the fixed nurse assignments to the rest of the services. Moreover, the 2-phase heuristic achieves greater control over the fixed nurse shifts without the added difficulty of balancing three terms in the objective function, for which weights are challenging to obtain.

3.4 Numerical Analysis of the 2-Phase Heuristic Performance

In this section we evaluate the performance of the 2-phase heuristic through generating random test instances, and comparing solutions of MIP[Shift] to the 2-phase heuristic. Performance is evaluated based on how far away from optimal the heuristic solution is:

$$100\% \times \frac{C^H - C^*}{C^*} \tag{3.37}$$

where C^H is the solution of the 2-phase heuristic and C^* is the solution of MIP[Shift]. We measure average performance and worst case performance.

One of the inputs to the models is the available shift templates that can be staffed. The hospital uses 12 common shift templates for staffing, all of which we consider in our case study in the next section. But to ensure computational tractability for this analysis, we only consider 8-hour shifts from the available shift templates. The shifts considered are listed in Table 3.1. We further defined the three shift templates on the right as undesirable (11AM - 7PM, 3PM - 11PM, 11PM - 7AM), as they fall outside regular work hours.

Hours	Duration (hrs)	Hours	Duration (hrs)
7AM - 3PM	8	11AM - 7PM	8
9AM - 5PM	8	3PM - 11PM	8
		11PM - 7AM	8

Table 3.1: Shift templates considered in the heuristic performance analysis.

To test the robustness of the 2-phase heuristic, we tested randomly generated instances, and varied problem parameters. One parameter to be varied is the number of services. We considered three scenarios: small (3 services, 1 team), medium (5 services, 2 teams) and large (8 services, 2 and 3 teams). Services were randomly sampled with equal probability from the available services at the hospital, and they were randomly assigned to teams when applicable, i.e., there is more than one team. Each scenario was tested on 30 test instances. Table 3.2 describes the scenarios in more detail in terms of number of services per team.

We also varied weights in the objective function. The weight of Q, the minimum ratio of nurses to workload over teams was kept one for all instances, and the penalty of undesirable shifts was 0.0, 0.001 and 0.002. We ensured that service assignments

to teams were unique when parameters were the same.

		Medium		Large		
		Instance 1	Instance 2	Instance 1	Instance 2	Instance 3
2 available teams	Team 1	2	1	4	3	2
	Team 2	3	4	4	5	6
3 available teams	Team 1	-	-	2	2	-
	Team 2	-	-	3	2	-
	Team 3	-	-	3	4	-

Table 3.2: Heuristic performance analysis setup. Note that only large scenarios have three teams, and we consider two ways to distribute services to three teams.

In this study we consider a limited number of shift template choices, i.e., 8-hour shifts only, which necessitates adjustments to problem parameters to ensure feasibility of test instances, such as the number of available FTEs, and the number of nurses needed during night periods. The current number of FTEs at the hospital 138, which we adjust to $138 \times 12/8 = 207$, to account for the fact that we do not consider shifts longer than 8 hours, e.g., 12-hour shifts. Thus the large scenario has 207 available FTEs. For the other scenarios, we assign FTEs relative to the scenario's size in terms of number of services considered. Therefore, the small scenario type has $207 \times 3/8 = 78$ FTEs, and the medium scenario has $207 \times 5/8 = 129$ available FTEs.

The number of nurses needed during night periods was similarly adjusted based on scenario size, shown in Table 3.3. The large scenario is consistent with current hospital practice. We also changed constraint (3.11) in MIP[Shift] and constraint (3.31) in the decomposition heuristic into less than or equal to constraints, to avoid infeasibility due to the way 8-hour shifts cover the day.

Table 3.4 presents the computational results. To further ensure tractability, the optimality tolerance was set to 3% for all instances. Across 270 tested instances, the 2-phase heuristic solution is on average 4.61% away from the optimal solution, and

Scenario	Night period		
	7PM - 9PM	9PM - 11PM	11PM - 7AM
Small	8	3	2
Medium	13	5	3
Large	20	8	4

Table 3.3: Number of nurses needed during night periods by scenario.

at most 16.20% away from optimal. Note that the average performance results are based on setting of optimality tolerance to 3% when solving the models. Thus the actual performance could be better. Moreover, this is the reason why performance can be negative. The results of Table 3.4 show, that as the undesirable shift penalty increases, 2-phase heuristic performance deteriorates. Moreover, the 2-phase heuristic tends to perform better for smaller instances rather than large instances. However, the 2-phase heuristic computational time is significantly lower, especially for larger instances.

Penalty	Scenario	# of teams	# of services	# of nurses	# of instances	Performance		CPU seconds	
						Avg	Max	MIP[Shift]	2-phase
0	Small	1	3	78	30	-0.39%	1.51%	2.64	0.44
	Medium	2	5	129	30	0.28%	1.69%	22.52	1.42
	Large	2	8	207	18	0.20%	1.58%	207.08	2.73
	Large	3	8	207	12	0.85%	1.59%	396.44	4.13
0.001	Small	1	3	78	30	2.41%	4.51%	2.25	0.45
	Medium	2	5	129	30	4.98%	7.30%	30.60	1.90
	Large	2	8	207	18	6.95%	7.93%	329.93	3.28
	Large	3	8	207	12	7.33%	8.42%	416.06	5.46
0.002	Small	1	3	78	30	4.76%	9.56%	2.55	0.46
	Medium	2	5	129	30	8.20%	12.25%	32.40	1.89
	Large	2	8	207	18	13.66%	16.20%	329.81	3.36
	Large	3	8	207	12	13.67%	15.41%	456.93	5.30
Overall		[1, 3]	[3, 8]	[78, 207]	270	4.61%	16.20%	124.54	2.01

Table 3.4: 2-phase heuristic performance compared to MIP[Shift]. Parameters varied include the penalty of undesirable shifts, and scenario size in terms of number of services considered.

3.5 Hospital Case Study Results

The following case study is based on data provided to us by our partner hospital, where they currently have 11 services grouped into three teams; 32 ORs, 4 of which were only recently opened, making surgical volume predictions challenging; and 138 FTEs of surgical nurses. We have analyzed data provided by our partner hospital, including their block schedule and historical surgical volume by service by hour by day, and used the models described in Sections 3.3.1 and 3.3.2 to obtain computational results. In this section we highlight the model outputs and compare their performance to the hospital.

3.5.1 Service Group Team Design Results

The goal of MIP[Team] is to combine services into teams, in order to take advantage of skill flexibility in the next stage. The following 11 services were considered: acute care surgery (ACS), dental (DENT), general surgery (GSA), obstetrics and gynecology (GYN), neurosurgery (NEURO), oral surgery (ORAL), orthopaedics (ORTHO), otolaryngology (OTO), plastic surgery (PLASTICS), transplant (TPT), urology surgery (URO). Due to the similarity and the low block time allocated to the ORAL and DENT services, they were combined.

The main inputs to MIP[Team] are the data that related to the three components of the objective function: (1) the training time matrix, that defines the amount of training needed in weeks to learn skills in different services, and is used as a measure of difficulty of different services; (2) the list of services that need a specialized on-call group at night, and the overnight call volume for each service; and (3) the block schedule of the ORs that assigns rooms to services on weekdays. Moreover, the number of overnight call groups was set to 6, and the number of teams was set to 3 to match hospital needs, based on a long and complex decision process involving hospital personnel.

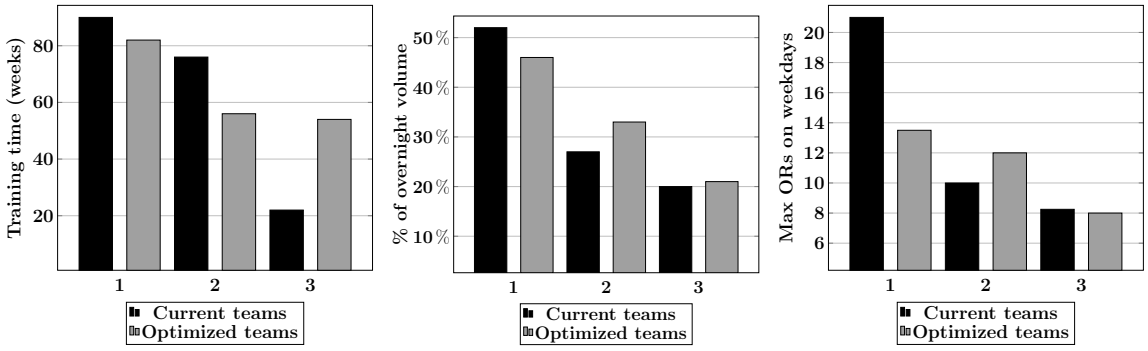
We analyzed several instances of MIP[Team], varying the weight associated with different terms in the objective: the weight for the maximum number of weeks of training across teams (c_W), the weight for the maximum percent of overnight surgical volume across teams (c_U), and the weight for the maximum number of ORs by day across teams (c_R). Instances were run with $c_U \in \{0, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2, 3, 4\}$ and $c_W = 1$. We also allowed $c_U = 1$ with $c_W = 0$ to reflect situations in which we do not want to incentivize similar training times across teams. In all these 12 instances $c_R = 0.1$, to reflect the lower emphasis on the maximum number of ORs by day expressed by decision makers worked with. In general, the overall weight structure reflects the importance of the different components of the objective function. We tested a diverse set of cost parameters, but in the majority of the test cases the components with the highest weights are the ones that are associated with fairness to nurses.

We found that varying the cost parameters does not have a significant impact on the team assignment. The proposed team assignment, which is shown in Table 3.5, was chosen because it resulted in the optimal solution in most instances (7 out of the 12), and this solution was at most 17% away from the optimal solution in the instances it did not achieve the optimal solution, with an average gap of 9.4%. The chosen solution was furthest away from the optimal solutions in the instances where either c_U or c_W was set to 0. If we exclude these instances, the maximum gap is reduced to 9.7% with an average gap of 5%.

Current teams			Optimized teams		
Team 1	Team 2	Team 3	Team 1	Team 2	Team 3
DENT	GSA1	ACS	ACS	ORTHO	GSA
NEURO	ORTHO	GSA2	NEURO	OTO	GYN
ORAL	TPT	GYN	ORAL & DENT	PLASTICS	URO
OTO		URO			TPT
PLASTICS					

Table 3.5: Current and optimized service group teams.

To understand how the optimized teams perform compared to the current teams, we looked at the three performance metrics considered: the maximum training time, the maximum percent of overnight surgical volume, and the maximum number of ORs per team by day. Figure 3.3 shows each of the performance metrics. Note that the emphasis of the model is to reduce variation across teams. To make the pattern clear, the results are presented in decreasing order for the current teams and optimized teams separately. Thus the first column corresponds to the highest value of the performance metric, and not to team 1.



(a) (b) (c)

Figure 3.3: Comparison of current and optimized teams in terms of: (a) weeks of training, (b) percent of overnight surgeries, and (c) maximum number of ORs over week days.

Table 3.6 further highlights the improvement that is achieved through the optimized teams. The results show that the spread of the performance metrics, defined as the difference between the maximum and minimum over the teams, decreased in each case by at least 22%, and by as much as 59%. Table 3.6 also shows that the average of the performance metrics is about the same in the optimized teams as in the current teams, which is not surprising, as the objective was not the reduction of the average, but to create more balanced teams, i.e., reduce the spread.

Performance metric	Current teams		Optimized teams		% improvement	
	Average	Spread	Average	Spread	Average	Spread
Weeks of training	62.7	68	64	28	-2%	59%
% of overnight volume	33.3%	32.1%	33.3%	25.2%	0%	22%
Max ORs on weekdays	13.1	12.75	11.2	5.5	15%	57%

Table 3.6: Average and spread (maximum minus minimum) of performance metrics for current and optimized teams, and percent improvement of optimized teams over current teams.

3.5.2 Shift Design and Allocation Results

The purpose of the 2-phase heuristic is to allocate shifts to services and to the teams created in stage 1. The data we used for this problem includes the optimized teams, i.e., assignment of services to teams as decided by MIP[Team]; total nurse hours available, i.e., FTEs, the number of ORs needed by period by day for each service; and the required nurse-to-OR ratio. We enforce all available nurses to be fully allocated. Consistent with current practice, the minimum nurse-to-OR ratio, q, was considered to be 2 in all cases, while the fixed nurse-to-OR ratio was 1. The minimum work hours of an FTE per week, h_L, was set to 36, while the maximum, h_U, was set to 40. Moreover, there are 12 common shift types used at our partner hospital, which are shown in Table 3.7.

#	Hours	Duration (hrs)	#	Hours	Duration (hrs)
1	7AM - 3PM	8	7	11AM - 7PM	8
2	7AM - 5PM	10	8	11AM - 9PM	10
3	7AM - 7PM	12	9	11AM - 11PM	12
4	9AM - 5PM	8	10	3PM - 11PM	8
5	9AM - 7PM	10	11	11PM - 7AM	8
6	9AM - 9PM	12	12	7PM - 7AM	12

Table 3.7: Shift types considered in the shift design and allocation model.

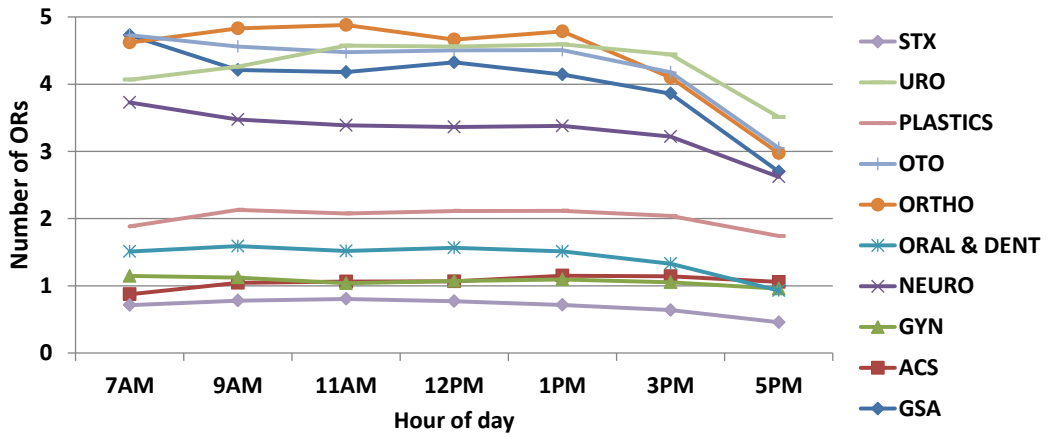


Figure 3.4: Average OR usage of surgical services by hour of day.

One year of historical surgical volume data was used together with the block schedule to form the model input n_{dip}, the number of ORs used by service i in period p on day d. Figure 3.4 shows the OR usage of services by hour of day from historical data. To see how historical data is combined with the block schedule to calculate n_{idp}, let us consider the URO service as an example. As Table 3.8 shows, we used data about the number of ORs running by time of day, and the percent of the running ORs that belong to each service. For example, we calculated the number of ORs URO has in the 2 hour period between 7AM and 9AM to be 14.53% of 28 ORs, for a total of 4.07 ORs. We did this for intervals covering the day. Finally, since URO received 10.5 additional hours on Mondays when the 4 new ORs were opened, we add 1 to the appropriate periods to get the results in Table 3.8. If the new OR was designated an *open* room (i.e., not assigned to a service, reserved for add-on cases), then the additional OR time was evenly distributed across all services. Once the total number of ORs is calculated for a service, it is compared to the block schedule, and if there is significant deviation compared to the block schedule, the number is adjusted the following way. If the total number of ORs plus half and OR is less than the number indicated in the block schedule, then the number of ORs is set to that indicated in the block schedule. If the total number of ORs exceeds the number indicated in the block schedule by at least one OR, the total number of ORs is decreased by one. Note that service specific assignments only apply to day periods, hence Table 3.8 only covers day

periods. Night periods, the time between 7PM and 7AM, are staffed with float nurses.

Period start hour	7AM	9AM	11AM	12PM	1PM	3PM	5PM
Period end hour	9AM	11AM	12PM	1PM	3PM	5PM	7PM
Total running ORs (#)	28	28	28	28	28	26	20
URO usage (%)	14.53	13.36	14.24	13.55	13.85	16.69	16.49
URO ORs (#)	4.07	3.74	3.99	3.79	3.88	4.34	3.30
New ORs (#)	1	1	1	1	1	1	0
Total URO ORs (#)	5.07	4.74	4.99	4.79	4.88	5.34	3.30

Table 3.8: Example calculation for the number of ORs used for the URO service on Monday.

To understand how the optimized shift design performs, we analyzed one month of data for weekdays, i.e., 20 days. We separated every day into 15 minute intervals and measured the *coverage* of surgical demand, which is positive if there were more than enough nurses to cover each surgery (overstaffed), or negative if there were too few nurses to cover each surgery (understaffed).

The 2-phase heuristic objective is to maximize the minimum number of nurses available over day periods for all teams across all days, and minimize the penalty of using undesirable shifts, while also minimizing the number of fixed nurses. We considered two scenarios in this stage. In the first scenario we used the penalties for undesirable shifts to control the *shift mix*, i.e., the percent utilization of shifts by duration and by type. This ensured that the optimized shift mix was similar to the current shift mix. In the first scenario we used $\alpha_1 = 0.1$ for shifts that end at 5PM, because no shift starts at 5PM. Thus, there is no arriving staff to relieve the nurses whose shift ends at 5PM, and nurses often experience mandated overtime. Moreover, a penalty of $\alpha_2 = 0.002$ was used for the usage of 12-hour shifts to control shift mix. These parameter choices allowed for ensuring that optimized shift mix was similar to the current shift mix. In the second scenario we allowed deviations from current shift mix, and only penalized for the usage of shifts that end at 5PM, due to the challenges

staffing this shift presents to nurse managers. Moreover, instances were solved with a 5% optimality gap. We examined these two scenarios, because the first facilitates easier implementation due to similar shift mix, and wanted to evaluate how limiting certain shift types impacts coverage of surgical demand. Moreover, the second scenario can be used to inform hiring decisions.

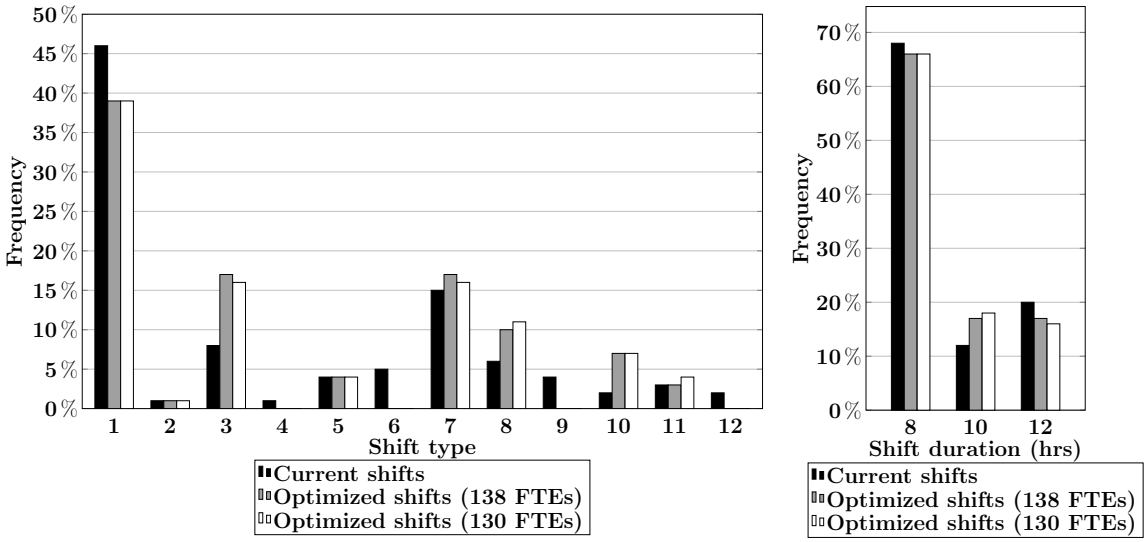
Scenarios 1 and 2 both considered two cases. In case 1 the current number of FTEs available at the hospital is respected, which was 138 for the period we studied. In case 2 we allowed the number of available FTEs to be reduced.

3.5.2.1 Scenario 1: Respecting Current Shift Mix

We first analyze results in the first scenario, where we strive to respect the current shift mix at the hospital, controlling the chosen shifts by the penalty α_s. Model outputs that respect current shift mix can aid implementation, as fewer changes will be required to the nurse schedule.

Figure 3.5 shows the current and the optimized shift mixes both in terms of shift types and shift durations for the two cases: when the number of FTEs is the same as at the hospital, namely 138, or the number of FTEs is reduced to 130. The graphs show that in the optimized shifts the shift mix is close to the current shift mix for both shift types and shift durations. In fact, each shift type frequency is within 8% of the current values for both cases, and within 6% for shift duration.

To measure the performance of the optimized shifts, we evaluate their coverage of surgical demand. Table 3.9 shows the relative improvement of coverage of surgical demand of the two cases over current state, where the cases are defined by the number of FTEs used. From the table we can see that average coverage and standard deviation of coverage show improvements in both cases. Note that as expected, the more FTEs are available, the more we can improve on nurse shortfall, while with more nurses the improvements in nurse excess decrease, as there is more opportunity for nurses to be



(a) (b)

Figure 3.5: Optimized shift mix under scenario 1, where current shift mix is respected: (a) by shift type and (b) by shift duration.

idle. From the patient point of view, avoiding understaffed periods is more desirable; however, from the hospitals point of view, avoiding overstaffed periods can present financial savings.

(a) Understaffed periods

	Case 1	Case 2
FTEs	138	130
Avg nurse shortfall	44%	4%
Stdev	45%	4%

(b) Overstaffed periods

	Case 1	Case 2
FTEs	138	130
Avg nurse excess	1%	12%
Stdev	13%	19%

Table 3.9: Relative improvement of coverage over current state with optimized teams in scenario 1. Case 1 is representative of the current state with 138 FTEs, while case 2 considers only 130 FTEs.

We recognize that changing the team structure can be challenging at hospitals, and requires a major investment of time and resources. Thus we also compare model performance to current teams in Table 3.10, and show that improvements in terms of coverage of surgical demand can be achieved without changing the team structure,

but only changing shift allocation and design.

(a) Understaffed periods

	Case 1	Case 2
FTEs	138	130
Avg nurse shortfall	62%	17%
Stdev	36%	25%

(b) Overstaffed periods

	Case 1	Case 2
FTEs	138	130
Avg nurse excess	1%	12%
Stdev	12%	16%

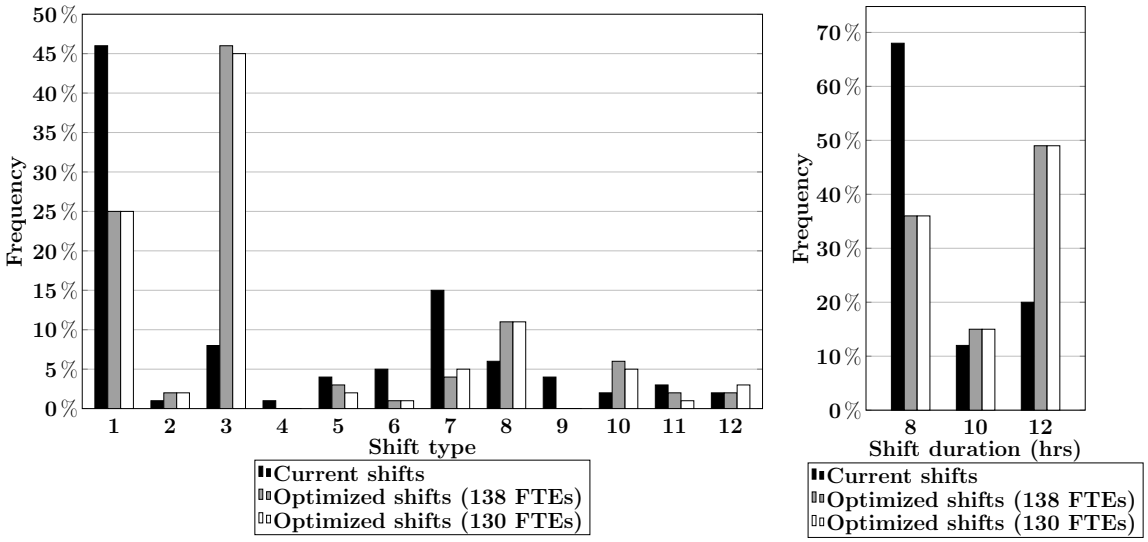
Table 3.10: Relative improvement of coverage over current state with current teams in scenario 1. Case 1 is representative of the current state with 138 FTEs, while case 2 considers only 130 FTEs.

3.5.2.2 Scenario 2: Deviating from Current Shift Mix

In the second scenario we analyzed the benefit of allowing larger deviations from the current shift mix. Respecting current shift mix can aid implementation with the current nurses that are on staff. However, due to high nurse turnover, new nurses frequently join hospital surgical services. Therefore the results of this scenario are useful for informing managers about ideal long-term targets that could be achieved through decisions about which shifts to add over time.

Figure 3.6 shows the current and the optimized shift mixes both in terms of shift types and shift durations for the two cases described earlier: when the number of FTEs is the same as at the hospital, namely 138, or the number of FTEs is reduced to 130. As expected, the graphs show much larger deviation from shift mix than in scenario 1. Shift types can deviate by as much as 38%, while shift durations can deviate by as much as 32%. Note that in this scenario, the model suggests more use of 12-hour shifts and less use of 8-hour shifts than currently used at the hospital. This observation was true for both cases.

Table 3.11 shows the relative improvement of coverage of surgical demand of the two cases over current state. In addition to seeing improvements over the current



(a) (b)

Figure 3.6: Optimized shift mix under scenario 2, where deviation from current shift mix is allowed: (a) by shift type and (b) by shift duration.

state, we also see improved results over scenario 1, except for the average and standard deviation of nurse shortfall in case 1, every other performance metric is an improvement over scenario 1. Thus we conclude that it is beneficial to allow deviations from current shift mix if hospital constraints allow for it.

(a) Understaffed periods

	Case 1	Case 2
FTEs	138	130
Avg nurse shortfall	33%	5%
Stdev	31%	15%

(b) Overstaffed periods

	Case 1	Case 2
FTEs	138	130
Avg nurse excess	5%	15%
Stdev	30%	30%

Table 3.11: Relative improvement of coverage over current state with optimized teams in scenario 2. Case 1 is representative of the current state with 138 FTEs, while case 2 considers only 130 FTEs.

We again compare model performance to that with current teams in Table 3.12, to show improvements in terms of demand coverage can be achieved with current teams. Average nurse shortfall in case 1 is the only performance metric with a larger

improvement in scenario 1. Similarly to the optimized teams, here we see greater improvements compared to the first scenario, in which we respected the hospital's shift mix: the mean improved by 17% on average, and the standard deviation by 49%.

(a) Understaffed periods

	Case 1	Case 2
FTEs	138	130
Avg nurse shortfall	34%	28%
Stdev	36%	27%

(b) Overstaffed periods

	Case 1	Case 2
FTEs	138	130
Avg nurse excess	4%	14%
Stdev	29%	29%

Table 3.12: Relative improvement of coverage over current state with current teams in scenario 2. Case 1 is representative of the current state with 138 FTEs, while case 2 considers only 130 FTEs.

3.6 Conclusions

We proposed a two-stage approach to planning level decisions for surgical nurse staffing. In the first stage, surgical services are grouped together into teams with the objective of balancing training time, overnight surgical volume, and the maximum number of ORs over weekdays across teams. In the second stage, we allocated shifts to the newly designed teams with the objective of balancing the number of nurses per ORs during the day, and minimizing the number of undesirable shifts that are used to cover demand.

In our case study, that is based on hospital data, we found that the optimized teams improve the fairness of teams by improving the spread, i.e., the difference between the maximum and the minimum across teams, and thus make the teams more balanced in terms of each of the performance metrics: training time by 59%, overnight surgical volume by 22%, and maximum ORs over weekdays by 57%.

Furthermore, we showed that optimized shifts achieve better coverage of surgical

demand both for the currently used and for optimized teams: average nurse shortfall can be improved by at least 33%, and average nurse excess by 1% with the current number of FTEs, while average nurse shortfall can be improved by at least 4% and average nurse access by at least 12% with a reduced number of FTEs. We also identified the benefit of deviating from the current shift mix, and showed that performance can be improved even with less FTEs than are currently employed at the hospital.

The fact that performance can be improved with fewer FTEs is important, as this fact can aid implementation of this methodology for the following reason. Our methods generate planning level models, that allocated shifts to days, services and teams. As a last step to generate a complete staffing schedule, charge nurses must assign specific nurses to the recommended shifts. Each nurse has their own predefined skill set and a set of shift types they are willing to work, which makes fitting nurses into the optimized shift schedules a challenging task. If, however, the shifts corresponding to 130 FTEs are predetermined, and the rest of the shifts corresponding to the remaining 8 FTEs can be allocated based on skill specifications and nurse preferences with the assurance of good performance, this is likely to ease the implementation of the new optimized shifts based on our model.

It remains to be determined if the methodology presented in this chapter can be implemented, as it would involve major institutional changes.

3.7 Appendix

3.7.1 Surgical Nurse Training Time Data Collection

One of the crucial inputs to the service group team design model, MIP[Team], is training time, which can also be used as a measure of difficulty of a service. The current training system is set up to train new nurses within the current team structure. Thus training time is only established for services within the prespecified teams.

However, for our model we needed training time input for each service to allow the model to choose the teams.

To find out how long it takes to train people in each service, nurse managers and nurse educators were asked by a hospital employee to independently fill out a training time matrix, where the dimension of the matrix was 10×10 (since we considered 10 services), and entry w_{ij} denoted how many weeks of training a nurse would need to learn service j, if they are already skilled in service i.

Based on the results of the survey, a meeting was called with the key stakeholders so that they would come to consensus on training time. For each entry of the matrix of training time averages, each stakeholder was asked whether or not they supported the result. If the result was not supported, the group discussed the entry and came to consensus on what the value should be. The final training time matrix was then sent out to each key stakeholder, so that even those who could not attend the meeting would have a chance to say if they did not agree with the result, and the matrix was again shown to key stakeholders during follow-up meetings. The agreed upon matrix is shown in Table 3.13.

	GSA	ACS	GYN	NEURO	ORAL & DENT	ORTHO	OTO	PLASTICS	URO	TPT
GSA	0	4	8	12	8	14	8	6	6	4
ACS	4	0	8	12	8	14	8	6	6	6
GYN	8	8	0	12	8	14	8	6	6	8
NEURO	8	8	8	0	8	12	8	6	8	8
ORAL & DENT	8	8	8	12	0	14	4	6	8	8
ORTHO	8	8	8	12	8	0	8	6	8	8
OTO	8	8	8	12	4	14	0	6	8	8
PLASTICS	8	8	8	12	6	14	6	0	8	8
URO	8	8	6	12	8	14	8	8	0	8
TPT	4	4	8	12	8	14	8	8	8	0

Table 3.13: Agreed upon surgical nurse training time by service.

CHAPTER IV

Capacity Reservation Heuristics to Manage Access Delay in Operating Rooms

4.1 Introduction

Operating room (OR) utilization is of great importance, as OR capacity is one of the most expensive resources at hospitals and in the surgery delivery system. In this chapter we devise methods that allow a hospital to schedule a surgery date for patients, so that the access delay is kept to an appropriate time window without causing excessive overtime. Simple heuristics, like first come, first served (FCFS) are often used in practice, but they perform poorly because they do not recognize variation in priority for different patients. Furthermore, competing criteria complicate the selection of the best method for the assignment of surgeries to ORs. On the one hand, an important criterion is to have full utilization of ORs and allocate OR capacity to patients well in advance. On the other hand, high priority patients might arise on short notice, who must be allocated capacity in a short time frame. Thus the stochastic nature of patient requests for surgery poses a great challenge in this problem. The surgical schedule and OR utilization is also impacted by the stochasticity in surgery durations, which can depend on surgical service and surgery type.

As noted in Chapter I, many hospitals use *block booking* when scheduling surgeries. This means that OR capacity is reserved for specific services or surgeons on a

weekly or monthly basis, i.e., services or surgeons have block time allocated to them, and services or surgeons can only schedule surgeries into their own block time. At a high level this is a form of capacity reservation of ORs; however, blocks generally do not consider patient attributes other than surgical service. Moreover, within the block time, surgeries get allocated to specific days based on surgeon and patient preference. This makes surgical schedules unpredictable, because demand for similar surgery types is not evenly distributed across days, and changes from week to week. This can make it difficult to plan for the resources needed after surgery (e.g. PACU beds, ICU beds, or post-surgical ward beds).

Another challenging aspect of assigning surgery dates to patients is the variation in patient needs for surgery. Hospitals typically have three patient urgency levels. The highest priority is emergent patients, who need immediate access to surgery. We refer to such patients as level 1. The second highest priority patients are urgent patients, who need access to surgery in a small time window, but do not necessarily need surgery immediately. We refer to such patients as level 2. The third urgency level is elective patients, who generally can wait days or weeks for their surgery. We refer to such patients as level 3. Hospitals need to quickly accommodate high priority level 1 and 2 patients, which arise on short notice. The booking of low priority level 3 surgeries often happens well in advance, and without careful planning and reservation of capacity, it is unlikely there would be sufficient time available for high priority surgeries that arise on short notice without the use of resource overtime.

We propose a capacity reservation heuristic, where OR capacity is reserved as *surgery slots* (slots for short) within the block schedule for different types and urgency levels of patients. The purpose is to ensure they have access to surgery within their surgery access target (SAT), which is an important strategic level issue for some hospitals. The decision for hospitals on how they choose to allocate their OR capacity, i.e., using surgery slot reservations, FCFS approach, or other methods, is a strategic level decision. However, to test the viability and the performance of the surgery slot

reservation approach, we use a tactical level discrete event simulation model.

In this setting we associate three attributes with patients: the surgical service providing the surgery, surgery type, and urgency level. Surgery type can be defined by procedure type (e.g., hip replacement, appendectomy), or by expected duration (e.g., short surgery, long surgery). We chose the latter definition, because certain procedure types have very low volume, and thus defining type by duration allows for greater flexibility in the use of the reserved slots.

To gain insights into the effects of capacity reservation heuristics, we evaluated each heuristic using a discrete event simulation model. The heuristics can be categorized into two groups: surgery slot reservation heuristics and FCFS based heuristics. In the surgery slot reservation heuristics, surgery slots are reserved for patients according to their attributes, and patients are assigned to the first available slot that corresponds to their attributes: surgical service, surgery type, and urgency level. To avoid wasting OR capacity, and to ensure that high urgency patients have fast access to surgery, reserved slots can also be released to patients with attributes other than those corresponding to the reservation.

We consider three FCFS based heuristics, the first one being the classical FCFS heuristic, where patients are considered in the order they request surgeries, and are assigned to the first day with sufficient capacity within their service's block time, not considering their type or urgency level. Second, we evaluate a priority based FCFS, where capacity is "carved out", i.e., reserved, for patients with a high urgency level (level 1 and 2), and they are guaranteed fast access to surgery through the use of overtime, while patients of urgency level 3 are scheduled according to FCFS. Third, we look at a utilization based FCFS, which only differs from the priority based FCFS in that patients with a low urgency level are assigned to the least utilized day within their SAT, or if no such day is available, they are assigned according to FCFS to the first day with sufficient capacity outside of their SAT.

Our main modeling assumptions are the following. Every patient that requests surgery is immediately assigned a surgery date to avoid the need for a wait list, which may be undesirable to patients. Patient requests for surgery happen once a day, at the beginning of the day, with the consideration of the order the requests arrive in, because our heuristics are daily level, and do not consider hour of day assignments. Moreover, cancellations are not allowed, rather overtime will be used if necessary. This is reasonable because cancellation rates tend to be very low in practice, due to the high impact they have on patients and their families, thus hospitals and physicians strive to keep all appointments. As is the case in some hospitals, we assume that there are no reserved ORs for emergencies. The case in which emergent patients are all operated on in dedicated ORs is an easier problem to solve, and subsumed by the model we study in this chapter, since it corresponds to the case of zero demand for emergent surgeries. We further assume that when patients are scheduled for surgery and are given a surgery date, they accept that surgery date and show up on their scheduled day, i.e., even with long wait times, patients do not seek service at other healthcare institutions. We show that using our methodology, patient wait times are expected to be low, and thus inadequate patient access to surgery is not an issue. As hospitals rarely have planned surgeries on weekends, we only focus on weekdays. We only consider the ORs and ignore supporting resources, such as PACU beds, ICU beds, and post-surgical ward beds. The usage of such resources is highly dependent on case sequences, and methods such as described in Chapter II can be used to plan for these resources once the sequence is available. Moreover, we assume surgical nurses are appropriately staffed to fulfill surgeries. Methods such as discussed in Chapter III can be used to find an appropriate staffing schedule to achieve this goal.

The remainder of this chapter is organized as follows. We review the relevant literature in Section 4.2, and highlight our contributions. In Section 4.3 we describe our proposed heuristics, and in Section 4.4 we discuss the discrete event simulation model. We present two case studies in Section 4.5. First we consider a system with

two identical services, compare heuristic performance in this setting, and analyze sensitivity to some parameters in Section 4.5.1. Then in Section 4.5.2 we analyze heuristic performance in a realistic setting based on hospital data. We present our conclusions in Section 4.6.

4.2 Literature Review

The topic of capacity reservation has been studied in several different settings: railway bookings (*You* (2008)), cruise ships (*Maddah et al.* (2010)), airport runways (*Janic* (2007)), airline seat protection levels (*Van Ryzin and McGill* (2000)), perishable goods (*Jacko* (2016)), and wireless networks (*Heredia-Ureta et al.* (2003)).

Most effort in healthcare in this area focuses on creating the block schedule, i.e., allocate OR time to surgical specialties. *Creemers et al.* (2012) used a bulk service queueing model that feeds into an optimization model to assign OR time to surgical services with the objective of minimizing total expected weighted waiting time of patients. They also proposed a 2-step heuristic for the problem, where in the first step a feasible solution is found, and in the second step remaining capacity is allocated in a greedy fashion. They did not consider OR utilization. *Zhang et al.* (2009) took a mixed integer programming approach to allocating ORs to surgical services, and used simulation to evaluate their allocation scheme. Their approach is more detailed in that they considered emergent patients and two kinds of elective patients: inpatient and outpatient. However, they did not consider surgery types. They assumed that an OR is dedicated to emergent patients, and they did not model overtime of ORs. They applied a penalty for surgeries that were delayed, but elective demand that was not met within a week was considered lost.

In a non-block booking setting, *Gerchak et al.* (1996) considered a stochastic dynamic programming model to schedule elective surgeries in advance, while considering emergency surgeries and the uncertain nature of OR utilization. Their approach was

to consider the current day, and pick the number of elective patients to be assigned to that day from a wait list. Our approach is fundamentally different in that each patient gets an assigned date on the day their request for surgery happens. Moreover, contrary to our approach, they assumed each patient's surgery duration is identically distributed, and considered no surgery types.

Surgery access targets have also been considered. *Astaraky and Patrick* (2015) used a simulation based approximate dynamic programming approach to set future appointment dates, considering their surgery duration and length of stay at the hospital, while meeting waiting time targets for patients of various class. They showed that their model outperforms the FCFS policy in terms of managing patients that are waiting for surgery when there is limited capacity. Moreover, their approach better managed OR capacity and made use of OR overtime to maintain a stable wait list, and it resulted in a higher overall bed utilization. *Gocgun and Ghate* (2012) looked at advanced scheduling, where each patient has a deadline, and their wait time is penalized. They took an MDP approach, established that it is weakly coupled, and developed an approximate dynamic programming approach to solve the problem. However, both these approaches are different in that we propose a surgery slot reservation scheme where patients are assigned to reserved slots. Moreover, we also allow reservations to be released, i.e., made available to other patients.

Dexter et al. (1999) used simulation to evaluate on-line bin packing algorithms for elective surgery planning in terms of the resulting OR utilization. Their recommendation was to allocate block time to surgeons based on their total expected elective surgery durations, and schedule patients into the first available block if one is available within 4 weeks. Otherwise use overtime. Our approach is more general in that we not only consider multiple urgency levels, but also different surgery types, and multiple SATs.

Closest to our approach are the following works. *Vermeulen et al.* (2009) presented

an operational level approach to planning and scheduling of the time of day of surgical appointments in a scheme that employed an urgency-specific capacity reservation plan for CT-scan visits, based on an initial reservation plan. Their main performance metric was based on percent of patients served within SAT for each patient type. They presented a search-based heuristic for the dynamic adjustment of the reservation plan over time, provided there is an initial reservation plan and assuming a single homogeneous set of CT scanners. *Green et al.* (2006a) considered the problem of scheduling outpatient, inpatient and emergent patients for an MRI machine. On a daily level they studied design of appointment scheduling for outpatients through heuristics, and the real-time allocation of higher priority inpatients and emergent patients using a finite horizon dynamic program. Their objective is to maximize the revenue obtained from serving patients, the penalty of patient wait time, and the penalty of not serving patients on the considered day. They assumed that all patients have the same appointment duration distribution. Our goal is to compare strategic level capacity reservation plans at the daily level. We consider a more diverse set of patients that are paired to specific services, while allowing for multiple surgery types and urgency levels. We also consider the mean overtime and undertime of ORs, and also mean wait time of patients for their appointment. Our version of a dynamic adjustment of the reservation is reflected in release policy that is simple and easy to implement.

Kazemian et al. (2017) considered different heuristics for scheduling a clinic appointment that is probabilistically followed by a surgical appointment. The heuristics were variations on the FCFS heuristic, and did not assume any capacity reservations, unlike our heuristic. They also assumed that patients are served within their access target, and overtime is always available when necessary. We allow patients to exceed their SAT to understand the trade-off between overtime and meeting the access target. Their work was applied to a destination hospital, where even elective surgeries need to take place within a few days of the clinic appointment, as patients might have traveled far to be seen at the hospital. We consider a typical hospital setting, where

elective patients can tolerate longer wait times.

4.2.1 Our Contributions to the Literature

Gupta and Denton (2008) highlighted indirect patient wait time, i.e., the number of days a patient waits for their surgery, as one of the open challenges of appointment scheduling, and called for approaches that model both indirect and direct wait time of patients, i.e., patient wait time before the day of surgery and on the day of surgery. Due to the challenges presented by modeling both types of patient wait times, they recommended a two stage approach with first allocating patients to days, and then assigning start times to the appointments in the second stage. We addressed the second stage problem in Chapter II, and we address the first stage problem in this chapter.

Gupta and Denton (2008) discussed three appointment scheduling environments in healthcare: primary care appointment scheduling, specialty clinic appointment scheduling, and elective surgery scheduling. We propose to apply the idea of reserving surgery slots for different types of patients from the primary care setting to making such reservations within the block schedule. For a hospital to choose to follow such a scheduling policy is a strategic level decision; however, to understand the benefits of this approach, we test heuristic performance using a tactical level discrete event simulation model.

We consider a surgical service, surgery type, and urgency level based reservation scheme to ensure that indirect patient wait times do not exceed predefined SATs. Moreover, to mitigate the effect of uncertainty in patient requests for surgery, and to ensure high utilization of ORs, we release certain reserved slots within a prespecified time frame to avoid having available capacity go unused. We test the proposed method on a simple stylized example we created with two identical services, and in a realistic setting where hospital data was used to construct plausible problem instances. We

show that the surgery slot reservation heuristic performs well compared to FCFS based heuristic benchmarks. In addition to the good performance, the surgery slot reservation heuristic also has the added benefit of the predictability of the surgery schedule, which is due to the fact that reserved slots stay constant, unless there are significant changes in surgical demand. This has the potential to aid in the planning of supporting resources.

4.3 Heuristic and Modeling Descriptions

Information we use to characterize patients in our model includes the following: surgery request day, surgical service, surgery type and urgency level. The distribution information of these inputs is assumed to be identical for the days of the week. The list includes data about available block time for each service, surgery duration distribution information (which includes turnover time after surgery), and surgery access target (SAT) for patients. We consider OR capacity reservation as a discrete finite horizon problem on the daily level, where each day requests for surgery arrive at the beginning of the day. However, we take into account the order in which the requests arrive to better match a real time scheduling system.

There are two sources of randomness in the system: patient requests for surgery and surgery duration. Request arrivals for surgery of each service, type and urgency class are assumed to follow a stationary Poisson process (*Green et al.* (2006b), *Zonderland et al.* (2010), *Kazemian et al.* (2017)). This stationary model does not capture the seasonality in demand for some types of surgery, but it could be relaxed for specific applications. When assigning patients to future dates, information about their estimated surgery duration is needed. Due to the uncertainty in surgery durations, it is necessary to pick this estimated surgery duration such that it mitigates the uncertainty in durations. In our models, we choose the estimated surgery duration as a prespecified percentile from the duration distribution that is expected to ensure good performance.

The following is a summary of the notation that we use in the remainder of this chapter.

Indices:

i	index of patients, $i = 1, \ldots, P$.
k	index of surgical services, $k = 1, \ldots, K$.
τ	index of surgery types, $\tau = 1, \ldots, T$.
u	index of urgency levels, $u = 1, 2, 3$.

Inputs:

$\alpha(i)$	day on which the surgery request for patient i arrives.
E_u	minimum acceptable access delay in days after an urgency level u patient's request for surgery arrives.
$x_{(k,\tau)}$	surgery duration estimate of patients of service k, type τ, which is chosen as a percentile from the duration distribution.
p	proportion of block time that is "carved out" in total for level 1 and 2 patients.
B_k	block time allocated to or used by service k in minutes. Note that this input is assumed to be day independent.
$B_{(k,3)}$	block time of service k intended for level 3 patients in minutes, $B_{(k,3)} \leq B_k$.
$SAT_{(k,\tau,u)}$	surgery access target, i.e., number of days in which a patient of service k, type τ, and urgency level u needs to have surgery.
$\theta_{(k,\tau,u)}$	number of surgery slots reserved for service k, type τ, urgency level u patients.
$s_{(k,\tau)}$	total number of available surgery slots for patients of service k and type τ to be distributed across urgency levels.
$\mu_{(k,\tau,u)}$	mean surgery request arrival rate of service k, type τ, and urgency level u patients.

q additional proportion of the mean surgery request arrival rate, used in the calculation of $s_{(k,\tau)}$.

$\bullet(i)$ input $\bullet$, that corresponds to the attributes and characteristics of patient i, i.e., $k(i), \tau(i)$, and $u(i)$ are the service, surgery type, and the urgency level of patient i, respectively.

Decision variables:

$\gamma(i)$ day on which patient i is scheduled to have their surgery.

As described before, the heuristics distinguish between three urgency levels for patients: level 1, level 2, and level 3, with level 1 being the most urgent, and level 3 being the least urgent level. For each level we consider the minimum acceptable access delay for urgency level u, denoted by E_u, which is the first day a patient can be scheduled to after the day their request for surgery was received. For example, $E_1 = 0$ denotes that level 1 patients can be scheduled as early as the day they request surgery. Note that this model is time stationary, i.e., parameters such as the surgery request arrival rate and SAT for the three urgency levels do not change by day of week or across weeks. The model can be extended to allow the parameters to vary by day and week; however, sharper insights can be gained in the stationary setting.

4.3.1 Surgery Slot Reservation Heuristic

In the surgery slot reservation heuristic, the number of slots reserved for each surgical service, surgery type, and urgency level each day are defined by the heuristic. We call this a *template*. Figure 4.1 shows an example of a template for a single day for a service that has two surgery types and three urgency levels. In this example, $\theta_{(1,1,1)} = 2,\ \theta_{(1,1,2)} = 3$, and $\theta_{(1,1,2)} = 8$; while $\theta_{(1,2,1)} = 2,\ \theta_{(1,2,2)} = 2$, and $\theta_{(1,2,2)} = 6$. In our case studies in Section 4.5 we test different templates.

In this setting, patients are considered in the order they requested surgery, and are scheduled to the first day with an available slot that is reserved for their surgery type

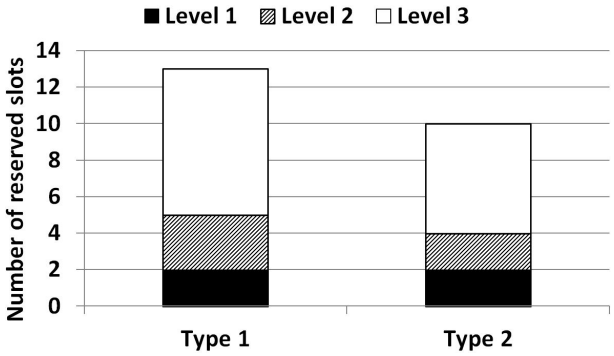


Figure 4.1: A single day example of surgery slot reservations for service 1 with two surgery types and three urgency levels.

and urgency level within their service, starting with day $\alpha(i) + E_u$ for patient i of urgency level u, the earliest acceptable surgery day after the day they requested surgery.

The surgery slot reservation heuristic is a strict reservation policy that could result in wasted OR time if demand does not follow the template closely. To avoid wasting OR capacity, we allow for unused reserved slots to be released, i.e., made available to patients with attributes other than those the slot was reserved for. Due to the short time window within which level 1 and 2 patients need to be seen, it is reasonable not to release those reservations, and only release slots reserved for level 3 patients within the same service. Note that unused level 3 capacity is released across surgery types. For example, the unused capacity of a type 1, level 3 patient can be used by a type 2 and level 1 patient; however, capacity is not shared across services to be consistent with block scheduling rules and the needs of hospital practice.

Unused reserved capacity is released the following way. We must check for the earliest day within the level 1 or 2 patient's SAT, at which there is enough unallocated level 3 capacity to satisfy the request. Suppose that on a given day some of the reserved capacity for level 3 patients, $\theta_{(k,\tau,3)}$ is unused. Also suppose that on that day a request for surgery comes in from a level 1 or 2 patient i. Then reserved slot availability is checked within the SAT of patient i. If there is an available reserved slot within the SAT of patient i that corresponds to their attributes, the patient is

assigned to that slot. If no available slots are found, we check if there is sufficient unused level 3 capacity that can be released to patient i. To check this, first we calculate the block time in minutes, that is available for level 3 patients, as follows:

$$B_{(k,3)} = \sum_{\forall \tau} \theta_{(k,\tau,3)} x_{(k,\tau)}. \tag{4.1}$$

Note that this calculation takes into account time reserved for all surgery types within the same service. We also need to find the time available to be released in minutes, i.e., the time that has not already been allocated to level 3 patients when the request of patient i was received. The right hand side of equation (4.2) represents this quantity, which is the block time that is available to level 3 patients minus the time already allocated to level 3 patients, whose requests were received before the request of patient i. Now for each day d after the earliest acceptable surgery day that is within the SAT of patient i, we check if the estimated surgery duration of patient i, shown on the left hand side of equation (4.2), is less than or equal to the time available to be released, shown on the right hand side of equation (4.2). If this is the case, then patient i is assigned to that day, d. Note that patient i is not assigned to a reserved slot that corresponds to their attributes, i.e., urgency and type, but rather patient i is allocated some of the unused time that is reserved for level 3 patients. If the level 3 time available to be released is not sufficient, then patient i is assigned to the first day with an available reserved slot that corresponds to their attributes, outside of their SAT.

$$x_{(k(i),\tau(i))} \leq B_{(k,3)} - \sum_{\forall j:\, \gamma(j)=d,\, u(j)=3} x_{(k(j),\tau(j))}$$
$$\forall d = \alpha(i) + E_{u(i)}, \ldots, \alpha(i) + E_{u(i)} + SAT_{(k(i),\tau(i),u(i))}. \tag{4.2}$$

4.3.2 FCFS Based Heuristics

In this section we describe three heuristics that were motivated by the FCFS heuristic, which are used as benchmarks to evaluate the performance of the surgery

slot reservation heuristic. In describing these heuristics, we use the term *length of a block* on day d, denoted by $\ell_d(B_k)$, which is the sum of the estimated surgery durations of all patients that have been previously assigned to day d, calculated as follows:

$$\ell_d(B_k) = \sum_{\forall j:\, \gamma(j)=d} x_{(k(j),\tau(j))}. \tag{4.3}$$

We use this notation with the understanding that all allocations prior to the time of calculation are incorporated. Note that the length of block B_k can exceed B_k, which indicates overtime. Let $\ell_d(B_{(k,3)})$ denote the length of the block that is used by level 3 patients on day d. This is calculated as the sum of the estimated surgery durations of level 3 patients that have been previously assigned to day d, shown in the following equation:

$$\ell_d(B_{(k,3)}) = \sum_{\forall j:\, \gamma(j)=d,\, u(j)=3} x_{(k(j),\tau(j))}, \tag{4.4}$$

where $B_{(k,3)} \leq (1-p)B_k$, i.e., level 3 patients are allowed to use up to $(1-p)B_k$ of block B_k.

The first benchmark heuristic we explore is the *classical FCFS* heuristic, where patients are assigned to the first day with sufficient capacity within their service's block time for their estimated surgery duration, starting with their earliest acceptable surgery day after the day they requested surgery. For patient i of urgency u, the first such day would be $\alpha(i)+E_u$. Therefore patient i is only assigned to day $d \geq \alpha(i)+E_u$, if the following inequality holds on that day:

$$B_k - \ell_d(B_k) \geq x_{(k(i),\tau(i))}. \tag{4.5}$$

The second heuristic we propose is the *Carve-out* heuristic. In this heuristic, patients with different urgency levels are allocated according to different rules, where the rules for each urgency level are motivated by how quickly that urgency level

needs access to surgery. Each level 1 patient i is assigned to the day their request for surgery is received, $\alpha(i)$, even if that causes OR overtime. When a request for surgery is received from a level 2 patient, j, on day $\alpha(j)$, capacity is first checked on their surgery request day, $\alpha(j)$, then on day $\alpha(j)+1$. We must check that assigning patient j to one of these two days would not cause overtime based on the estimated surgery durations of patient j and all other patients already assigned to day in consideration. If both days would have overtime with the addition of patient j to that day, then patient j is assigned to the least utilized of day $\alpha(j)$ and $\alpha(j)+1$, defined as follows, even if that causes OR overtime:

$$\underset{\{\alpha(j),\,\alpha(j)+1\}}{\operatorname{argmin}} \left\{ \frac{\ell_{\alpha(j)}(B_k)}{B_k}, \frac{\ell_{\alpha(j)+1}(B_k)}{B_k} \right\}. \tag{4.6}$$

When a level 3 patient, m, arrives, they are assigned to the first available day with sufficient capacity, i.e., to day $d \geq \alpha(m)+E_3$ with both $\ell_d(B_k)+x_{(k(m),\tau(m))} \leq B_k$ and $\ell_d(B_{(k,3)}) + x_{(k(m),\tau(m))} \leq (1-p)B_k$. This heuristic gets its name from the proportion of the block time, p, that is "carved out", i.e., reserved for level 1 and 2 patients.

The final heuristic we developed is named the *Utilization* heuristic. In this heuristic level 1 and level 2 patients are treated exactly the same way as in the Carve-out heuristic. When a level 3 patient, m, arrives, the Utilization heuristic strives to assign the patient to the least utilized day after their earliest acceptable surgery day, but within their SAT, where the least utilized day, d^*, is defined as:

$$d^* = \underset{d=\alpha(m)+E_3,\ldots,\alpha(m)+E_3+SAT_{(k(m),\tau(m),u(m))}}{\operatorname{argmin}} \left\{ \frac{\ell_d(B_{(k,3)})}{B_{(k,3)}} \right\}. \tag{4.7}$$

Patient m is only assigned to day d^* if both $\ell_{d^*}(B_k) + x_{(k(m),\tau(m))} \leq B_k$ and $\ell_{d^*}(B_{(k,3)}) + x_{(k(m),\tau(m))} \leq (1-p)B_k$. This ensures that scheduling patient m to day d^* will not result in overtime, and that the proportion of block time used by level 3 patients does not exceed $1-p$, considering each patient's estimated surgery duration. Each day within the SAT of patient m is checked in increasing order of

utilization. If no suitable day is found, then patient m is assigned to the earliest day outside of their SAT according to the FCFS heuristic, starting the search from day $\alpha(m) + E_3 + SAT_{(k(m),\tau(m),u(m))} + 1$.

4.4 Discrete Event Simulation Implementation Description

This section presents the characteristics of the discrete event simulation, written in C++, that is used to compare the heuristics described in Section 4.3. As mentioned before, the two sources of randomness in the system are the number of requests for surgery by day of each patient class and the surgery durations. Request arrivals for surgery by patient attributes is randomly generated according to a stationary Poisson process (*Green et al.* (2006b), *Zonderland et al.* (2010), *Kazemian et al.* (2017)). The simulation has a discrete (daily) finite time horizon, where patient request for surgery happen at the beginning of each day.

We assume a lognormal distribution for surgery durations (*May et al.* (2000); *Zhou and Dexter* (1998)). Moreover, based on our previous research presented in Chapter II, we use the 60th percentile as an estimate of surgery durations, i.e., $x_{(k,\tau)}$ is the 60th percentile from the surgery duration distribution of a patient in service k and of type τ. This allows for a form of hedging that is intended to reduce the overtime which is likely to be higher if surgeries were scheduled according to the median duration.

For each simulation run, using the same arrival pattern and surgery durations, each of the heuristics described in Section 4.3 were considered: three parameterizations of the surgery slot reservation heuristic (which differed in their templates, i.e., the number of slots allocated by patient attributes), and the three FCFS based heuristics: Utilization, Carve-out, and FCFS. The three surgery slot reservation heuristic templates are described in Table 4.1. The case studies in the following sections show that good performance is achievable with the considered templates. Simulation input includes the total available slots by service and type. Each template is characterized

by the number of slots that level 1 and 2 patients receive, with level 3 patients being allocated the remaining available slots. Thus the template name follows the format: (number of level 1 slots, number of level 2 slots).

Template	Level 1 slots	Level 2 slots
(1,1)	1	1
(1,2)	1	2
(2,2)	2	2

Table 4.1: Templates tested in the surgery slot reservation heuristic.

Given the total number of slots available by surgical service and surgery type, the available block time by service is calculated as the product of available slots and the estimated duration of the surgery, summed over surgery type, described as follows:

$$B_k = \sum_{\forall \tau} s_{(k,\tau)} x_{(k,\tau)}. \tag{4.8}$$

To test cases with different system workload, we can vary the number of available slots, $s_{(k,\tau)}$, in equation (4.8). We determine this quantity by choosing the additional proportion of mean surgery request arrival rate to be considered, where the proportion is denoted by q:

$$s_{(k,\tau)} = (1+q) \sum_{u} \mu_{(k,\tau,u)}. \tag{4.9}$$

To mimic hospital policies, the SAT of level 1 patients is 0, i.e., they need to be served on the day they arrive, and the SAT of level 2 patients is 1, i.e., they need to be served either the day of their arrival or the following day. As a result, level 1 and 2 patients are always scheduled within their SAT in the Utilization and Carve-out heuristics. For level 3 patients there is no standard scheduling policy at many hospitals. *Dexter et al.* (1999) conducted a survey of patients to evaluate what their longest acceptable waiting time is for surgery, and they found that the median was 2 weeks. Therefore we set the SAT of level 3 patients to 10 days, i.e., 2 weeks. Moreover, the

survey of *Dexter et al.* (1999) also included a question about the shortest acceptable waiting time for surgery, and they found that the median was 4 days. This choice is also in line with our SAT parameters, as it is reasonable to set the minimum acceptable access delay for level 3 patients such that it is greater than the SATs of level 1 and level 2 patients. Therefore, each of our policies assume that $E_3 = 4$. For example, if a level 3 patient arrives on Monday, the earliest they have have surgery is the Friday of the same week. We further assume that $E_1 = E_2 = 0$, i.e., level 1 and 2 patients can be scheduled for surgery as early as the day their request for surgery arrives.

We used a warm-up period of a year, after which the following performance metrics are tracked for 3 years:

- mean OR overtime and undertime,
- mean proportion of patients exceeding their SAT,
- mean number of days waited for appointment past SAT, given the SAT was exceeded,
- mean number of days waited for appointment given the SAT was not exceeded.

Note that the heuristics considered used estimated surgery durations when assigning patients to days, where these estimated surgery durations are defined as the 60th percentile from the duration distribution. Once patients are allocated to days, random surgery durations are realized using the discrete event simulation, and performance metrics, such as mean OR overtime and undertime, are calculated using the realized surgery durations (note that even if a heuristic has no overtime in the deterministic setting, expected overtime based on random surgery durations may be greater than zero).

4.5 Performance Analysis

To analyze the performance of the heuristics, we considered two scenarios we describe in this section: a stylized system and an example motivated by our partner

hospital. In the stylized system, we consider identical services, while in the second case study we analyze the heuristics' performance on hospital based data.

4.5.1 Stylized System

In this set of numerical experiments we consider two identical services, where each of the services has one surgery type and three urgency levels. Surgery duration distribution information is given in Table 4.2: the mean, 60th percentile, variance and coefficient of variation of the distributions. Note that we assume that urgency levels have no impact on surgery durations.

Type	Service 1				Service 2			
	Mean	60th	Variance	CV	Mean	60th	Variance	CV
1	100	87	10000	1	100	87	10000	1

Table 4.2: Stylized system surgery duration distribution information: mean, 60th percentile, variance and coefficient of variation.

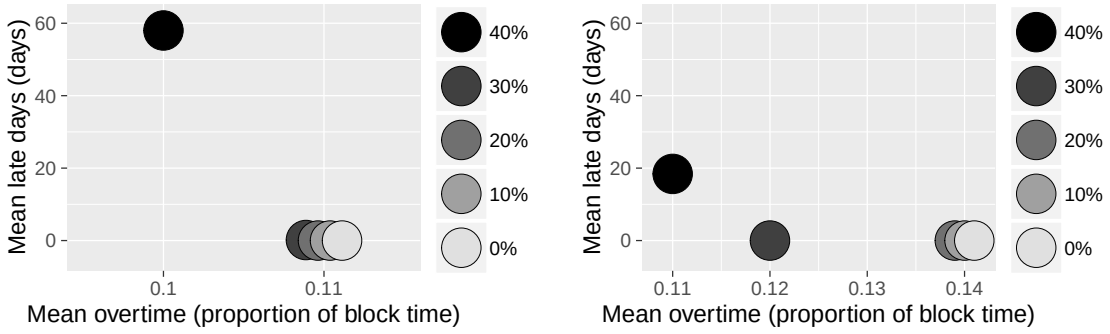
Another input to our model is the arrival distribution of patients, shown in Table 4.3. We assume that 10% of arriving patients are of level 1 urgency, 20% are level 2, and 70% are level 3.

Type	Urgency level	Service 1 Mean	Service 2 Mean
1	1	1	1
	2	2	2
	3	7	7

Table 4.3: Surgery request arrival distribution information in the stylized system.

As an input, we also need to find the appropriate choice of p, the proportion of the block capacity that is carved out for level 1 and 2 patients in the Utilization and Carve-out heuristics. To choose p, we searched over a set of reasonable values of p, to see which one provides the best performance. We considered 0.4, 0.3, 0.2, 0.1, and 0. We compared results using two performance metrics: mean OR overtime as a proportion

of block time, and mean late days, which is the mean number of days patients wait past their SAT over the simulation time horizon. Figure 4.2 shows the result of this study for the two heuristics, using $q = 0.2$ to find the number of available slots in equation (4.9). Both metrics were calculated with a 95% confidence interval, and the half width of the confidence intervals was less then 2% in all instances. This choice of q represents a system that has some additional capacity above the expected mean request arrival rate. Some choices of parameters result in very similar performance in terms of the two performance metrics, and this is represented by the overlap of data points in Figure 4.2.



(a) Utilization heuristic.

(b) Carve-out heuristic.

Figure 4.2: Comparison of Utilization and Carve-out heuristic performance with different choices of p in a stylized system.

Figure 4.2 shows that $p = 0.4$ results in the most late days, while the rest of the choices of p result in little to no average wait days for both heuristics. For the Utilization heuristic, the rest of the choices of p performed very similarly. Therefore to break the tie between the values of p that perform very similarly, we set $p = 0.3$, because 30% of the patient population is of level 1 and 2 urgency in total. For the Carve-out heuristic, $p = 0.3$ achieved the best performance in mean overtime without having late days, thus we set $p = 0.3$ for this heuristic as well.

We tested three cases for the described parameters, with each characterized by the amount of capacity available, where available capacity is calculated according to equation (4.8) and (4.9). Table 4.4 describes the characteristics of the three cases we

considered. The intuition behind having $q \geq 0$ is to have enough capacity to ensure that the system is stable, and practically all patients are served within the simulation time horizon.

Case	q	Slots per service	Block time per service (min)
1	0	10	870
2	0.1	11	957
3	0.2	12	1044

Table 4.4: Cases tested in the stylized system, characterized by the available capacity that is defined by q, the additional proportion of mean surgery request arrival rate considered.

Now that all input parameters have been determined, we can compare the performance of the heuristics: three surgery slot reservation heuristics with different templates, and three FCFS based heuristics: Utilization, Carve-out and FCFS. All heuristics were stable, meaning that all patients received a surgery date within the simulated time horizon, except Template (2,2) when $q = 0$, where almost 20% of level 3 patients were not served. Therefore this heuristic was excluded from consideration when $q = 0$.

Figure 4.3 shows the mean OR utilization results and Figures 4.4 - 4.6 show the patient related metrics for the three cases that are characterized by the choice of q. All performance metrics were calculated with a 95% confidence interval, and the half width of the confidence intervals was less then 2% in all instances. Figure 4.3 shows that as q increases, overtime decreases and undertime increases. Moreover, for the same value of q, the different heuristics tend to perform similarly, differing by at most 8% in mean OR overtime, and by 5% in mean OR undertime. Deviation across heuristics also decreases as the value of q increases.

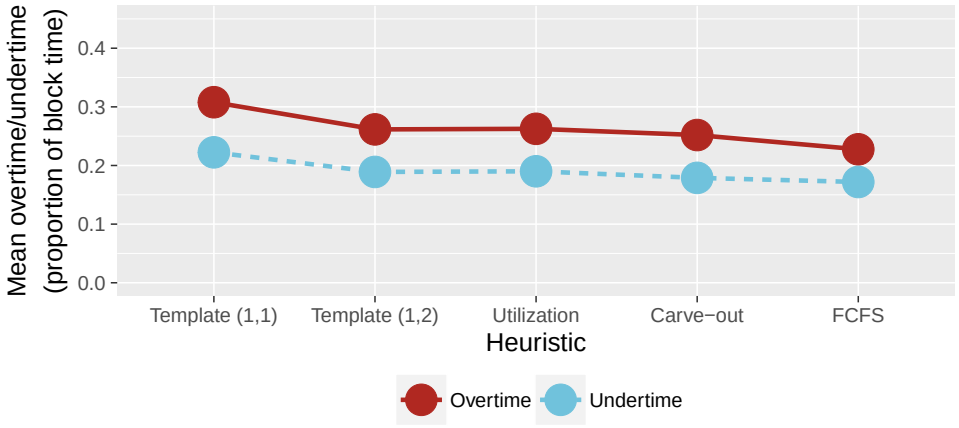
Figure 4.4 shows that the FCFS based heuristics only achieve high performance

in terms of fraction of patients served within their SAT for level 3 patients when $q = 0.2$. Recall, that the Utilization and Carve-out heuristics are set up in a way that level 1 and 2 patients are always served on time. In contrast to all other heuristics, Template (1,1) performs well for all choices of q, while avoiding the violation of SAT for all patients, even level 3. As intuitively expected, the FCFS heuristic results in the worst performance for level 1 and 2 patients for all choices of q considered.

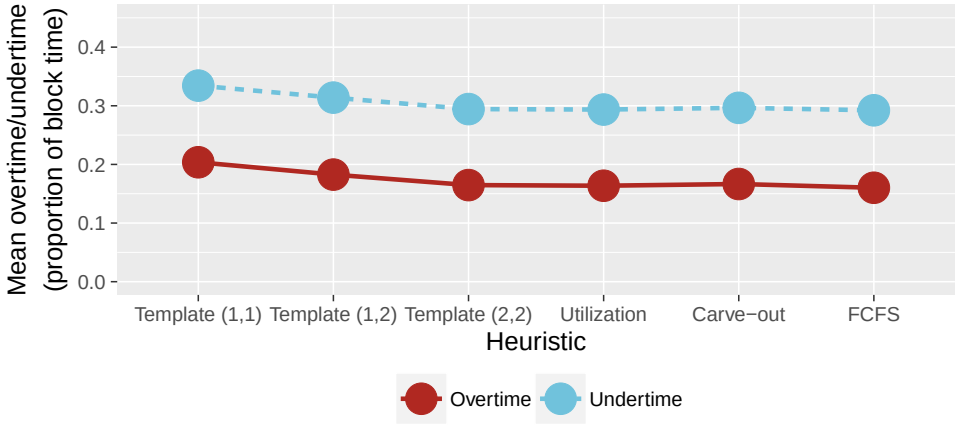
Figure 4.5 shows that even though for lower values of q Template (1,2), Template (2,2), the Utilization heuristic, and the Carve-out heuristic do not perform well in terms of proportion of level 3 patients served within their SAT, the additional time level 3 patients wait for their appointment past their SAT is no more than 8 days, thus they receive an appointment within a month after they request surgery. Recall that the earliest acceptable surgery day for level 3 patients was 4 days. Figure 4.6 shows that for the cases where level 3 patients are served within their SAT, they tend to wait a little over 4 days on average.

We further tested a scenario with $q = -0.1$, i.e., the system is expected to be overutilized. In this setting, all heuristics were unstable, except Template (1,1). The 95% confidence interval half widths were within 1% of all performance metrics. This extreme setting resulted in high overtime (42% of the block time) and low undertime (13% of the block time), which is not unexpected. Moreover 42% of level 3 patients exceeded their SAT, but level 1 and 2 patients were all served on time. The level 3 patients, who exceeded their SAT had to wait 3 days past their SAT for an appointment on average. Thus, even in this extreme setting, where system overutilization is expected, the surgery slot reservation heuristic performs reasonably well.

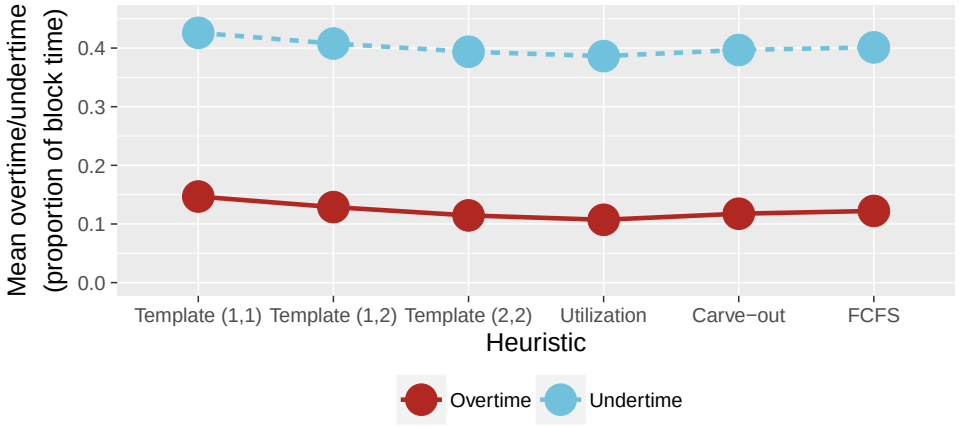
This study shows that the surgery slot reservation heuristic performs well even when there is limited capacity available, as opposed to FCFS based heuristics, whose performance suffers under this condition.



(a) $q = 0$

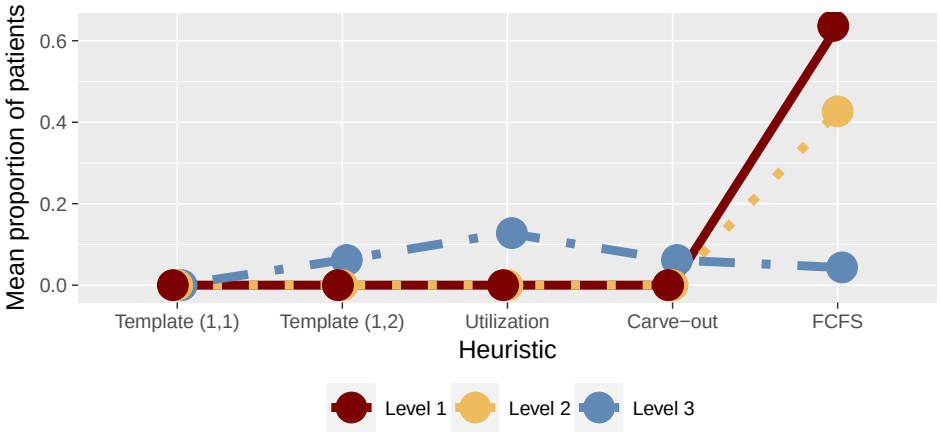


(b) $q = 0.1$

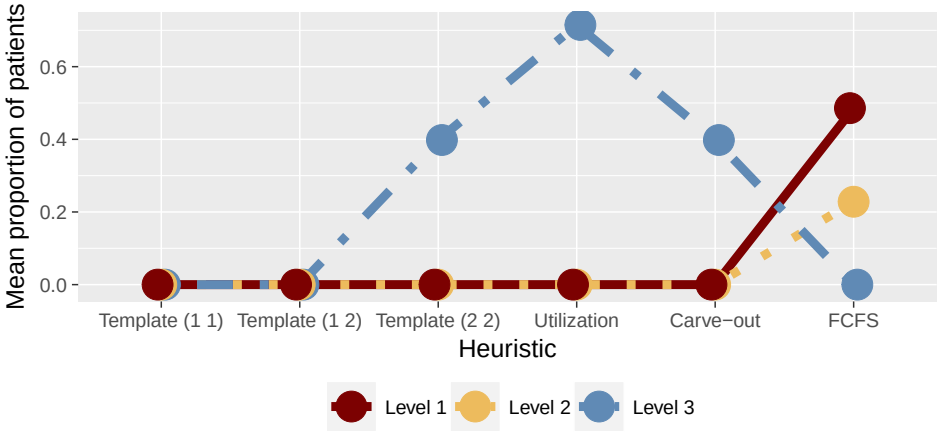


(c) $q = 0.2$

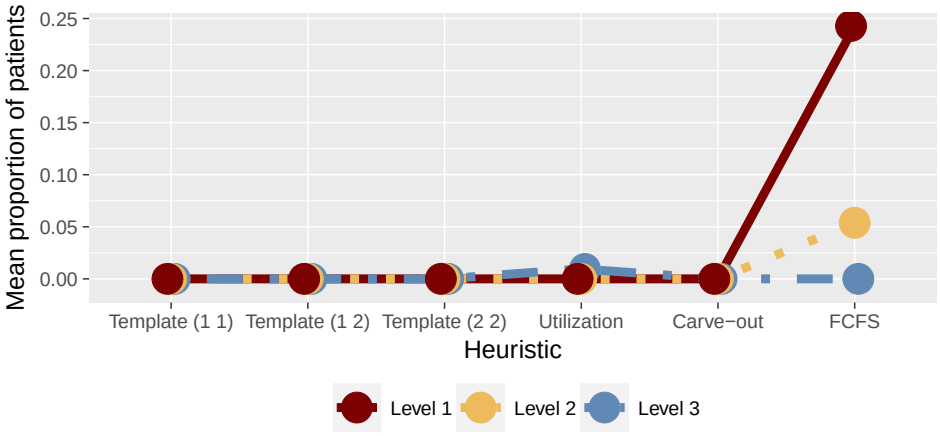
Figure 4.3: Mean OR overtime and undertime as a proportion of block time by heuristic in a stylized system with $q = 0, 0.1, 0.2$. Note that Template (2,2) is unstable when $q = 0$, and as such it is excluded from consideration.



(a) $q = 0$

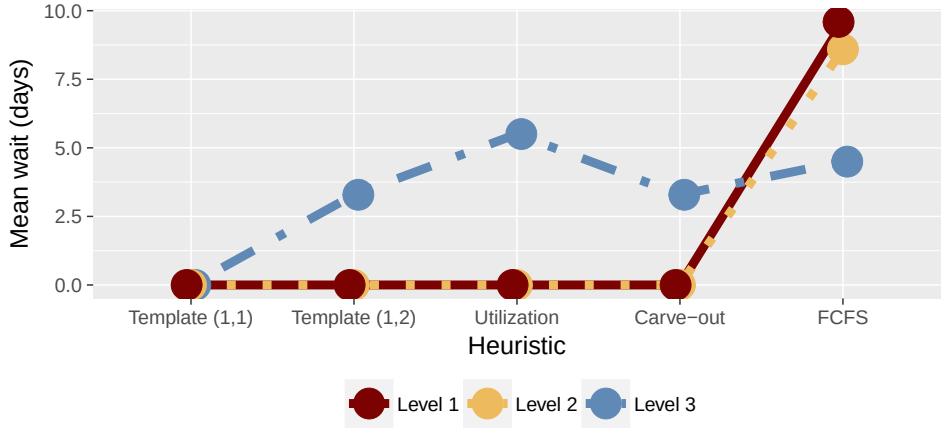


(b) $q = 0.1$

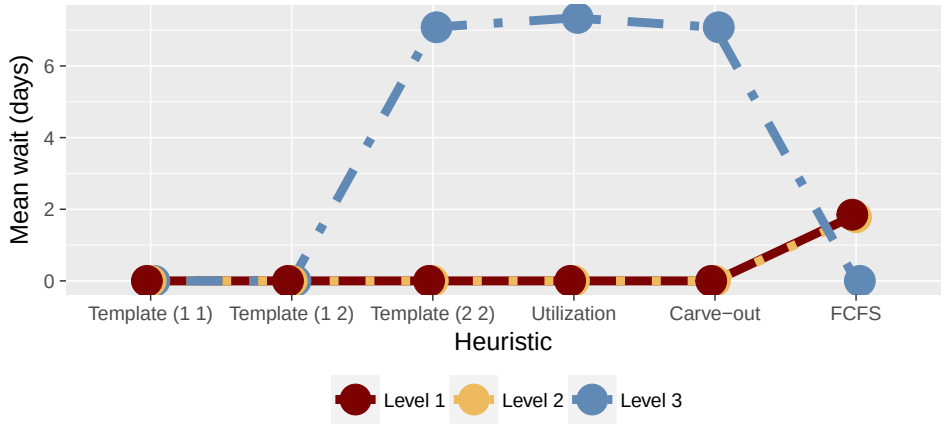


(c) $q = 0.2$

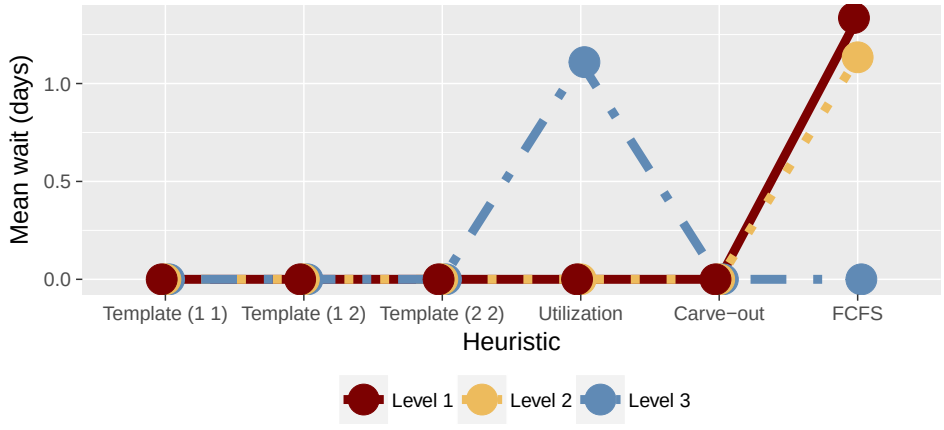
Figure 4.4: Mean proportion of patients that exceed their SAT in a stylized system with $q = 0, 0.1, 0.2$. Note that Template (2,2) is unstable when $q = 0$, and as such it is excluded from consideration.



(a) $q = 0$

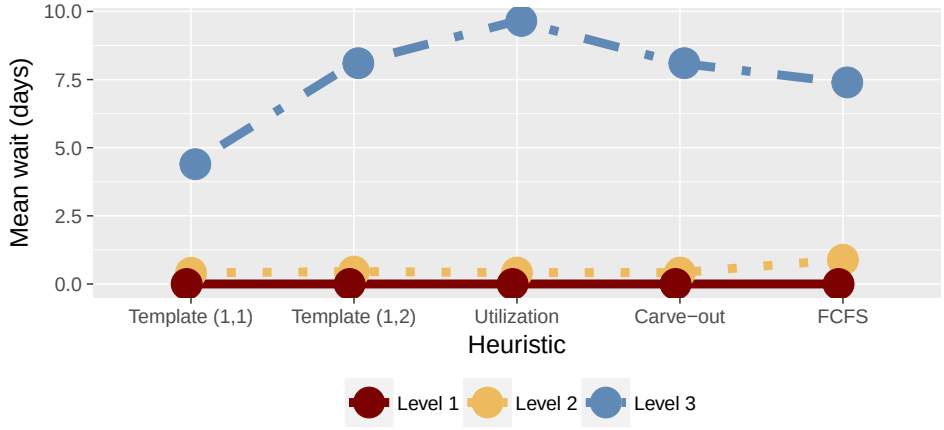


(b) $q = 0.1$

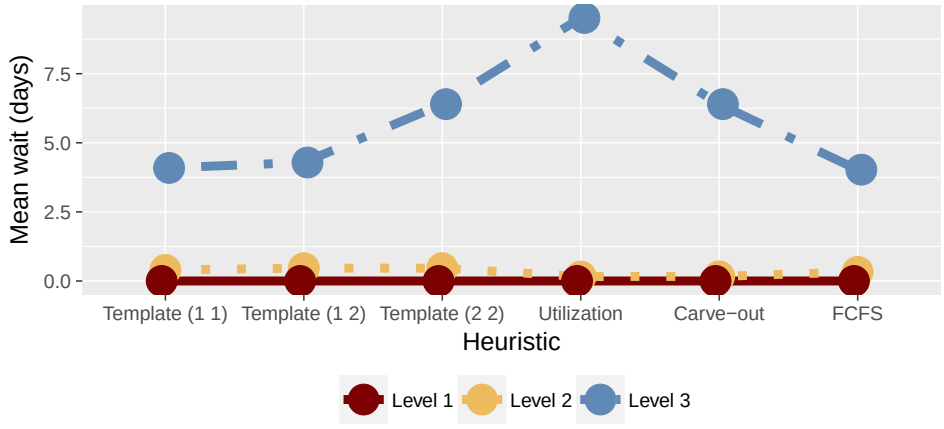


(c) $q = 0.2$

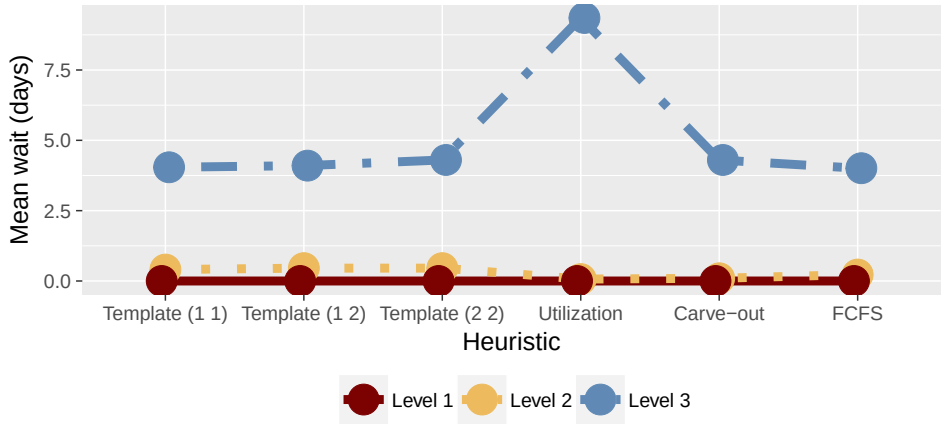
Figure 4.5: Mean number of days patients waited past their SAT, given they exceeded their SAT in a stylized system with $q = 0, 0.1, 0.2$. Note that Template (2,2) is unstable when $q = 0$, and as such it is excluded from consideration.



(a) $q = 0$



(b) $q = 0.1$



(c) $q = 0.2$

Figure 4.6: Mean number of days waited if SAT was not exceeded in a stylized system with $q = 0, 0.1, 0.2$. Note that Template (2,2) is unstable when $q = 0$, and as such it is excluded from consideration.

4.5.2 Hospital Case Study

Similar to the experiments in 4.5.1, we also conducted a case study based on hospital data from one of our partner hospitals, that spanned 2 years. In this study we considered two services: orthopedic and general surgery. For each of the services, two surgery types were considered, characterized by surgery duration: short and long. To determine which surgeries are short and long, mean surgery duration was calculated for each procedure. If the mean surgery duration for a procedure did not exceed the overall median surgery duration, the procedure was categorized as short, otherwise it was categorized as long. A total of 2955 short and 2918 long surgeries were considered in the orthopedic service, and 2300 long and 2294 long surgeries in the general service. Table 4.5 shows the surgery duration distribution information of the surgery types. We again assume that urgency level does not have an affect on surgery duration, and we again use the 60th percentile from the duration distributions for slot length. Similar to our results in Chapter II, we also find here that the 60th percentile of surgery durations is close to the mean surgery duration.

Type	Orthopedic				General			
	Mean	60th	Variance	CV	Mean	60th	Variance	CV
Short	117	121	1622	0.34	120	121	1476	0.32
Long	165	166	2254	0.29	198	200	6623	0.41

Table 4.5: Hospital surgery duration distribution information: mean, 60th percentile, variance and coefficient of variation.

We further analyzed data in our partner hospital that included surgical service, surgery duration, and surgery request arrival day to obtain surgery request arrival distribution information. However, this data did not include urgency class. Therefore we again assume that 10% of the arriving patients are level 1, 20% are level 2, and 70% are level 3. The resulting arrival distribution that is used in the heuristics is shown in Table 4.6. For this study, we used $q = 0.1$, to provide sufficient capacity to ensure the system is stable, i.e., patients are served within the time horizon simulated. Based on the arrival rate and the choice of q, a block time of 3323 minutes

was allocated to the orthopedic service, and 2768 minutes to the general service.

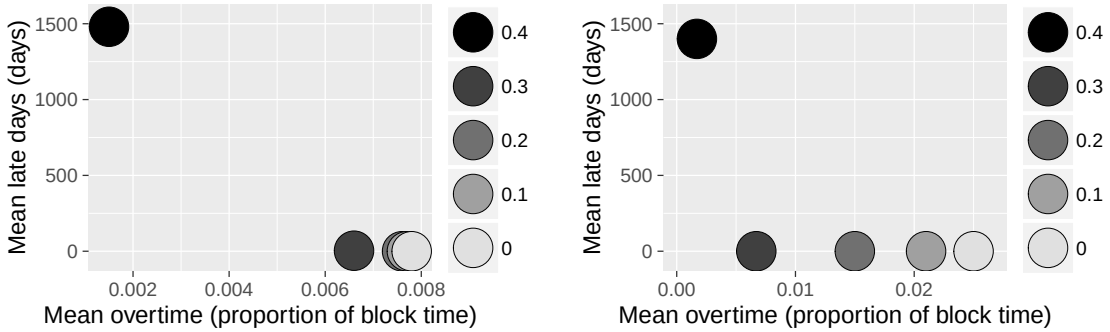
Type	Urgency class	Orthopedic		General	
		Mean	Slots	Mean	Slots
Short	1	1.04	11	0.74	12
	2	2.08		1.49	
	3	7.29		5.21	
Long	1	1.09	8	0.83	9
	2	2.18		1.66	
	3	7.65		5.80	

Table 4.6: Surgery request arrival distribution information in the hospital.

As before, to choose the value of p, the proportion of block time that is carved out for level 1 and 2 patients in the Utilization and Carve-out heuristics, we again considered a set of plausible values: 0.4, 0.3, 0.2, 0.1, and 0. The 95% confidence interval half widths were within 9% of metrics; however, when the half width exceeded 1% of the metric, the absolute value of the half width was less than 10^{-3}. The results in Figure 4.7 are consistent with the stylized system results: $p = 0.4$ results in high values of late days, while the rest of the choices of p result in no late days. Contrary to the stylized case study, where we had to break a tie, here we find that for both heuristics the $p = 0.3$ clearly outperforms the rest of the choices of p in terms of mean overtime, when $p = 0.4$ is excluded from consideration due to the high late day values. Thus we set $p = 0.3$ for both the Utilization and the Carve-out heuristics.

In this study, using Template (2,2), 12% of level 3 patients did not receive an appointment within the three year simulation time horizon, thus we consider this heuristic unstable, and exclude it from consideration.

Figure 4.8 shows mean OR overtime and undertime across heuristics as a proportion of block time. Performance metrics were calculated with a 95% confidence interval, and the half width of the confidence intervals was less then 6% in all instances; however, when the half width exceeded 1% of the metric, the absolute value



(a) Utilization heuristic.

(b) Carve-out heuristic.

Figure 4.7: Comparison of Utilization and Carve-out heuristic performance with different choices of p, based on hospital data.

of the half width was less than 10^{-3}. In this case, the FCFS heuristic achieved the lowest of both of these performance metrics, while all heuristics were all within 4% of each other. Figure 4.9 compares the heuristics in terms of the mean proportion of patients exceeding SAT, and mean patient wait time. For these performance metrics, with a 95% confidence interval, the half width of the confidence intervals was less then 1% in all instances. Template (1,1) and the Carve-out heuristics achieve the best performance in terms of these performance metrics, with Carve-out being slightly better in terms of mean OR utilization. But as before, the added benefit of the ability to better plan for supporting resources gives an advantage to the template based policy, which is beyond our scope to quantify in this chapter.

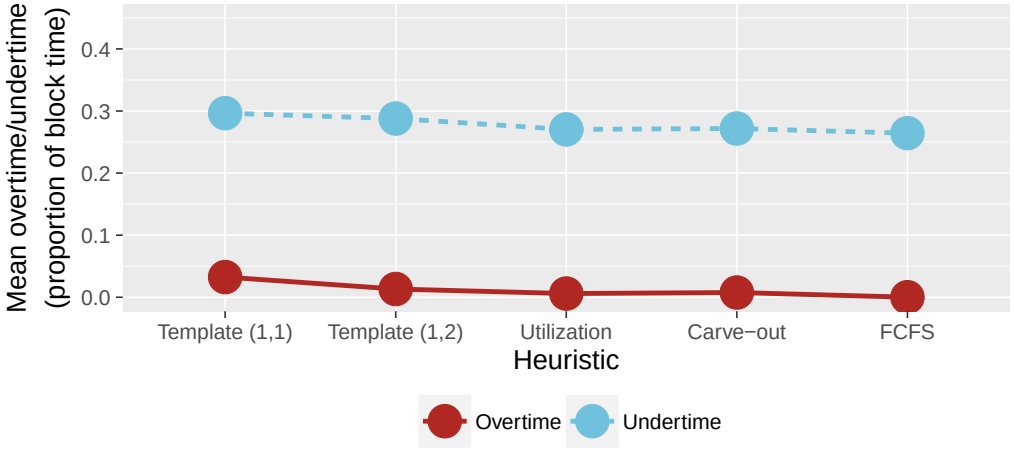
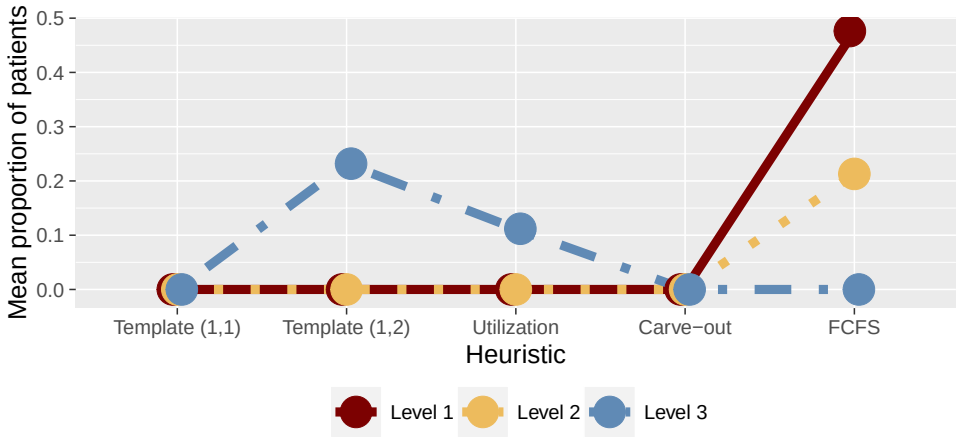
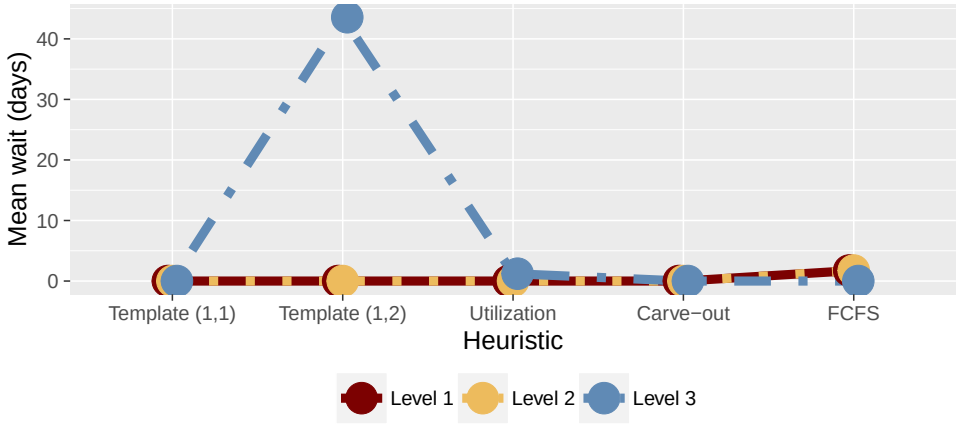


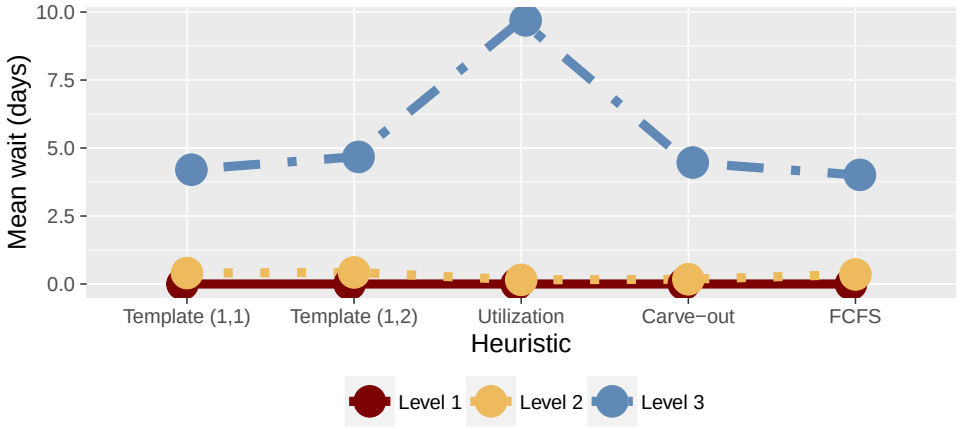
Figure 4.8: Average OR overtime and undertime as a proportion of block time by heuristic based on hospital data.



(a) Mean proportion of patients that exceed their SAT.



(b) Mean number of days patients waited past their SAT, given they exceeded their SAT.



(c) Mean number of days waited if SAT was not exceeded.

Figure 4.9: Comparison of performance metrics by heuristic based on hospital data.

4.6 Conclusions

We conducted a study to analyze the performance of using a surgery slot reservation heuristic with different template settings to assign patients to surgery dates. We used three FCFS based heuristics as benchmarks: classical FCFS; Carve-out, a priority based FCFS policy; and Utilization, an OR utilization based policy. We compared the heuristics based on mean OR overtime and undertime, mean proportion of patients that exceeded their surgery access target, SAT, and mean patient wait time for their appointment.

We found that when limited capacity is available, carefully chosen template settings of the surgery slot reservation heuristic outperformed the benchmarks. These templates tend to be the ones, that reserve limited capacity for level 1 and 2 patients, and allocate most of the capacity to level 3 patients. In our hospital case study, the best performing template reserved 20% of the total capacity for level 1 and 2 patients, even though 30% of the patient population is expected to be of level 1 and 2. This policy performed well due to the rule that releases unused level 3 capacity to more urgent level 1 and 2 patients. This releasing rule makes it hard to develop an intuition on how templates should relate to workload.

In addition to great performance, a carefully selected surgery slot reservation heuristic has the added benefit of providing hospital personnel with additional detail on the quantity and types of patients they can expect, which aids with the planning of supporting resources such as the post-anesthesia care unit (discussed in Chapter II), and surgical nurses (discussed in Chapter III). Thus, using the surgery slot reservation heuristic could help with leveling the workload of downstream resources because of the large level of control over patient type by day, without losing performance in other metrics.

CHAPTER V

Conclusions and Future Research

5.1 Summary and Conclusions

The surgery delivery system is one of the most essential parts of a hospital, as it is a great contributor to both hospital revenue and costs. The many sources of uncertainty contribute to the complexity of the system, as well as the fact that stages of the system are closely coupled, i.e., decisions made in one stage can affect other stages of the system. This thesis focused on developing mathematical models to address three aspects of the surgery delivery system: surgery sequencing (operational level decisions), service group team design and staff shift design and allocation (strategic and tactical level decisions), and operating room (OR) capacity reservation (strategic level decisions).

Due to the complexity of these aspects of surgical care delivery, we focused on developing fast approximation methods that perform well under uncertainty. Such methods have the advantage of lowering the barrier to implementation, as not all hospitals are able to invest in commercial software and expertise necessary to solve complex optimization models. Moreover, algorithms that can be understood without a high level of scientific training help with communicating with healthcare professionals, and convincing them of the effectiveness of the methodology.

The research presented in Chapters II-IV was based on collaboration with experts in healthcare operations management and the resulting methods were tested using data provided to us by our partner hospitals, providing evidence that our results are relevant in a real world setting.

Chapter II focused on sequencing surgeries on a single day, while considering several relevant resources that support surgery: ORs, surgeons, PACU. To address this problem, we developed a deterministic fast 2-phase heuristic that makes decisions on the number of ORs to open, and assigns surgeons to open ORs using the LPT heuristic in the first phase. This phase ignores PACU resources, and only considers the ORs and surgeons with the objective of minimizing the weighted cost of OR overtime and the cost of opening the ORs. In the second phase, considering PACU resources, sequencing decisions of patients and surgeons are made using the difference heuristic. The second phase objective is to minimize surgeon elapsed time, while making sure OR boarding is avoided. We provided tight worst-case performance guarantees for both phases of the heuristic, and showed they have excellent average case performance.

As a benchmark for schedules generated by the 2-phase heuristic, we developed an optimization model that we decomposed into two stages, where each stage corresponds to the phases of the 2-phase heuristic in terms of the decisions made and the objective considered. Moreover, we developed a discrete event simulation to compare the surgery schedules generated by the 2-phase heuristic and the decomposition heuristic in the stochastic setting.

We found that deterministic estimates of surgery and recovery durations used by the heuristics needed to be carefully set to ensure good performance in the stochastic setting. We conducted a study to find what deterministic surgery and recovery durations result in the best schedules when evaluated under uncertainty. We found that the choice of above median percentiles from the surgery and recovery duration distribution led to superior performance: 60th percentile for surgery durations and

70th percentile for recovery durations. We also showed our heuristics can generate surgery schedules that are highly reliable in the stochastic setting.

Based on hospital data, our case study showed that the 2-phase heuristic performs extremely well in the stochastic setting, when compared to the optimization based decomposition heuristic. When evaluated by the discrete event simulation, the cost of 2-phase heuristic schedules were within 10% of the cost of decomposition heuristic schedules in 93% of the test instances, and within 5% in 74% of the instances considered. Moreover, in the 2-phase heuristic schedules, on average 0.05% of the OR time was used for OR boarding, with a maximum of 0.34%. When we compare 2-phase heuristic schedules to hospital schedules, we find that using the heuristic has the potential to lower both OR overtime and surgeon elapsed time, especially for larger services.

Another advantage of the 2-phase heuristic is that decisions that cannot be changed due to hospital policy or personal preference can easily be fixed within the heuristic. For example, in surgical services, where specialized equipment or ORs are needed, surgeon-to-OR assignments can be fixed in phase 1 according to medical needs. Consider another example where a surgeon is only able to operate in the morning due to other obligations in the afternoon, e.g., clinic hours. In this case the second phase decision of sequencing that surgeon can be fixed so that the surgeon operates in the morning. Another example is resequencing of cases. Suppose 2 out of 5 surgeries were performed on a day when an emergent case arose that needed to be operated on. Then in the second phase, the emergent patient can be considered the first patient, and the remaining 3 patients can be resequenced with respect to the emergent patient, or the heuristic can be used to fit all or some of the remaining 3 patients into other ORs. This demonstrates the ability of the 2-phase heuristic to adapt to different hospital settings.

In Chapter III we proposed a two-stage planning level model to address surgical

nurse staffing decisions. The first stage groups surgical services into disjoint teams in a way that balances three key metrics that are important to achieve fairness among nurses across teams: surgical nurse training time, overnight surgical volume of a team, and team size, which is associated with the flexibility of reallocating nurses from their regular service to another service within their team for a short period. We show that the spread, the difference between the maximum and minimum for the performance metrics across teams can be reduced by at least 22% compared to current state.

Once teams are set, the second stage model staffs nurses to services (fixed nurses), and to teams (float nurses). In this stage the model used a weighted objective to balance the minimum number of nurses available per OR during the day, and the number of undesirable shifts. We further proposed a decomposition of the second stage model that first assigned fixed nurses to each service, and then distributed the remaining nurses across teams. We conducted a numerical study of the heuristic, and found that it performs well, while significantly reducing the computational time of instances. Moreover, if parameters change, and the model needs to be resolved, the heuristic has the advantage of causing only moderate disruption in the staffing schedule. For example, if demand for one service changes significantly, the model can be solved for that single service and for the float nurse assignment to teams, while leaving unchanged the shift schedules of the remaining fixed nurses that are assigned to the rest of the services.

Performance of a nurse staffing schedule is measured by how well it covers surgical demand, i.e., are there too few nurses to cover demand (nurse shortfall), or are there too many nurses to cover demand (nurse excess). We conducted a case study based on data from our partner hospital, and we compared coverage of surgical demand of the hospital's nurse staffing schedule to the decomposition heuristic's nurse staffing schedule in two settings. In the first setting parameters were set so that optimized shift mix closely followed the hospital shift mix. This setting is very relevant, as respecting current hospital shift mix can aid implementation of results. In the second

setting, deviations were allowed from the current shift mix, where results can be used in making decisions about what shifts to hire incoming nurses into. We found that significant improvements can be achieved through our methodology, both with the optimized and the current teams. With the current number of FTEs and optimized teams, we showed a minimum of 33% improvement in average nurse shortfall and a 1% improvement in average nurse excess over current state. With the current number of FTEs and current teams, we showed a minimum of 34% improvement in average nurse shortfall and a 1% improvement in average nurse excess over current state.

In Chapter IV we considered the problem of OR capacity reservation within the block schedule. We proposed to use a surgery slot reservation heuristic, where surgery slots were reserved for patients according to patient attributes and patients were assigned to a slot that corresponds to their attribute, unless reservations were released. Patient attributes included surgical service, surgery type, and urgency level. We tested different templates against benchmark heuristics, where a template defines the number of slots allocated to different patient attributes.

Three benchmark heuristics were considered. The first benchmark was the classical first come, first served (FCFS) heuristic, where patients are assigned to the first day with sufficient capacity. The second was the Carve-out heuristic, where the most urgent patients were assigned to the day they arrive, medium urgency patients are assigned to their arrival day or the day following their arrival day, and the least urgent patients were assigned on a FCFS basis, but only a proportion of the block time was made available to them. The third benchmark was the Utilization heuristic, which differed from the carve-out heuristic in that the utilization heuristic tries to assign the least urgent patients to the least utilized day within the patients' surgery access target.

We used a discrete event simulation model to show that the surgery slot reservation heuristic performs very well in terms of all performance metrics we considered:

mean OR overtime, mean OR undertime, mean proportion of patients that exceed their surgery access target, and mean patient wait for appointment. Not only does the heuristic perform well, but it also has the added benefit of aiding the hospital in planning for supporting resources, such as post-anesthesia care unit (PACU) beds, intensive care unit (ICU) beds, and post-surgical ward beds, since the template informs hospitals of the number and types of patients they can expect on any given day.

5.2 Future Research

In summary, we addressed three major areas surrounding the surgery delivery system; however, several more avenues of research can be conducted to build on this thesis. In the context of surgery sequencing, other resources that support and are coupled to surgery could be included, that were not considered in Chapter II, such as the preoperative unit, the ICU, and post-surgical wards. The latter two pose greater challenges, as the length of stay in the preoperative unit is on the same order as the length of stay in surgery and in the PACU (hours); however, length of stay in the ICU and in post-surgical wards is on the order of days. Another way to expand the work of Chapter II is to consider other human resources not mentioned in the chapter, like specialized surgical nurses, PACU nurses, an anesthesiologists, which would lead to more realistic models.

There are several ways the work presented in Chapter III could be augmented. First, we could explore further objective functions and objective function weights in the service group team design problem. In the shift design and allocation problem, we could look into the additional benefit of skill flexibility if float shifts could be applied across teams. *Jordan and Graves* (1995) showed that limited flexibility can yield almost as much benefit as total flexibility. Their *chaining* concept in our context would mean that each nurse would cover one service outside of the team they are assigned to, ideally creating a chain, linking all services. According to *Iravani et al.* (2005),

even a partial chain is sufficient and may be more practical. Another direction is the integration of the service group team design with the shift design and allocation problems into a single model. Due to the computationally challenging nature of this problem, this direction would require further development of heuristic methods to ensure results in a reasonable time. Moreover, the current models are deterministic, and only indirectly consider the variability in surgical case volume and nurse availability due to sickness and other unforeseen events. Our approach is a reasonable one for a planning model, even so, if tractable stochastic models could be developed, they could explicitly take these sources of uncertainty into account. Another future direction could be to further incorporate specific nurse assignments to designed shifts into our model, e.g., nurse A works in service S every weekday from 7AM until 3PM. This would also require the addition of constraints about nurse skills and availability, and possibly union rules or hospital policies about allowable nurse assignments.

Future work related to Chapter IV could explore incorporating additional urgency levels and other metrics into the simulation, such as downstream resources (PACU, ICU, post-surgical ward beds), which would allow for the leveling of resource utilization across the hospital. It may also be beneficial to explore optimization or heuristic based methods for creating templates that would vary by day of week, or week of month, and evaluate additional ways to release reserved capacity. Moreover, comparing our heuristic to additional scheduling schemes, such as priority queues or off-line allocation models could further provide insight into the heuristic performance. This work could also be augmented by considering additional input distributions for surgery request arrivals and surgery durations, which would enable the incorporation of the seasonality of demand into the models. Additionally, we could explore using surgery slot reservation methods to aid hospitals on deciding whether they need to expand their OR capacity, i.e., build new ORs, and if so, how many additional ORs they need. Finally, another avenue of research that could add to this work is patient choice modeling. Currently patient preference has a great impact on what day a patient is assigned to, and this aspect of the problem was not included in our heuristics.

Based on the evidence we have provided, we believe the heuristics we have proposed can be valuable for generating high quality results for hospitals. The methods should also be relatively easy to implement in hospitals from a technical perspective. Finally, the work presented in this thesis provides a foundation for future research on surgery delivery systems.

BIBLIOGRAPHY

Aksin, Z., M. Armony, and V. Mehrotra (2007), The modern call center: A multi-disciplinary perspective on operations management research, *Production and Operations Management*, *16*(6), 665–688.

Anderson, C., and A. Talsma (2011), Characterizing the structure of operating room staffing using social network analysis, *Nursing Research*, *60*(6), 378 – 385.

Astaraky, D., and J. Patrick (2015), A simulation based approximate dynamic programming approach to multi-class, multi-resource surgical scheduling, *European Journal of Operational Research*, *245*(1), 309–319.

Augusto, V., X. Xie, and V. Perdomo (2010), Operating theatre scheduling with patient recovery in both operating rooms and recovery beds, *Computers & Industrial Engineering*, *58*(2), 231 – 238.

Bard, J. F. (2010), Nurse scheduling models, *Wiley Encyclopedia of Operations Research and Management Science*.

Barnhart, C., A. M. Cohn, E. L. Jonson, D. Klabjan, G. L. Nemhauser, and P. H. Vance (2004), Airline crew scheduling, in *Handbook of Transportation Science*, edited by R. W. Hall, chap. 14, pp. 517–560, Kluwer Academic Publishers.

Berg, B. P., and B. T. Denton (2017), Fast approximations for online scheduling of outpatient procedure centers, *In press with INFORMS Journal on Computing*.

Cardoen, B., E. Demeulemeester, and J. Belin (2009a), Optimizing a multiple objective surgical case sequencing problem, *International Journal of Production Economics*, *119*(2), 354–366.

Cardoen, B., E. Demeulemeester, and J. Belin (2009b), Sequencing surgical cases in a day-care environment: An exact branch-and-price approach, *Computers and Operations Research*, *36*(9), 2660–2669.

Cardoen, B., E. Demeulemeester, and J. Belien (2010), Operating room planning and scheduling: A literature review, *European Journal of Operational Research*, *201*(3), 921–932.

Cheang, B., H. Li, A. Lim, and B. Rodrigues (2003), Nurse rostering problemsa bibliographic survey, *European Journal of Operational Research*, *151*(3), 447 – 460.

Creemers, S., J. Beliën, and M. Lambrecht (2012), The optimal allocation of server time slots over different classes of patients, *European Journal of Operational Research*, *219*(3), 508–521.

Dell'Olmo, P., H. Kellerer, M. G. Speranza, and Z. Tuza (1998), A 13/12 approximation algorithm for bin packing with extendable bins, *Information Processing Letters*, *65*(5), 229 – 233.

Denton, B. T., A. J. Miller, H. J. Balasubramanian, and T. R. Huschka (2010), Optimal allocation of surgery blocks to operating rooms under uncertainty, *Operations Research*, *58*(4), 802–816,1028–1031.

Dexter, F., A. Macario, R. D. Traub, M. Hopwood, and D. A. Lubarsky (1999), An operating room scheduling strategy to maximize the use of operating room block time: computer simulation of patient scheduling and survey of patients' preferences for surgical waiting time., *Anesthesia & Analgesia*, *89*(1), 7–20.

Erdogan, S. A., and B. T. Denton (2010), *Surgery Planning and Scheduling*, John Wiley & Sons, Inc.

Ernst, A., H. Jiang, M. Krishnamoorthy, B. Owens, and D. Sier (2004), An annotated bibliography of personnel scheduling and rostering, *Annals of Operations Research*, *127*(1), 21–144.

Etzioni, D., J. Liu, M. Maggard, and C. Ko (2003), The aging population and its impact on the surgery workforce, *Annals of Surgery*, *238*(2), 170–177.

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

Fry, M. J., M. J. Magazine, and U. S. Rao (2006), Firefighter staffing including temporary absences and wastage, *Operations Research*, *54*(2), 353–365.

Garey, M. R., and D. S. Johnson (1979), *Computers and intractability: a guide to the theory of NP-completeness*, W. H. Freeman & Co. New York, NY, San Francisco.

Gerchak, Y., D. Gupta, and M. Henig (1996), Reservation Planning Under for Elective Demand Surgery Surgery for Uncertain Emergency, *Management Science*, *42*(3), 321–334.

Gocgun, Y., and A. Ghate (2012), Lagrangian relaxation and constraint generation for allocation and advanced scheduling, *Computers and Operations Research*, *39*(10), 2323–2336.

Green, L. V., S. Savin, and B. Wang (2006a), Managing Patient Service in a Diagnostic Medical Facility, *Operations Research*, *54*(1), 11–25.

Green, L. V., J. Soares, J. F. Giglio, and R. A. Green (2006b), Using queueing theory to increase the effectiveness of emergency department provider staffing, *Academic Emergency Medicine*, *13*(1), 61–68.

Guerriero, F., and R. Guido (2011), Operational research in the management of the operating theatre: a survey, *Health Care Management Science*, *14*(1), 89–114.

Gul, S., B. T. Denton, J. W. Fowler, and T. Huschka (2011), Bi-criteria scheduling of surgical services for an outpatient procedure center, *Production and Operations Management*, *20*(3), 406–417.

Gupta, D., and B. Denton (2008), Appointment scheduling in health care: Challenges and opportunities, *IIE Transactions*, *40*(9), 800–819.

Heredia-Ureta, H., S. Member, and L. Ortigoza-guerrero (2003), Capacity optimization in multiservice mobile wireless networks with multiple fractional channel reservation, *IEEE Transactions on Vehicular Technology*, *52*(6), 1519–1539.

Hopp, W. J., and M. P. Van Oyen (2004), Agile workforce evaluation: a framework for cross-training and coordination, *IIE Transactions*, *36*(10), 919–940.

Ingolfsson, A., M. A. Haque, and A. Umnikov (2002), Accounting for time-varying queueing effects in workforce scheduling, *European Journal of Operational Research*, *139*(3), 585 – 597.

Inman, R. R., D. E. Blumenfeld, and A. Ko (2005), Cross-training hospital nurses to reduce staffing costs, *Health care management review*, *30*(2), 116–125.

Iravani, S. M., M. P. Van Oyen, and K. T. Sims (2005), Structural flexibility: A new perspective on the design of manufacturing and service operations, *Management Science*, *51*(2), 151–166.

Jacko, P. (2016), Resource capacity allocation to stochastic dynamic competitors: knapsack problem for perishable items and index-knapsack heuristic, *Annals of Operations Research*, *241*(1-2), 83–107.

Jackson, R. L. (2002), The business of surgery. managing the or as a profit center requires more than just it. it requires a profit-making mindset, too., *Health Management Technology*, *23*(7), 20.

Janic, M. (2007), A Heuristic Algorithm for the Allocation of Airport Runway System Capacity, *Transportation Planning and Technology*, *30*(5), 501–520.

Jebali, A., A. B. H. Alouane, and P. Ladet (2006), Operating rooms scheduling, *International Journal of Production Economics*, *99*(1), 52 – 62.

Jordan, W. C., and S. C. Graves (1995), Principles on the benefits of manufacturing process flexibility, *Management Science*, *41*(4), 577.

Kazemian, P., Y. Dong, T. R. Rohleder, J. E. Helm, and M. P. Van Oyen (2013), An ip-based healthcare provider shift design approach to minimize patient handoffs, *Health Care Management Science*, *17*(1), 1–14.

Kazemian, P., M. Y. Sir, M. P. Van Oyen, J. K. Lovely, D. W. Larson, and K. S. Pasupathy (2017), Coordinating clinic and surgery appointments to meet access service levels for elective surgery, *Journal of Biomedical Informatics*, *66*, 105–115.

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

Maass, K. L., B. Liu, M. S. Daskin, M. Duck, Z. Wang, R. Mwenesi, and H. Schapiro (2015), Incorporating nurse absenteeism into staffing with demand uncertainty, *Health Care Management Science*, pp. 1–15.

Maddah, B., L. Moussawi-Haidar, M. El-Taha, and H. Rida (2010), Dynamic cruise ship revenue management, *European Journal of Operational Research*, *207*(1), 445–455.

Marcon, E., and F. Dexter (2006), Impact of surgical sequencing on post anesthesia care unit staffing, *Health Care Management Science*, *9*, 87–98.

May, J. H., D. P. Strum, and L. G. Vargas (2000), Fitting the lognormal distribution to surgical procedure times, *Decision Sciences*, *31*(1), 129–148.

Muñoz, E., W. Muñoz III, and L. Wise (2010), National and surgical health care expenditures, 2005–2025, *Annals of surgery*, *251*(2), 195–200.

Pronovost, P., and J. Freischlag (2010), Improving teamwork to reduce surgical mortality, *JAMA*, *304*(15), 1721–1722.

Saadouli, H., B. Jerbi, A. Dammak, L. Masmoudi, and A. Bouaziz (2015), A stochastic optimization and simulation approach for scheduling operating rooms and recovery beds in an orthopedic surgery department, *Computers & Industrial Engineering*, *80*(0), 72 – 79.

Van Ryzin, G., and J. McGill (2000), Revenue Management Without Forecasting or Optimization: An Adaptive Algorithm for Determining Airline Seat Protection Levels, *Management Science*, *46*(March 2017), 760–775.

Vermeulen, I. B., S. M. Bohte, S. G. Elkhuizen, H. Lameris, P. J. M. Bakker, and H. L. Poutré (2009), Adaptive resource allocation for efficient patient scheduling, *Artificial Intelligence in Medicine*, *46*(1), 67–80.

Villarreal, M. C., and P. Keskinocak (2016), Staff planning for operating rooms with different surgical services lines, *Health Care Management Science*, *19*(2), 144–169.

Wang, Y., J. Tang, Z. Pan, and C. Yan (2014), Particle swarm optimization-based planning and scheduling for a laminar-flow operating room with downstream resources, *Soft Computing*.

Woodall, J. C., T. Gosselin, A. Boswell, M. Murr, and B. T. Denton (2013), Improving patient access to chemotherapy treatment at duke cancer institute, *Interfaces*, *43*(5), 449–461.

You, P. S. (2008), An efficient computational approach for railway booking problems, *European Journal of Operational Research*, *185*(2), 811–824.

Zhang, A. B., P. Murali, M. M. Dessouky, D. Belson, B. Zhang, P. Murali, M. M. Dessouky, and D. Belson (2009), Linked references are available on JSTOR for this article : A mixed integer programming approach for allocating operating room capacity, *Journal of Operational research society*, *60*(5), 663–673.

Zhou, J., and F. Dexter (1998), Method to assist in the scheduling of add-on surgical cases–upper prediction bounds for surgical case durations based on the log-normal distribution.

Zonderland, M. E., R. J. Boucherie, N. Litvak, and C. L. A. M. Vleggeert-Lankamp (2010), Planning and scheduling of semi-urgent surgeries, *Health Care Management Science*, *13*(3), 256–267.