Surgery Scheduling with Recovery Resources

Maya Bam[1]*, Brian T. Denton[1], Mark P. Van Oyen[1],
Mark Cowen, M.D.[2]

[1]Industrial and Operations Engineering,
University of Michigan, Ann Arbor, MI

[2]Quality Institute,
St. Joseph Mercy Health System, Ypsilanti, MI

August 7, 2015

Abstract

Surgical services are both great revenue sources and account for a large portion of hospital expenses. Thus, efficient resource allocation is crucial in this system; however, this is a challenging problem due to the interaction of the different stages of the surgery delivery system, and the uncertainty of surgery and recovery durations. This paper focuses on elective surgery scheduling considering surgeons, operating rooms (ORs), and the post-anesthesia care unit (recovery). We propose a mixed integer programming formulation of this problem and then present a fast 2-phase heuristic: phase 1 is used for determining the number of ORs to open for the day and surgeon-to-OR assignments, and phase 2 is used for surgical case sequencing. Both phases have provable worst-case performance guarantees and excellent average case performance. We evaluate schedules under uncertainty using a discrete event simulation model based on data provided by a mid-sized hospital. We show that the fast and easy-to-implement 2-phase heuristic performs extremely well, both in the deterministic and stochastic settings. The new methods developed reduce the computational barriers to implementation and demonstrate that hospitals can realize substantial benefits without resorting to sophisticated optimization software implementations.

Keywords: Surgery Scheduling, Post-Anesthesia Care Unit, Fast Heuristics, Simulation

*Corresponding author: mbam@umich.edu

1 Introduction

Hospital surgical services are sources of both great revenue and high expenses for human and physical resources. Since most of these resources represent long-term investments, there is a very high fixed cost associated with opening an operating room (OR) unnecessarily. 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]. If these predictions are correct and surgical volume increases in the future, inefficient use of ORs, or nurse and assistant overtime costs caused by poor scheduling will have greater financial impact on the hospital, and therefore increased efficiency will become even more important.

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 a challenging task. 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.

There are several resource assignment challenges as well. In most cases, patient-surgeon assignments have to be respected and each surgeon should perform all their surgeries 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 overtime and surgeon elapsed time (the time between when the surgeon starts their first case and finishes their last case).

This article makes new contributions to surgery scheduling based on collaboration with a mid-sized hospital. First, we present a deterministic mixed integer program (MIP) for cre-

ating elective surgery schedules that consider resources that directly support surgery (e.g., surgeon, OR), and also the limited availability of the PACU, to capture how shortages of one resource can affect the others. Next we present a 2-phase scheduling heuristic (the first phase finds the number of ORs to open and assigns surgeons to ORs, and the second phase sequences cases within a surgeon's block of time while considering the PACU) to provide practical solution methods to the problem, as well as insights. We establish the heuristic's theoretical worst-case performance and average case performance. We further show that the heuristic can be used to provide near optimal solutions to the MIP with much less computational effort. We use numerical experiments based on historical data from our partner hospital to establish the importance of considering the PACU. Finally, using a discrete event simulation model calibrated using hospital data, we evaluate the schedules under uncertainty to demonstrate the heuristic's performance and show that it is scalable to large problems found in practice.

The remainder of the paper is organized as follows. In Section 2 we provide some background on the surgery delivery system, a brief review of the most relevant literature, and describe our approach. In Section 3 we define and formulate the deterministic problem as a MIP. In Section 4 we introduce the 2-phase heuristic and propose a decomposition heuristic for the MIP to be used a benchmark for the 2-phase heuristic. In Section 5 we present a discrete event simulation that we use to evaluate the schedules created by the deterministic models under uncertainty in surgery and recovery durations. In Section 6 we analyze the performance of the proposed 2-phase heuristic to demonstrate that it can generate optimal or near optimal solutions. Finally, in Section 7 we present a case study based on data from our partner hospital to demonstrate how well our schedules perform under uncertainty.

2 Background and Literature Review

The scope of this work is the main ORs of a hospital, and methods to generate elective surgery schedules. 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. 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. If a block time that was assigned to a service does not have a surgery scheduled in it within five business days of the day of surgery, then it becomes open time that can be used by other surgical groups and surgeons within the same service. Block time not used up by 72 hours before the day of surgery is released to all surgical services. The schedule is finalized on the day before surgery. 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.

Similar to other hospitals, Figure 1 shows the stages of the surgery delivery system at our partner hospital. 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 a disadvantage for 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.

There is a substantial literature on surgery planning and scheduling. In our review we focus on the 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, Cardoen et al., 2012]. One approach is to generate schedules considering the

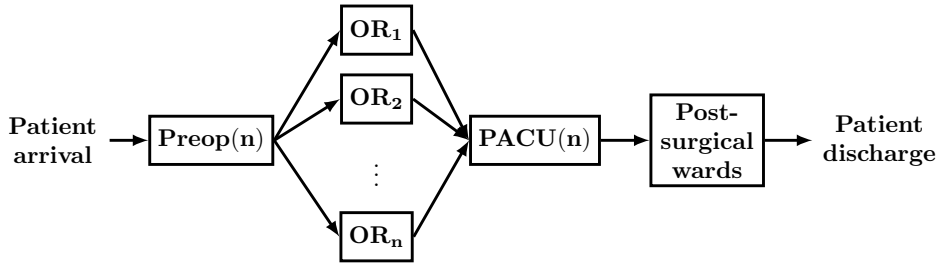

Figure 1: Stages of the surgery delivery system for elective surgeries.

ORs only, and then study the effect of the schedule on the interaction of the ORs and the PACU. In this vein, [Marcon and Dexter, 2006] considered seven sequencing rules trying to find 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] use mathematical programming to decide which cases to perform, and in which ORs to perform the cases without accounting for PACU resources, then they use a discrete event simulation to account for PACU resources.

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 hedging levels with respect to the competing criteria of expected patient wait time and expected OR overtime, where they account for intake, preop, 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 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 phase method for daily OR scheduling. In phase 1 they assign cases 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 phase 2 they sequence

the cases assigned to each OR with the possibility of reconsidering patient-OR assignments and also considering recovery constraints, while minimizing OR overtime. They used two different MIPs in the two phases 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, it is unclear their approach could scale to problems encountered by large hospitals.

[Fei et al., 2010] developed a two-stage heuristic approach, where in the first phase they assign dates to surgeries using a column generation based heuristic to solve their set-partitioning IP model. They model the second phase as a flexible flow shop problem, where they assign surgeries to ORs and sequence them using a hybrid genetic algorithm. Their models respect patient-surgeon assignments, but a surgeon might not perform all their cases consecutively. They also account for recovery time and allow for OR boarding assuming deterministic surgery and recovery durations.

[Wang et al., 2014] considered a particle swarm optimization algorithm for the surgery scheduling problem with post-anesthesia resources. They formulate the problem as a deterministic mixed integer program and propose a discrete particle swarm optimization algorithm combined with heuristic rules, where they find the number of ORs to open and the number of PACU beds needed. They find that their method performs well when compared to optimal solutions. However, their deterministic heuristic is not intuitive and not easy to understand for healthcare professionals. Moreover, they do not consider surgeon blocks or uncertainty. [Cardoen et al., 2009a] use 6 objectives, including minimizing PACU overtime and the peak number of PACU beds used, to optimize case sequencing in an outpatient procedure center. They show that the surgical case sequencing optimization problem is NP-hard and develop exact and heuristic solution approaches for the mixed integer program. [Cardoen et al., 2009b] elaborated on this approach by proposing an exact branch-and-price approach.

[Augusto et al., 2010] investigate the benefit of OR boarding when PACU workload is

greater than OR workload. They consider 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 use 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 show that if the ratio of PACU beds to ORs is lower than 3:2, allowing recovery in the ORs can improve efficiency when PACU workload is greater than OR workload. Their tested instances had 10-30 surgeries, 2-6 ORs, 1-4 PACU beds, and 1-2 transporter teams. Depending on the algorithm they use to build a feasible schedule their worst-case duality gap based on computational experiments is 16.5% or 31.25%. Our paper indicates the significance of PACU congestion when the ratio is 1:1. Even when PACU workload is lower than surgical workload in total for the day, poor sequencing can cause instances where the PACU is full and causes OR boarding.

2.1 Our Contributions to the Literature

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., genetic algorithms, particle swarm algorithms, Lagrangian based methods) which are not accessible to most healthcare professionals at hospitals. In contrast to this prior work, we address the relatively complex problem of scheduling surgeries under limited availability of ORs and PACU beds with a fast, easy to understand, and easy to implement novel 2-phase heuristic, supported by a combination of theoretical analysis of worst-case performance and computational analysis of average case performance.

We first present a new MIP that uses deterministic surgery time and recovery time (both durations are surgeon and case specific) that are carefully selected 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. In our deterministic optimization, we ensure

that there is no OR boarding and patient-surgeon assignments are respected with the objective of minimizing the fixed cost of opening the ORs, the variable cost of OR overtime and the variable cost of surgeon elapsed time. Then we propose a fast 2-phase heuristic that exploits the problem structure, and thus is intuitive for healthcare professionals, and is easy to implement. In addition, we provide worst-case performance guarantees for each of the two phases, and show that on average the heuristic solutions are very close to the optimal solutions. After a schedule is generated, we evaluate it under uncertainty using the discrete event simulation model with the same objective to provide realistic estimates of expected cost. Using the simulation we evaluate the heuristic surgery schedules and optimization based surgery schedules and compare their cost to measure performance of the heuristic in this more realistic setting.

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 [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 reasonable time. Indeed, as we will 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 combine 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 most patients have to go to the PACU after surgery; 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. 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 operations 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 duration of surgeries 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. The length of a time slot is chosen 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.
- 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:

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

$$\text{s.t.} \sum_{i=1}^{P} \alpha_{ijt} \leq x_j \qquad \forall j, t \tag{2}$$

$$\sum_{i=1}^{P} \sum_{j=1}^{R} q_{ijt} \leq \sum_{j=1}^{R} x_j \qquad \forall t \tag{3}$$

$$\sum_{j=1}^{R} x_j \leq R \tag{4}$$

$$\sum_{j=1}^{R} \sum_{t=1}^{T} \alpha_{ijt} = 1 \qquad \forall i \tag{5}$$

$$\sum_{i=1}^{P} q_{ijt} \leq 1 \qquad \forall j, t \tag{6}$$

$$q_{ijt} \geq \alpha_{ijt} \qquad \forall i, j, t \tag{7}$$

$$\sum_{t'=t}^{t+d_i-1} q_{ijt'} \geq d_i \alpha_{ijt} \qquad \forall i, j, t = 1, \ldots, T - d_i + 1 \tag{8}$$

$$\sum_{j=1}^{R} \sum_{t=1}^{T} q_{ijt} = d_i \qquad \forall i \tag{9}$$

$$t q_{ijt} \leq S_j + o_j \qquad \forall i, j, t \tag{10}$$

$$\sum_{i=1}^{P} u_{ikt} \leq 1 \qquad \forall k, t \tag{11}$$

$$\sum_{t=1}^{T} u_{ikt} = d_i s_{ik} \qquad \forall i, k \tag{12}$$

$$\sum_{j=1}^{R} q_{ijt} = \sum_{k=1}^{K} u_{ikt} \qquad \forall i, t \tag{13}$$

$$\sum_{i=1}^{P} t u_{ikt} \leq \Delta_k \qquad \forall k, t \tag{14}$$

$$\sum_{i=1}^{P}(T-t)u_{ikt} \leq \delta_k \qquad \forall k,t \tag{15}$$

$$x_j, \alpha_{ijt}, q_{ijt}, u_{ikt} \in \{0,1\}; o_j, \delta_k, \Delta_k \geq 0 \qquad \forall i,j,k,t. \tag{16}$$

The objective function (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) make sure that ORs are not opened unless they have patients assigned to them. Constraints (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. Constraint (4) makes sure that the number of ORs opened does not exceed the number of ORs available. Constraints (5) make sure that every patient starts surgery, thus no cancellations are allowed. Constraints (6) make sure that at most one patient can occupy an OR in any given time slot. Constraints (7) 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 (8) 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 (9) make sure that the number of time slots allocated to each patient in the OR equals the patient's surgery duration. Constraints (10) make sure that if a patient is in the OR after the allowed session length of the OR, then overtime is used. Constraints (11) make sure that each surgeon can operate on at most one patient at any given time. Constraints (12) 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 (13) make sure that a surgeon operates on the patient when the patient is in the OR. Constraints (14)-(15) are used to calculate the first and last time slots a surgeon is busy.

To speed up solve 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:

$$\sum_{j=1}^{R} \sum_{t=T-d_i+1}^{T} \alpha_{ijt} = 0 \qquad \forall i. \tag{17}$$

We also add additional constraints to eliminate symmetry in the problem [Denton et al., 2010].

Next we 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 (1)-(16) 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

$$\min \sum_{j=1}^{R} \left(c^f x_j + c^v o_j\right) + \sum_{k=1}^{K} c^s(\Delta_k - (T - \delta_k) + 1 - n) \tag{18}$$

s.t. Constraints (2)-(15)

$$\beta_{i,t+d_i-n} \leq \sum_{j=1}^{R} \alpha_{ijt} \qquad \forall i, t = 1, \ldots, T - d_i \tag{19}$$

$$\sum_{t=1}^{T} \beta_{it} = 1 \qquad \forall i \tag{20}$$

$$z_{it} \geq \beta_{it} \qquad \forall i, t \tag{21}$$

$$\sum_{t'=t}^{t+r_i-1} z_{it'} \geq r_i \, \beta_{it} \qquad \forall i, t = 1, \ldots, T - r_i + 1 \tag{22}$$

$$\sum_{t=1}^{T} z_{it} = r_i \qquad \forall i \tag{23}$$

$$\sum_{i=1}^{P} z_{it} \leq B \qquad \forall t \tag{24}$$

$$x_j, \alpha_{ijt}, q_{ijt}, u_{ikt}, \beta_{it}, z_{it} \in \{0, 1\}; o_j, \delta_k, \Delta_k \geq 0 \qquad \forall i, j, k, t. \tag{25}$$

The objective function, equation (18), is 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 (19) 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 (20) make sure that recovery starts exactly once. Constraints (21) make sure that if the patient starts recovery in a time slot, then the patient is in the PACU. Constraints (22) 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 (23) make sure that the number of time slots allocated to each patient in the PACU equals the patient's recovery duration. Constraints (24) 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 true utilization (i.e., overtime and OR boarding is not counted towards utilization). Picking the number of ORs to open, surgeon-to-OR assignments and sequencing patients to avoid OR boarding while minimizing OR idling will result in high utilization.

As before, we can add additional constraints to fix α_{ijt} variables knowing that surgery

had 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 $d_i - n + 1$.

4 Solution Methods

Due to the fact that MIP[OR,PACU] is computationally challenging to solve for realistic problem instances, 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: 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 compare heuristic solutions to the optimal solution due to the computational challenges, to evaluate the performance of the 2-phase heuristic we propose a decomposition heuristic in Section 4.2 that similarly 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 7 we compare the approaches on the basis of computational time and solution quality.

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.

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.

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 least utilized OR, calculated as total surgical time over the planned hours for that room, and breaking ties arbitrarily. (Note that this does not consider the PACU at all; rather, that will be considered in the second phase.) 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. The proofs, which are presented in Appendix A, 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. Let C^H be the cost of the heuristic solution and C^* be the cost of the optimal solution.

Theorem 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, this bound is tight for every even number of ORs.

To complete phase 1, we employ exhaustive search in R, the number of ORs available, to easily find the solution with minimal cost, which will also possess the above shown worst-case performance guarantee.

4.1.2 Phase 2: Sequencing Heuristic

LPT assigns surgeon blocks to ORs, and for this we only need to know 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 naturally defines a sequence of surgeries within a surgeon's block; in fact, 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 (i.e., OR overtime) due to the relatively lower cost of operating the PACU and the objectives of a typical hospital practice. We propose a heuristic for sequencing patients within a single surgeon's block. OR overtime is a nondecreasing function of surgeon elapsed time, thus through minimizing surgeon elapsed time we also minimize OR overtime. 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 $W_{ii} = \infty$. To pick the first patient, i^*, find $W^i = \min_j W_{ij}$ for all i and let $i^* = \text{argmax}_i W^i$. Then the following is the proposed heuristic.

while $\exists i \in \{1, \ldots, P\}$ *that has not been sequenced* **do**

  if $\min_j W_{i^*j} > 0$ **then**

    $i^*_{\text{new}} = \text{argmin}_j W_{i^*j}$

  else

    $i^*_{\text{new}} = \underset{j:W_{i^*j} \leq 0}{\text{argmax}} W_{i^*j}$

  end

  exclude from consideration the row and column corresponding to patient i^*

  $i^* = i^*_{\text{new}}$

end

Once the sequence is set, we assign start times to patients, inserting idle time into the OR schedule to avoid OR boarding. We will refer to this as the *difference heuristic* (DH).

We have the following performance bound when the difference heuristic is used to create a schedule.

Theorem 2. *Letting*

$$D_i = \max_{j:i \neq j}\{(r_i - d_j)^+\} - \min_{j:i \neq j}\{(r_i - d_j)^+\},$$

then for any instance we have

$$C^{DH} - C^* \leq c^s \left(\sum_{i=1}^{P} D_i - \min_i D_i \right),$$

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

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

Theorem 3. *The difference heuristic gives an optimal schedule for any instance where the number of cases assigned to a single surgeon is two.*

For proofs of these theorems, please refer to Appendix B.

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 future, 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.

4.2 MIP Decomposition Heuristic

To evaluate the performance of the 2-phase heuristic, we propose the following decomposition heuristic, which also has two parts, which we will call *steps* to avoid confusion with the phases defined in Section 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 3. That more complex formulation was necessary to lay the foundation for incorporating PACU constraints into the model. The OR scheduling problem, however, 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]:

$$\min \sum_{j=1}^{R} \left(c^f x_j + c^v o_j\right) \tag{26}$$

$$\sum_{k=1}^{K} \left(\theta_{jk} \sum_{i=1}^{P} d_i s_{ik}\right) \leq S_j x_j + o_j \qquad \forall j \tag{27}$$

$$\sum_{j=1}^{R} \theta_{jk} = 1 \qquad \forall k \tag{28}$$

$$\theta_{jk}, x_j \in \{0,1\}; o_j \geq 0 \qquad \forall j,k. \tag{29}$$

The objective function (26) minimizes the fixed cost of opening the ORs and the variable cost of OR overtime. Constraints (27) make sure that if a surgeon is assigned to an OR it will be open and that overtime is used if necessary. Constraints (28) 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]:

$$\sum_{t=1}^{T} q_{ijt} \geq s_{ik}\theta_{jk} \qquad \forall i,j,k. \tag{30}$$

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 (11)-(15) in MIP[OR,PACU] by the following constraints to reduce the number of decision variables:

$$\sum_{i=1}^{P} t q_{ijt} s_{ik} \leq \Delta_k \qquad \forall j,k,t \tag{31}$$

$$\sum_{i=1}^{P} (T-t) q_{ijt} s_{ik} \leq \delta_k \qquad \forall j,k,t. \tag{32}$$

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 represents 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.

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 shows the steps of generating and evaluating a schedule. To generate a schedule using the 2-phase heuristic, we use LPT to get surgeon-to-OR assignments and then the difference heuristic to sequence patients within a surgeon's block, and then surgeons that are assigned to the same OR. In the decomposition heuristic setting we first use the MIP[OR] to get surgeon-to-OR assignments, then use the restricted MIP[OR,PACU] 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.

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 with 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

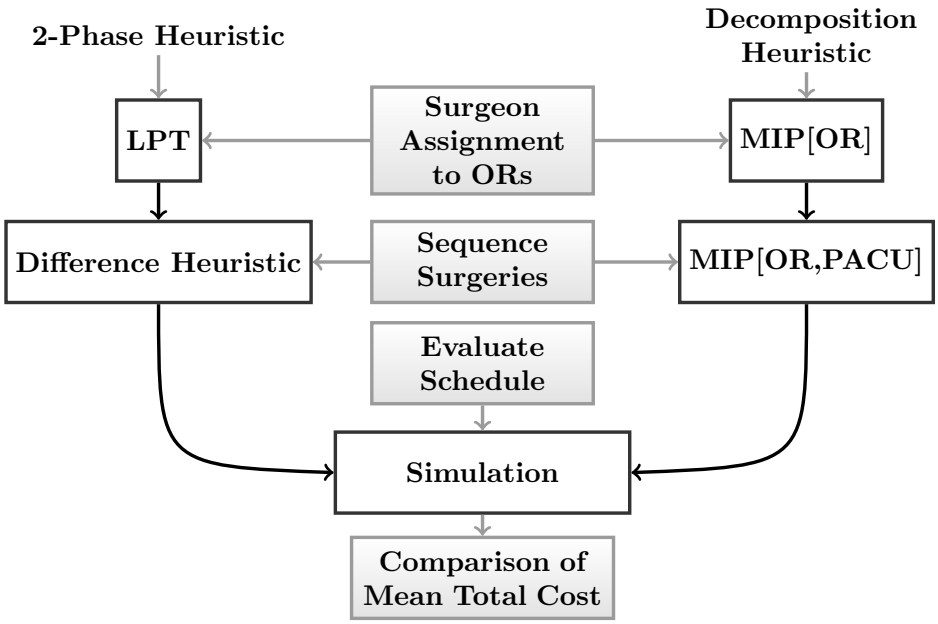

Figure 2: The process of schedule generation and evaluation using two two-stage heuristics: the decomposition heuristic and the 2-phase heuristic.

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. There a random surgery duration is sampled for the patient 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 they board in the OR until a bed becomes available or their recovery duration is up, which similarly to surgery duration is sampled randomly 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.

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. In this section, for a set of randomly generated test cases we compare the numerical performance of the phases of the 2-phase heuristic: LPT and the difference heuristic.

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 sampled from a uniform distribution between 0 and 1, and $S = 1$. Instances were defined in terms of the number of surgeon blocks and the value of c^v; 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:

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

Overall, the average performance was 0.42%, the worst-case performance was 6.99%, and the optimal solution was found 77.41% of the time.

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 heuristic's performance was calculated based on the following formula:

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

Overall, the average performance was 0.70%, the worst-case performance was 30.30%, and the optimal solution was found 95.19% of the time.

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.

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 about arrival and departure times in the ORs and the PACU, and procedure and surgeon information, and spans 14 months.

To test our proposed heuristics, we selected three services (orthopedic, general and urology), that are common to most hospitals. 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.

7.2 Surgery and Recovery Duration Hedging

Since our scheduling models are deterministic, we selected the percentile to be used from the surgery and recovery duration distributions by performing experiments in which schedules based on various percentile combinations were evaluated with the simulation model. 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 percentile, we randomly sampled days for the practices considered (general, orthopedic and urology) to create a set of test instances. We considered 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 limited the size of the test suite due to computational challenges. Figure 3 shows the cost for 12 instances considered, 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 described in Section 7.3.

We establish the importance of considering the PACU 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

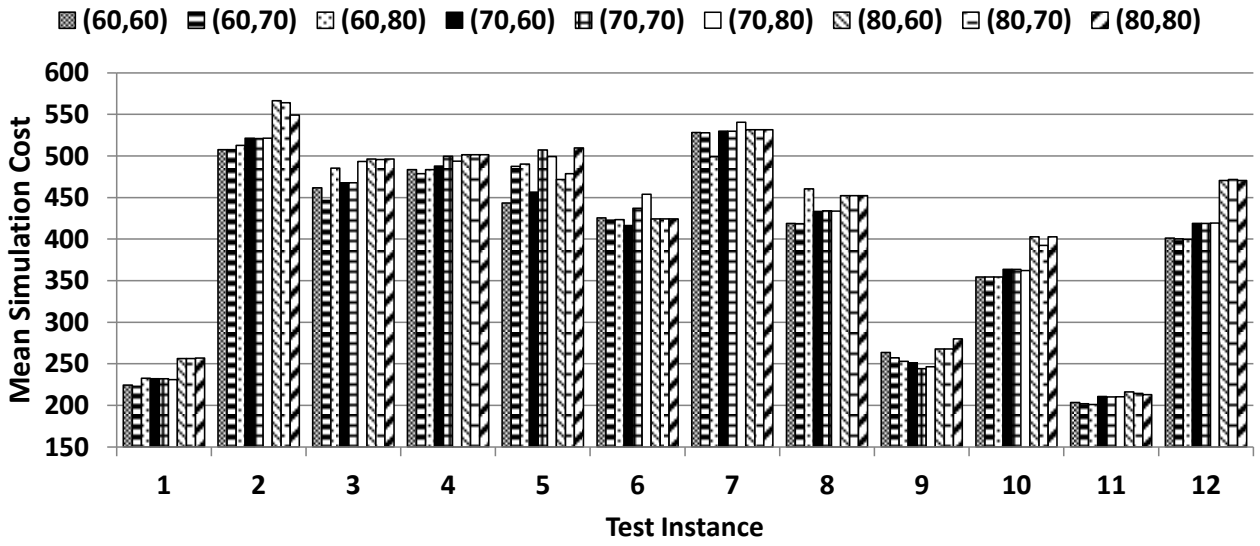

Figure 3: Hedging analysis of randomly sampled days with surgeon and case specific surgery and recovery durations under the decomposition heuristic. Nine pairs of surgery and recovery percentiles are compared for each test instance. The cost scale begins at 150 to accentuate differences.

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 is 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.

7.3 Case Study Results

We considered 43 randomly sampled days. Statistical information about the data considered and computation times is given in Table 1.

Based on the assessment of the importance of criteria for the hospital, the following parameters were used. First, so that about 1.5 hours of overtime would be equivalent to opening a new OR, we set $c^f = 20$ and $c^v = 4$. Second, $c^s = 1$ was selected to ensure surgeon waiting is minimized and to ensure 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 of our partner hospital.

	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
Heuristic CPU time (seconds)	0.000	0.005	0.016
MIP CPU time (seconds)	149	14954	123520

Table 1: Statistics about the data and computational time for the 43 days considered for the case study.

Figure 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 decomposition heuristic in part due to the stochastic performance analysis.

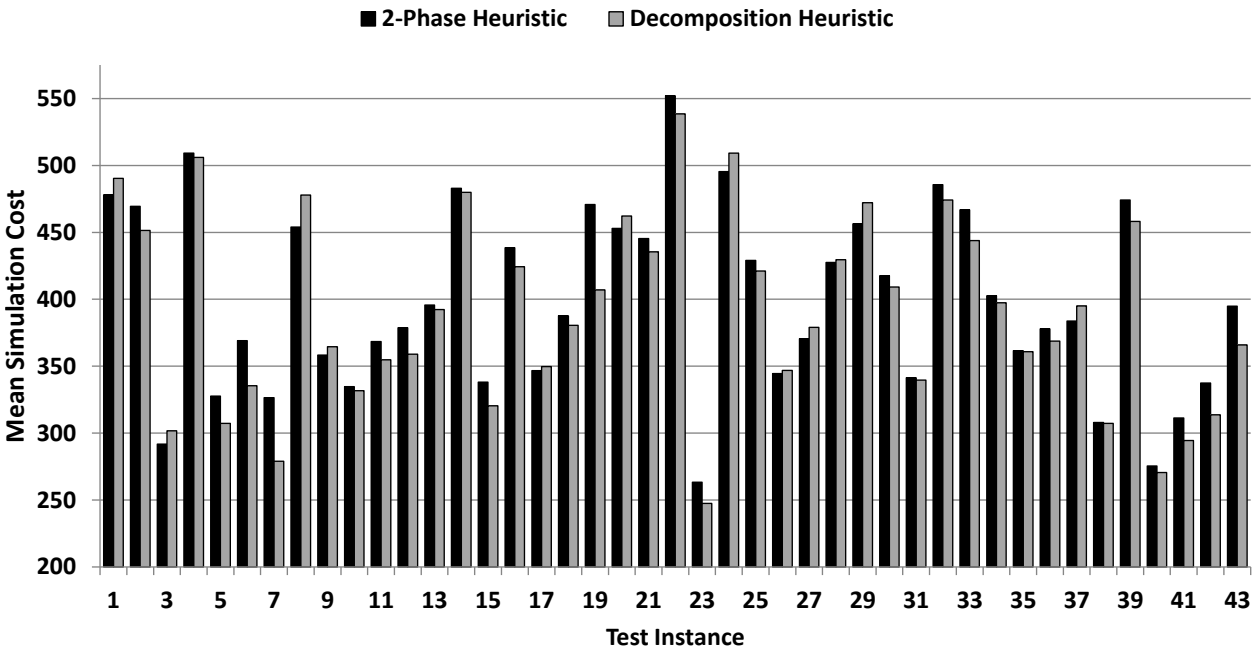

Figure 4: Simulation cost comparison between the decomposition and the 2-phase heuristic. The results are equally good when the cost of OR boarding is considered at the same rate as OR overtime cost. The cost scale begins at 200 to accentuate differences.

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 from the decomposition heuristic in Section 4.2. 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. When evaluated using the simulation model, in the schedules obtained through the 2-phase heuristic 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 possible due to the stochastic performance analysis.

8 Conclusions, Limitations, and Future Work

This paper focused on the problem of creating 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]. Moreover, the 2-phase heuristic is not only fast, and performs well, it is also very intuitive and 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 mixed-integer-programming 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 this bound is tight for LPT for even number of ORs, and that the bound is tight for the difference heuristic.

We recognize the limitation that, although our methodology can contribute to reducing hospital costs, surgeon-to-OR assignments and resequencing cases might have additional complications. Surgeon clinic hours have to be considered, and surgeons may wish to perform the most difficult case first, and some will want to control sequencing. Moreover, unexpected changes in staff availability, or changes in patient condition may require changes to schedules. Nevertheless, we believe the heuristic we have proposed can be valuable for generating a high quality schedule as a starting point which can be adapted to accommodate unexpected needs. We believe that these methods could be implemented in hospitals to achieve great benefits to both the hospital and to the patients.

Future work could include other resources not considered in this paper that support and are coupled to surgery, such as post-surgical wards and the preoperative unit. Consideration of other human resources not mentioned in this paper, like specialized surgical teams, OR and PACU nurses, an anesthesiologists may also lead to more realistic models.

References

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

[Cardoen et al., 2012] Cardoen, B., Demeulemeester, E., and Belien, J. (2012). Operating room planning and scheduling: A literature review. *European Journal of Operational Research*, 201(3):921–932.

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

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

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

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

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

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

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

[Garey and Johnson, 1979] Garey, M. R. and Johnson, D. S. (1979). *Computers and intractability: a guide to the theory of NP-completeness*. W. H. Freeman.

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

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

[Jebali et al., 2006] Jebali, A., Alouane, A. B. H., and Ladet, P. (2006). Operating rooms scheduling. *International Journal of Production Economics*, 99(12):52 – 62.

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

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

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

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

[Zhou and Dexter, 1998] Zhou, J. and Dexter, F. (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. *Anesthesiology*, 89(5):1228–1232.

Appendix A 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 least utilized bin available, 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 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 durations, 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*.

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 semipreemptive 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 semipreemptive solution associated with instance $\mathcal{I}$ (when obvious from the context, we will omit the reference to the instance when talking about costs). We will 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 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 will reintroduce a definition from [Dell'Olmo et al., 1998].

Definition 2. We say that OR R_j^* of 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 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 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 3. □

Lemma 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 semipreemptive 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 3 can be found, similar to the argument in the proof of Lemma 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 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 semipreemptive 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 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 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 5, it is possible to create an optimal semipreemptive 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 will still be 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 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 7 to prove our main result about the worst case performance of LPT for MIP[OR].

Theorem 1. *For any instance $\mathcal{I}$, the following bound holds:*

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

Moreover, for even m the 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 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 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 least utilized OR. 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 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 for even m, consider the following instance. 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.

Appendix B Worst-Case Performance Guarantee of the Difference Heuristic

Theorem 2. *Let*

$$D_i = \max_{j:i\neq j}\{(r_i - d_j)^+\} - \min_{j:i\neq j}\{(r_i - d_j)^+\}.$$

Then for any instance we have

$$C^{DH} - C^* \leq c^s \left(\sum_{i=1}^{P} D_i - \min_i D_i \right),$$

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

Proof. Suppose the optimal schedule is $\mathcal{O}$, and the schedule given by the heuristic is $\mathcal{H}$. Both of these are permutations of the list of surgeries provided. Starting the first two surgeries in the heuristic schedule, $\mathcal{H}(1)$ and $\mathcal{H}(2)$, the maximum benefit of replacing $\mathcal{H}(2)$ by the patient that follows $\mathcal{H}(1)$ in the optimal schedule is

$$c^s \left(\max_{j:\mathcal{H}(1)\neq j}\{(r_{\mathcal{H}(1)} - d_j)^+\} - \min_{j:\mathcal{H}(1)\neq j}\{(r_{\mathcal{H}(1)} - d_j)^+\} \right)$$

By reducing idling, we reduce surgeon elapsed time, thus the multiplication with the appropriate costs.

Note that with P patients there are $P-1$ potential opportunities to reduce idling, however, it is not guaranteed that $\mathcal{H}(1) = \mathcal{O}(1)$. This is the reason we sum over all patients, and then subtract the minimum improvement possible, to get a lower bound.

To see that the bound is tight, consider the following example. Suppose we have three patients with the following surgery and recovery durations: (10,5), (5,17), and (4,12), where the first entry denotes the patient's surgery duration, and the second entry denotes the patient's recovery duration in time slots. With this data the difference heuristic will pick the order 1,2,3 with a surgeon elapsed time of 30, while the optimal solution is 3,1,2 with a

surgeon elapsed time of 19. The difference is exactly as described in the theorem once we scale with the variable cost of surgeon elapsed time. Thus the bound is tight. □

Theorem 3. *The difference heuristic gives an optimal schedule for any instance where the number of cases assigned to a single 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 $\min\{(r_2 - d_1)^+, (r_1 - d_2)^+\}$.

□