IISE Transactions

Operations Engineering & Analytics

Scheduling & Logistics

Taylor & Francis
Taylor & Francis Group

IISE Transactions

ISSN: 2472-5854 (Print) 2472-5862 (Online) Journal homepage: https://www.tandfonline.com/loi/uiie21

Surgery scheduling with recovery resources

Maya Bam, Brian T. Denton, Mark P. Van Oyen & Mark E. Cowen

To cite this article: Maya Bam, Brian T. Denton, Mark P. Van Oyen & Mark E. Cowen (2017) Surgery scheduling with recovery resources, IISE Transactions, 49:10, 942-955, DOI: 10.1080/24725854.2017.1325027

To link to this article: https://doi.org/10.1080/24725854.2017.1325027

IISE TRANSACTIONS
2017, VOL. 49, NO. 10, 942–955
https://doi.org/10.1080/24725854.2017.1325027

Taylor & Francis
Taylor & Francis Group

Surgery scheduling with recovery resources

Maya Bam[a], Brian T. Denton[a], Mark P. Van Oyen[a] and Mark E. Cowen[b]

[a]Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI, USA; [b]Quality Institute, St. Joseph Mercy Health System, Ypsilanti, MI, USA

ABSTRACT

Surgical services are large revenue sources that account for a large portion of hospital expenses. Thus, efficient resource allocation is crucial in this system; however, this is a challenging problem, in part due to the interaction of the different stages of the surgery delivery system and the uncertainty of surgery and recovery durations. This article focuses on single-day in-patient 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 two-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 two-phase heuristic performs extremely well, in both 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.

ARTICLE HISTORY
Received 18 December 2015
Accepted 18 April 2017

KEYWORDS
Surgery scheduling; post-anesthesia care unit; fast heuristics; simulation

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 large and long-term investments, there is a very high fixed cost associated with inefficient scheduling that requires an unnecessarily high number of operating rooms (ORs). 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). Moreover, "aggregate surgical expenditures are expected to grow from $574 billion in 2005 (4.6% of US GDP [gross domestic product]) to $912 billion (2005 dollars) in the year 2025 (7.3% of US GDP)" Muñoz *et al.* (2010, p. 195). If these predictions are correct and surgical volume increases in the future, inefficient use of ORs, supporting resources, and nurse 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 (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.

There are several resource assignment challenges as well. In most cases, patient–surgeon assignments have to be respected and surgeons should perform all of 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. As 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 his or her first case and finishes his or her last case), which is equivalent to minimizing surgeon idle time.

This article 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 that we propose using a discrete-event simulation model based on data from a partner hospital.

2. Background and literature review

The scope of this article 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. In most hospitals, a surgeon can only schedule a surgery if his or her service has block time allocated to him or her or if there is open OR time available. The basic idea of

CONTACT Mark P. Van Oyen vanoyen@umich.edu
Supplemental data for this article can be accessed on the publisher's website.

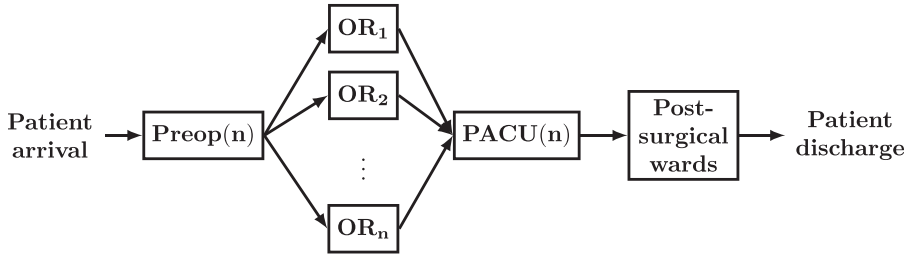

Figure 1. Stages of the surgery delivery system for elective surgeries with *n* preop bays, *n* ORs, and *n* PACU beds.

block scheduling is that either a surgeon or a service is guaranteed the use of a set of rooms for either the entire day or a fraction of a day, and this reservation is known well in advance. The length of the block in block scheduling indirectly places a limit on the number of cases a surgeon can perform and on the choices for assigning physicians to rooms when there is a sufficiently low number of rooms available. However, most hospitals have methods by which unused block time is released and reallocated as the start of the day of surgery approaches, typically 72 hours before the day of surgery. Thus, the actual day-of scheduling may break the constraints of the block schedule. In the problem we solve, the block scheduling rules in place have already informed the list of surgeries to be performed by each surgeon. We only consider elective surgeries, because it is fairly common practice in hospitals to have ORs dedicated to emergent surgeries, and this is also the case at our partner hospital.

Figure 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, he or she is transferred to the preoperative unit. If the patient is just arriving to the hospital, he or she has to go to a check-in area before he or she can go to the preoperative unit. In the preoperative unit the patient is seen by a nurse, an anesthesiologist, and his or her 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, 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 his or her desired ward, an alternate ward if the desired ward is full, or be discharged.

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), Cardoen *et al.* (2010). Unlike the approach of this article, 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 article, 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 of his or her 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 as well as 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 surgeons are not guaranteed, whereas 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 article, they did not demonstrate that 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 article, a surgeon might not perform all of his or her 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.* (2015) 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 with optimal solutions. However, they did not consider surgeon blocks or uncertainty. Cardoen *et al.* (2009a) used six objectives, including minimizing PACU overtime and the peak number of PACU beds used, to optimize case sequencing in an outpatient procedure center but also considered factors such as 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, two to six ORs, one to four PACU beds, and one or two 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 article 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 is 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.

2.1. Our contributions to the literature

This article 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) that 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 with optimization-based solutions for small test instances and when compared with 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 two-phase heuristic, supported by a combination of theoretical analysis of worst-case performance and computational analysis of average case performance.

2.2. Article organization

The remainder of the article is organized as follows. To capture how shortages of one resource can affect the others, Section 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 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 4 we propose a fast two-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 4, we propose a decomposition heuristic for the MIP to be used as a benchmark for the two-phase heuristic, since the overall problem is too computationally challenging to solve to optimality. In Section 5 we describe a discrete-event simulation model that is used to evaluate the generated schedules under uncertainty. In Section 6 we provide worst-case performance guarantees for each of the phases of the two-phase heuristic and show that on average the heuristic solutions are very close to the optimal solutions. Section 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 two-phase 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 (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 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. We assume that 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 of his or her 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 that a bed is available. Moreover, similar to surgery duration, recovery time in the PACU is on the order of hours, whereas 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 an 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 of his or her 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:

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

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

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

$$\sum_{j=1}^{R}\sum_{t=1}^{T}\alpha_{ijt} = 1 \qquad \forall i \tag{4}$$

$$\sum_{i=1}^{P} q_{ijt} \le 1 \qquad \forall j, t \tag{5}$$

$$q_{ijt} \ge \alpha_{ijt} \qquad \forall i, j, t \tag{6}$$

$$\sum_{t'=t}^{t+d_i-1} q_{ijt'} \ge d_i \alpha_{ijt} \qquad \forall i, j, t = 1, \ldots, T - d_i + 1 \tag{7}$$

$$\sum_{j=1}^{R}\sum_{t=1}^{T} q_{ijt} = d_i \qquad \forall i \tag{8}$$

$$t q_{ijt} \le S_j + o_j \qquad \forall i, j, t \tag{9}$$

$$\sum_{i=1}^{P} u_{ikt} \le 1 \qquad \forall k, t \tag{10}$$

$$\sum_{t=1}^{T} u_{ikt} = d_i s_{ik} \qquad \forall i, k \tag{11}$$

$$\sum_{j=1}^{R} q_{ijt} = \sum_{k=1}^{K} u_{ikt} \qquad \forall i, t \tag{12}$$

$$\sum_{i=1}^{P} (T - t) u_{ikt} \le \delta_k \qquad \forall k, t \tag{13}$$

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

$$x_j, \alpha_{ijt}, q_{ijt}, u_{ikt} \in \{0, 1\};$$

$$o_j, \delta_k, \Delta_k \ge 0 \qquad \forall i, j, k, t. \tag{15}$$

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 opened if 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. Constraints (4) make sure that every patient starts surgery; thus, no cancellations are allowed. Constraints (5) make sure that at most one patient can occupy an OR in any given time slot. Constraints (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 (7) make sure that the number of time slots allocated to each patient in the OR after surgery is begun is at least the patient's surgery duration. Constraints (8) make sure that the number of time slots allocated to each patient in the OR equals the patient's surgery duration. Constraints (9) make sure that if a patient is in the OR after the planned session length of the OR, then overtime is used. Constraints (10) make sure that each surgeon can operate on at most one patient at any given time. Constraints (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 (12) make sure that a surgeon operates on the patient when the patient is in the OR. Constraints (13) and (14) are used to calculate the first and last time slots during which a surgeon is busy.

To speed up the 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:

$$\sum_{j=1}^{R}\sum_{t=T-d_i+1}^{T} \alpha_{ijt} = 0 \quad \forall i. \tag{16}$$

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 (1)–(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

$$\min \quad \sum_{j=1}^{R}(c^f x_j + c^v o_j) + \sum_{k=1}^{K} c^s(\Delta_k - (T - \delta_k) + 1 - n) \tag{17}$$

s.t. Constraints (2)–(14)

$$\beta_{i,t+d_i-n} \le \sum_{j=1}^{R} \alpha_{ijt} \qquad \forall i, t = 1, \ldots, T - d_i \tag{18}$$

$$\sum_{t=1}^{T} \beta_{it} = 1 \qquad \forall i \tag{19}$$

$$z_{it} \ge \beta_{it} \qquad \forall i, t \tag{20}$$

$$\sum_{t'=t}^{t+r_i-1} z_{it'} \ge r_i \beta_{it} \qquad \forall i, t = 1, \ldots, T - r_i + 1 \tag{21}$$

$$\sum_{t=1}^{T} z_{it} = r_i \qquad \forall i \tag{22}$$

$$\sum_{i=1}^{P} z_{it} \leq B \qquad \forall t \qquad (23)$$

$$x_j, \alpha_{ijt}, q_{ijt}, u_{ikt}, \beta_{it}, z_{it} \in \{0, 1\};$$
$$o_j, \delta_k, \Delta_k \geq 0 \quad \forall i, j, k, t. \qquad (24)$$

The objective function (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 (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 (19) make sure that recovery starts exactly once. Constraints (20) make sure that if the patient starts recovery in a time slot, then the patient is in the PACU. Constraints (21) make sure that the number of time slots allocated to each patient in the PACU after he or she starts recovery is at least the patient's recovery duration. Constraints (22) make sure that the number of time slots allocated to each patient in the PACU equals the patient's recovery duration. Constraints (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 $d_i - n + 1$.

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 two-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 two-phase heuristic as follows. We propose a decomposition heuristic in Section 4.2 that, similar to the two-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 two-phase heuristic. In Section 7 we compare the approaches on the basis of computational time and solution quality.

4.1. Fast two-phase heuristic

First, we introduce the very intuitive and easy-to-implement two-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 (LPT) first 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 duration 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 of the surgeries they perform for the day) and order the blocks in decreasing order based on 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 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 Online 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.

To complete phase 1, we employ exhaustive search in R; i.e., we perform the heuristic and vary 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 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.

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 the 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**
    $p^*_{\text{new}} = \text{argmax}_j W_{p^* j}$
  else
    $p^*_{\text{new}} = \text{argmax}_{j: W_{p^* j} \leq 0} W_{p^* j}$
  end
  add p^*_{new} to the end of the sequence and exclude this patient from further consideration.
  $p^* = p^*_{\text{new}}$
end

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*.

We have the following performance bound for the difference heuristic.

Theorem 2. *In the difference heuristic setting, where $W_{ij} = r_i - d_j$ for $i \neq j$ and $W_{ii} = \infty$, let*

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

Then for any instance we have

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

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 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 Online Appendix B. Note that the idea behind Theorem 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 his or her first patient and the recovery duration of his or her last patient, the argument proving Theorem 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 following, 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.

4.2. MIP decomposition heuristic

To evaluate the performance of the two-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 4.1. In step 1 we use an MIP to assign surgeons to ORs in the absence of PACU constraints; 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. 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]:

$$\min \quad \sum_{j=1}^{R} (c^f x_j + c^v o_j) \tag{25}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} \left(\theta_{jk} \sum_{i=1}^{P} d_i s_{ik} \right) \le S_j x_j + o_j \quad \forall j \tag{26}$$

$$\sum_{j=1}^{R} \theta_{jk} = 1 \quad \forall k \tag{27}$$

$$\theta_{jk}, x_j \in \{0, 1\}; o_j \ge 0 \quad \forall j, k. \tag{28}$$

The objective function (25) minimizes the fixed cost of opening the ORs and the variable cost of OR overtime. Constraints (26) make sure that if a surgeon is assigned to an OR it will be open and that overtime is used if necessary. Constraints (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]:

$$\sum_{t=1}^{T} q_{ijt} \ge s_{ik} \theta_{jk} \quad \forall i, j, k. \tag{29}$$

Since surgeons are preassigned to ORs, only one patient is allowed to be in an OR at any given time, and because surgeon elapsed time is minimized, there is no need for the variables u_{ikt}, and we can replace Constraints (10)–(14) in MIP[OR, PACU] by the following constraints to reduce the number of decision variables:

$$\sum_{i=1}^{P} t q_{ijt} s_{ik} \le \Delta_k \quad \forall j, k, t \tag{30}$$

$$\sum_{i=1}^{P} (T - t) q_{ijt} s_{ik} \le \delta_k \quad \forall j, k, t. \tag{31}$$

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 7.3.

5. Simulation model

Since the previous models assume deterministic surgery and recovery durations, the question arises of 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 two-phase heuristic. Figure 2 shows the steps of generating and evaluating a schedule. To generate a schedule using the two-phase heuristic, first we use LPT to get surgeon-to-OR assignments and, second, we use 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 4.2 to get surgeon-to-OR assignments and then use the restricted MIP[OR, PACU] from Section 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.

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 (Zhou and Dexter, 1998; May *et al.*, 2000). If enough data were available, we considered surgeon and case-specific surgery and recovery durations. However, some surgeries are performed often by a surgeon, whereas 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 waits in the OR until a bed becomes available or his or her recovery duration is up, which is generated

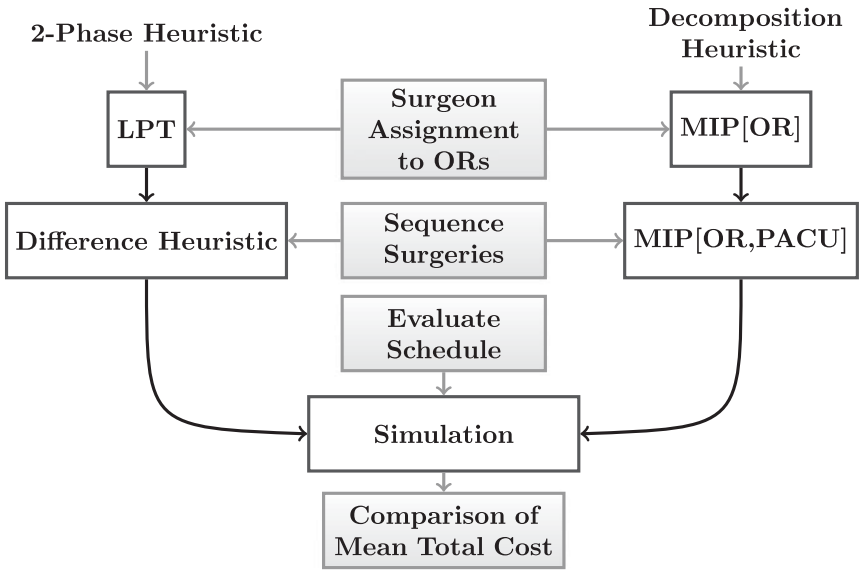

Figure 2. The process of schedule generation and evaluation using two two-stage heuristics: the two-phase heuristic and the decomposition heuristic.

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 two-phase heuristic provides an upper bound on the error across all possible model instances; however, the average performance is also a critical metric, as 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 two-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 two-phase heuristic, we tested LPT on 270 randomly generated instances where surgeon block durations were independent and identically distributed uniform random variables between zero and one 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 one 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 two, four, and eight. The choice of $c^v/c^f = 4$ is intended to be representative of a hospital setting with the additional values of two and eight selected. The performance was calculated using the following formula for the optimality gap:

$$\frac{C^{LPT} - C_N^*}{C_N^*} \times 100\%,$$

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, whereas surgeon block durations close to the OR day duration mean that 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.

6.2. Surgery sequencing: Difference heuristic

In order to estimate the average performance of phase 2 of the two-phase heuristic, we conducted a numerical analysis for the general, orthopedic, and urology surgery services, which 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 of his or her 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:

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

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.

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 were 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 and 20 patients of these types of surgeries. On each day, there were up to 15 ORs available to open. We compared the two heuristics (two-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 that surgeon waiting was minimized and that each surgeon performed all of his or her 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.

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 durations 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 were available, and we assumed a lognormal distribution to find the desired percentile (Zhou and Dexter, 1998; May *et al.*, 2000).

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, the 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, due to 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 nine 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 3 shows the cost for 12 instances considered with 18 patients and eight surgeons on average, as determined

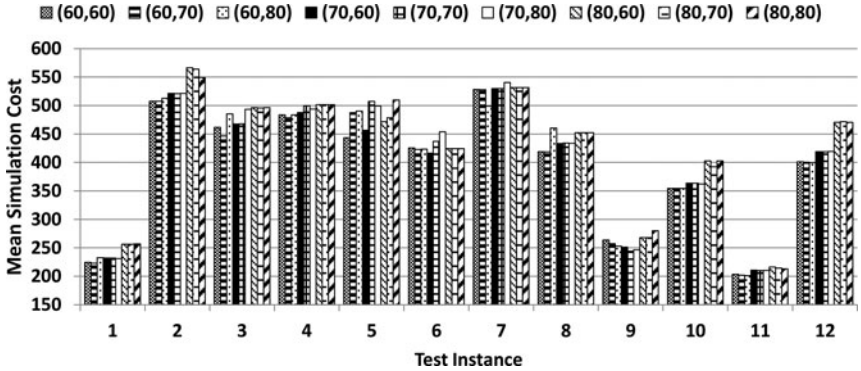

Figure 3. Hedging analysis of randomly sampled days with surgeon and case-specific surgery and recovery durations under the decomposition heuristic. Nine pairs of surgery and recovery percentiles are compared for each test instance.

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 four 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 7.3.

We show that modeling the PACU can significantly reduce overtime costs in the following analysis. As the benchmark for schedules that do not attempt to optimize sequencing, we used phase 1 of the two-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.

7.3. Two-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 1. Observe the dramatic reduction in processing time for the two-phase heuristic. On average, the decomposition heuristic took 3×10^6 times as much CPU time.

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

	Minimum	*Average*	*Maximum*
Surgery duration (minutes)	60	166	375
Recovery duration (minutes)	75	133	210
Number of ORs used	4	6	7
Number of patients	15	18	20
Number of surgeons	6	8	11
Two-phase heuristic CPU time (seconds)	0.000	0.005	0.016
Decomposition heuristic CPU time (seconds)	149	14 954	123 520

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 two-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 two-phase heuristic performed well when compared with the decomposition heuristic, sometimes even beating the decomposition heuristic in part due to the stochastic performance analysis.

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 4.2 to evaluate how often the heuristics found the optimal solution to the overall problem. The two-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 two-phase heuristic is likely to be very good; thus, the additional advantage

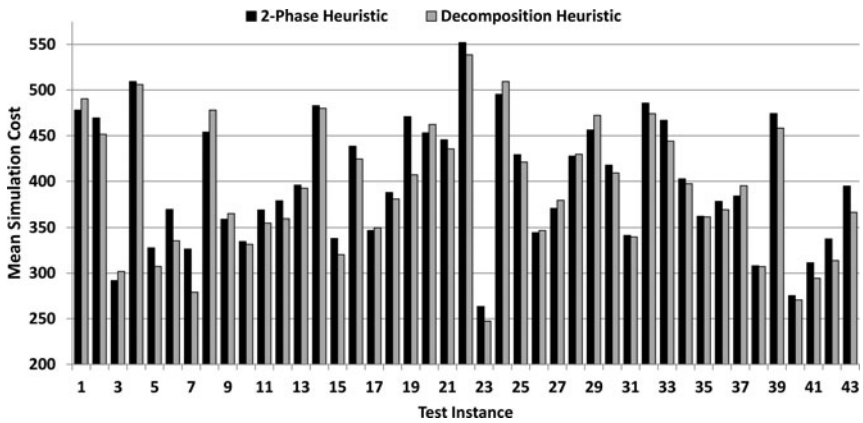

Figure 4. Simulation cost comparison between the decomposition and the two-phase heuristic. The results are equally good when the cost of OR boarding is considered at the same rate as OR overtime cost.

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

		A		*B*		*C*		*D*				
		Heuristic difference: realized – planned (min)		*Hospital difference: realized – planned (min)*		*Planned difference: hospital – heuristic (min)*		*Realized difference: hospital – heuristic (min)*				
Case #	*Surgical service*	*OT*	*SET*	*OT*	*SET*	*OT*	*SET*	*OT*	*SET*	*Avg # of ORs*	*Avg # of patients*	*Avg # of surgeons*
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

of using the computationally challenging optimization models is limited.

Overall, solutions generated by the two-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 two-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 two-phase heuristic, the simulation showed that the average percentage of OR time used for boarding was 0.05% with a maximum of 0.34%. For the decomposition heuristic, the average percentage of OR time used for boarding was 0.27% with a maximum of 3.16%. Moreover, in 33 out of the 43 cases (77% of the instances), the two-phase heuristic achieved less boarding than the decomposition heuristic. This is likely due to the stochastic performance analysis.

7.4. Hospital case study results

We conducted another case study to compare the partner hospital's performance with that of the two-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, which 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 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 two-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 and surgeon elapsed time 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. The results show that both the two-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 cases 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.

8. Conclusions, limitations, and future work

This article 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 two-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 two-phase heuristic, which is deterministic in nature, still performed well under uncertainty when evaluated with a discrete-event simulation model and achieved high resource utilization and improved schedule predictability when compared with a much more computationally intensive heuristic that achieves near-optimal solutions to

MIP[OR, PACU]. It also performed well when compared with hospital schedules. Moreover, the two-phase heuristic is not only fast and performs well but 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 an 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 two-phase heuristic, we proved theoretical worst-case performance guarantees for both phases and showed that the bounds are tight. We also conducted numerical experiments for each heuristic individually and showed that each has excellent average case performance.

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. For example, surgeons may wish to perform the most difficult case first or control sequencing in some other way. Moreover, unexpected changes in staff availability or changes in patient condition may require changes to schedules. Nevertheless, we believe that the heuristic we have proposed can be valuable for generating a high-quality schedule as a starting point, which can be subsequently 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 article 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 article, like specialized surgical teams, OR and PACU nurses, and anesthesiologists, may also lead to more realistic models.

Acknowledgements

The authors thank Jennifer Czerwinski for her significant efforts to create the data set that was used to conduct this research. Furthermore, the authors are grateful to the entire editorial team for their comments that helped improve the article.

Funding

This article is based in part upon work supported by the National Science Foundation under grant numbers CMMI 0844511 (Denton) and CMMI 1233095 and 1548201 (Van Oyen). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Notes on contributors

Maya Bam completed her Ph.D. and M.S. in Industrial and Operations Engineering at the University of Michigan and her B.Sc. in Mathematics at Gordon College. She has recently joined the Operations Research group at General Motors Global Research and Development. She is the recipient of the Rackham Merit Fellowship at the University of Michigan and the winner of the 2016 SAS and INFORMS Analytics Section Student Analytical Scholar Competition, the 2015 INFORMS Interactive Poster Competition, and the 2015 INFORMS Minority Issues Forum Poster Competition. Her research interests include developing well-performing fast approximation methods for computationally challenging problems that arise in scheduling and logistics.

Brian T. Denton is a professor in the Department of Industrial and Operations Engineering at University of Michigan, in Ann Arbor. He is also a faculty member in the School of Medicine and a member of the Cancer Center at University of Michigan. Previously he was an associate professor in the Department of Industrial & Systems Engineering at North Carolina State University, a Senior Associate Consultant at Mayo Clinic, and a Senior Engineer at IBM. He is past president of the INFORMS Health Applications Section and he currently serves as President of INFORMS. His primary research interests are in optimization of decision making under uncertainty with applications to healthcare. He completed his Ph.D. in management science at McMaster University, his M.Sc. in physics at York University, and his B.Sc. in chemistry and physics at McMaster University in Hamilton, Ontario, Canada.

Mark P. Van Oyen is a professor of Industrial and Operations Engineering at the University of Michigan. His interests include the analysis, design, and control of stochastic systems (models and applications). His current research focuses on healthcare operations and medical decision making. He co-authored papers that won the 2016 Manufacturing and Service Operations Management (MSOM) Best Paper award, 2016 MSOM Service Science SIG best paper award, and the 2010 Pierskalla Award. He has served as Associate Editor for *Operations Research*, *Naval Research Logistics*, *IISE Transactions*, and *IIE Transactions on Healthcare Systems and Engineering* and Senior Editor for *Flexible Services & Manufacturing*. He was a faculty member of the Northwestern University School of Engineering (1993–2005) and Loyola University of Chicago's School of Business Administration (1999–2005).

Mark E. Cowen is Chief, Clinical Decision Services for the St. Joseph Mercy Health System, Ann Arbor, Michigan. He also serves as the Director, Data Management Center for the Michigan Arthroplasty Registry Collaborative Quality Initiative. He received his undergraduate and medical degrees from the University of Michigan. He was a clinical instructor for the University of Michigan Medical School for a number of years. After receiving a master's degree in epidemiology from the Harvard School of Public Health, he was Vice-President, Performance Improvement, of Allegiance LLC, a physician–hospital organization. He is currently on the editorial board of the *American Journal of Managed Care*. His research interests include the development, validation, and implementation of prediction rules for patient outcomes.

References

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.

Berg, B.P. and Denton, B.T. (2017) Fast approximations for online scheduling of outpatient procedure centers. *INFORMS Journal on Computing*' (in press).

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, 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.

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

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

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, 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, M.R. and Johnson, D. S. (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., New York, NY.

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

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

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.

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

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, Y., Tang, J., Pan, Z. and Yan, C. (2015) Particle swarm optimization-based planning and scheduling for a laminar-flow operating room with downstream resources. *Soft Computing*, **19**(10), 2913–2926.

Zhou, J. and Dexter, F. (1998) Method to assist in the scheduling of addon surgical cases—Upper prediction bounds for surgical case durations based on the log-normal distribution. *Anesthesiology*, **89**(5), 1228–1232.