INFORMS Journal on Computing
Vol. 23, No. 2, Spring 2011, pp. 220–237
ISSN 1091-9856 | EISSN 1526-5528 | 11 | 2302 | 0220

DOI 10.1287/ijoc.1100.0396

Operating Room Pooling and Parallel Surgery Processing Under Uncertainty

Sakine Batun
Department of Industrial Engineering, University of Pittsburgh, Pittsburgh, Pennsylvania 15261,
sab79@pitt.edu

Brian T. Denton
Edward P. Fitts Department of Industrial and Systems Engineering, North Carolina State University,
Raleigh, North Carolina 27695, bdenton@ncsu.edu

Todd R. Huschka
Department of Health Care Policy and Research, Mayo Clinic, Rochester, Minnesota 55905,
huschka.todd@mayo.edu

Andrew J. Schaefer
Department of Industrial Engineering, University of Pittsburgh, Pittsburgh, Pennsylvania 15261,
schaefer@ie.pitt.edu

Operating room (OR) scheduling is an important operational problem for most hospitals. In this study, we present a novel two-stage stochastic mixed-integer programming model to minimize total expected operating cost given that scheduling decisions are made before the resolution of uncertainty in surgery durations. We use this model to quantify the benefit of pooling ORs as a shared resource and to illustrate the impact of parallel surgery processing on surgery schedules. Decisions in our model include the number of ORs to open each day, the allocation of surgeries to ORs, the sequence of surgeries within each OR, and the start time for each surgeon. Realistic-sized instances of our model are difficult or impossible to solve with standard stochastic programming techniques. Therefore, we exploit several structural properties of the model to achieve computational advantages. Furthermore, we describe a novel set of widely applicable valid inequalities that make it possible to solve practical instances. Based on our results for different resource usage schemes, we conclude that the impact of parallel surgery processing and the benefit of OR pooling are significant. The latter may lead to total cost reductions between 21% and 59% on average.

Key words: operating room scheduling; multiple operating rooms; two-stage stochastic mixed-integer programs; operating room pooling; parallel surgery processing
History: Accepted by Allen Holder, Area Editor for Applications in Biology, Medicine, and Health Care; received December 2009; revised March 2010; accepted April 2010. Published online in *Articles in Advance* July 2, 2010.

1. Introduction

Health-care expenditures in the United States exceeded $2.2 trillion in 2007, accounting for over 16% of the gross domestic product. Hospital expenditures account for approximately a third of this total amount (Centers for Medicare and Medicaid Services 2007), and surgery generates a large portion of a hospital's total expenses and revenues (Healthcare Financial Management Association 2003). Recognizing the significance of efficient allocation of surgery resources, many in the operations research and medical fields have studied operating room (OR) scheduling problems. The majority of articles have focused on either single-OR models or deterministic multi-OR models. In this paper, we consider a model for the optimal design of surgery schedules across multiple ORs under uncertainty. We also show how our model can be used for long-term strategic planning.

Because the OR is typically the bottleneck in the overall process, we make it the central focus of our study. We consider a multi-OR scheduling problem where the surgery durations are uncertain and ORs are identical. The main decisions are the number of ORs to open, the assignment of surgeries to ORs, the sequence of surgeries within each OR, and the times at which surgeons start their first surgery of the day. Our model explicitly considers the fact that some portion of surgeries can be completed in parallel because of the availability of multiple ORs and the assistance of other surgeons in addition to the primary staff surgeon (e.g., surgery fellows).

Beyond operational decisions, our model can also be used to quantify the potential benefit of sharing

ORs among surgeons. In practice, hospitals typically use *block-booking* policies in which surgical groups are given blocks of time in one or more ORs (Dexter et al. 1999a, b). The surgical groups, in turn, allocate these blocks of OR time to individual surgeons, and the further planning is made independently for each surgeon. Splitting resources in this way is motivated by the desire to simplify the planning process; however, it may lead to inefficiencies. Our model can be used to determine the optimal schedule for surgeons assuming that available ORs are pooled together as a common shared resource. Thus, we can use our model to assess the benefits of pooling ORs compared to the commonly used block-booking policy. The parallelizable nature of surgery is an important factor that must be considered to accurately estimate the benefits of OR pooling.

We formulate a two-stage stochastic mixed-integer program (SMIP) for the multi-OR scheduling problem with multiple surgeons. We analyze properties of our model such as the highly symmetric structure of the first-stage problem and the feasibility of the second-stage problem. Standard stochastic programming approaches, such as the L-shaped algorithm, fail for practical instances. Therefore, we exploit a number of structural properties of our model. We present valid inequalities that ensure feasibility of the second-stage subproblems. We show that subproblems can be solved using a fast procedure that exploits their special structure. We also propose a new and widely applicable set of valid inequalities based on Jensen's inequality (Jensen 1906). We perform a series of computational experiments to test our proposed methods.

We calibrate our numerical experiments based on data from the thoracic surgery department at the Mayo Clinic in Rochester, MN. Our results reveal that the benefit of pooling ORs and the impact of parallel surgery processing are substantial, and they become more striking when the cost of surgeon idle time is high. The total operating cost reduction that can be achieved through OR pooling changes between 21% and 59% on average.

The remainder of this paper is organized as follows. In §2, we provide some background on the OR scheduling problem. We review the relevant literature and identify our contribution in §3. In §§4 and 5, we provide the formulation of our model, discuss its structural properties, and present our solution methods. In §6, we present results from our numerical study of our algorithms and managerial insights based on empirical data. Finally, we summarize general insights of our analysis in §7.

2. OR Scheduling Background

The daily fixed cost of opening an OR is significant because of the cost of OR staff and staffing of supporting upstream (intake) and downstream (recovery) areas. Typically, ORs have planned session lengths of eight to nine hours per day. Using an OR beyond this period results in direct overtime costs and indirect costs resulting from staff dissatisfaction. In addition to these costs, there are also less tangible costs such as the costs of surgeon idle time, OR idle time, and patient waiting time.

The *surgery listing* of a surgeon defines the set of surgeries to be performed by him or her on a particular day. Surgeons typically define the order of the surgeries in their listing. The ordering is based on several factors such as the health status of the patients, difficulty and length of the surgery, and other patient- or surgery-related attributes. At many institutions, surgeons are allocated a block of time in an OR during which they may complete their surgeries.

Between two consecutive surgeries in an OR, there are cleaning and setup activities. The time spent on these activities is called *OR turnover time*. In addition to OR turnover time, surgeons also need time between surgeries, which we refer to as *surgeon turnover time*. These two types of turnover times include different resources (ORs and surgeons). Therefore, they may be completed in parallel.

Figure 1 illustrates important aspects of OR scheduling with a simple deterministic example. There are 11 surgeries to be performed by three surgeons. Each block represents a surgery, and the size of the blocks denote the lengths of the surgeries. For example, surgeon 1 has five surgeries, of which the third is the longest. The two white blocks represent the daily session length for two identical ORs (e.g., eight hours). A feasible schedule is illustrated where surgeries of surgeon 2 are scheduled in OR 1, and surgeries of surgeons 1 and 3 are allocated across both ORs. Surgeries in OR 1 are completed after the daily session ends, so there is a certain amount of overtime associated with it. As can be observed, the surgeon idle times between consecutive surgeries in this example are realized mainly because the OR turnover time is significantly greater than the surgeon turnover time (which is true in most surgical environments).

An important consideration in the design of surgery schedules is that surgery durations are highly uncertain (Strum et al. 2000, 2003; Dexter and Ledolter 2005; Dexter et al. 2006). When surgeries are scheduled based on expected values of surgery durations (as is often done in practice), high expected overtime and surgeon idle time may occur (Denton and Gupta 2003), which is illustrated in Figure 2. In this particular example, there are five surgeries to be completed by two surgeons and they are scheduled in two ORs based on their expected durations. In the first scenario, actual durations of the last surgeries in ORs are longer than their expected durations, and this results in unexpected overtime in both ORs. In the second

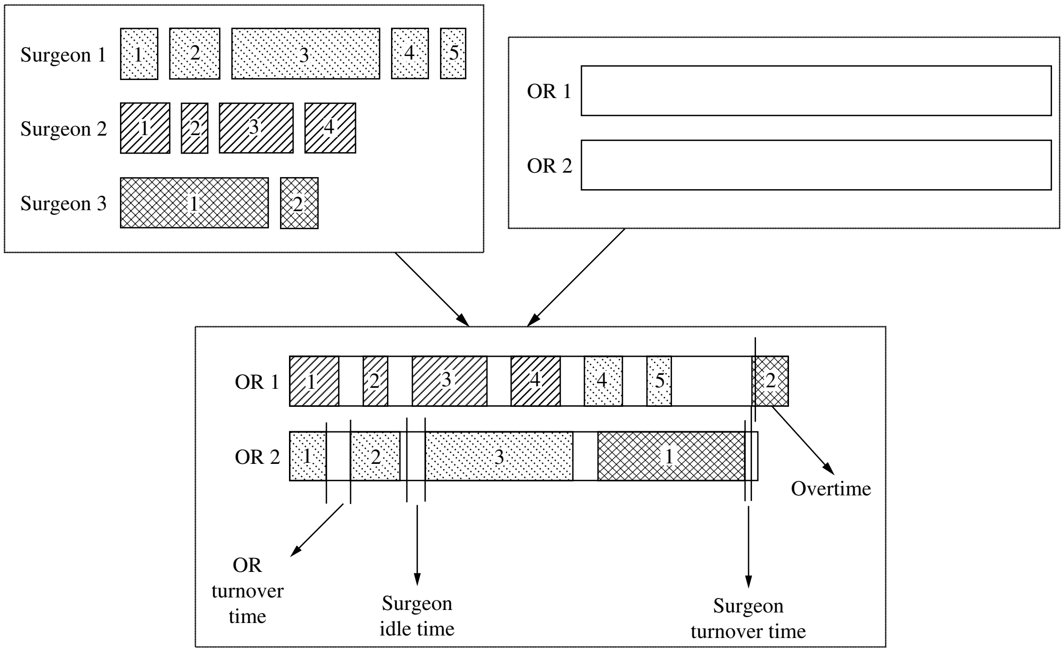

Figure 1 **A Feasible Surgery Schedule Illustrating Three Surgeons Sharing Two ORs**

scenario, the actual durations of the first and second surgeries in OR 2 are shorter than their anticipated durations, and as a result, we observe that the idle time of the corresponding surgeon increases.

A surgery consists of a sequence of several activities, including *preincision*, *incision*, and *postincision*. Although surgeons are key members of the surgical team, they need not be present in the OR for all parts of these activities. For example, the preincision phase includes positioning the patient on the OR bed and initiating anesthetic, and the postincision phase includes closing the incision. Because these activities may also be performed by other members of the team, they do not necessarily require the presence of the surgeon in the OR. This is particularly true for academic medical centers, where surgical fellows may perform these tasks while a staff surgeon operates in another nearby OR. For example, a pulmonary lobectomy consists of an initial incision and separation of the rib cage, followed by the actual lung lobe removal. In an academic medical center, much of this initial work can be done by an experienced surgical fellow,

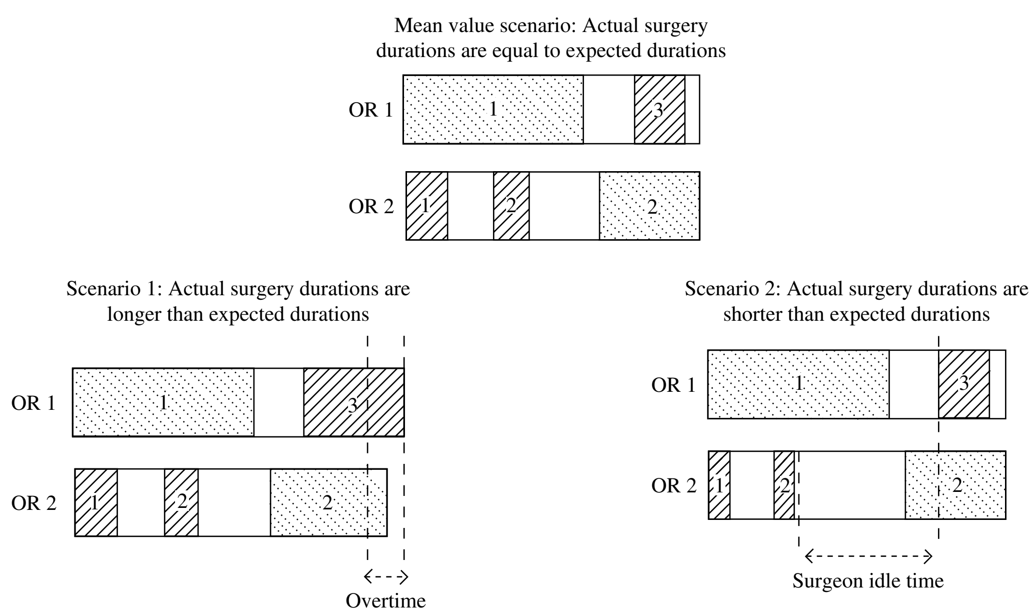

Figure 2 **An Illustration of Overtime and Surgeon Idle Time Under Different Scenarios**

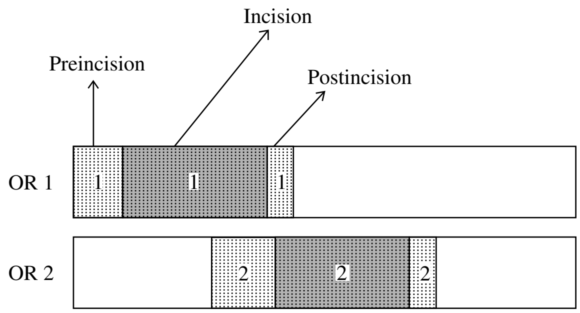

Figure 3 **Parallel Surgery Processing of Two Surgeries Across Two ORs**

with the primary staff surgeon then reviewing the work and performing the critical phase of the surgery. As a result of this flexibility, surgeries can be parallel processed if multiple ORs are available. In such a setting, the surgeon is considered idle if he or she is in the surgical ward but not performing the critical portion of a surgery. Figure 3 illustrates this situation, which we refer to as *parallel surgery processing*. The second surgery starts before the first surgery is completed, and hence the last phases of the first surgery and the first phases of the second surgery are processed in parallel. After performing the incision phase of the first surgery in OR 1, the surgeon uses his or her turnover time and then goes to OR 2 to perform the critical portion of the second surgery. If the surgeon is still occupied with the incision phase of surgery 1 when the preincision phase of surgery 2 is completed, then the incision phase of surgery 2 is delayed until the surgeon becomes available.

3. Literature Review

3.1. Prior Work

Deterministic and stochastic mathematical programming models, queuing models, simulation models, and heuristic approaches have all been widely used to investigate OR scheduling. We focus on those studies that are directly related to multi-OR scheduling or that consider stochastic programming models for OR scheduling. More-extensive reviews are found in Blake and Carter (1997), Gupta (2007), Gupta and Denton (2008), Erdogan and Denton (2011), and Cardoen et al. (2010).

Velásquez and Melo (2005) study the deterministic multi-OR scheduling problem where each surgery has a preferred starting time and the objective is to meet these preferences as much as possible. They formulate the problem as a set-packing problem by discretizing the planning horizon and considering all possible combinations of resources for each discrete unit of time. Exploiting the special structure of this problem, the authors use column generation and constraint branching. Their computational results show that practical instances can be solved within a reasonable amount of time.

Jebali et al. (2006) propose a two-step hierarchical approach to solve a deterministic multi-OR scheduling problem with eligibility constraints related to surgical equipment. In the first step, surgeries are assigned to ORs through the use of an integer programming (IP) model that minimizes the total cost of overtime hours, undertime hours (i.e., the OR idle time), and patient waiting time between initial hospitalization and surgery. In the second step, the surgery sequence within each OR is determined by solving an IP model that further minimizes the total overtime in ORs.

Testi et al. (2007) propose a three-phase method to generate weekly schedules for a multi-OR surgical suite. In the first phase, the available OR time is distributed among surgical wards based on their demands by solving an IP model that is similar to a bin-packing problem. In the next phase, a weekly cyclic timetable is determined by using an IP model that maximizes the surgeon preferences subject to several constraints. In the final phase, patients for the next available day are selected based on a priority score, and surgeries within each OR are sequenced using simple sequencing rules such as longest waiting time, longest processing time, and shortest processing time. The performance of these rules is analyzed by using a discrete event simulation model. Constructing a cyclic schedule of surgeries is also studied by several other researchers, including Beliën and Demeulemeester (2007), van Oostrum et al. (2008), and Adan et al. (2009).

Weiss (1990) considers the problem of minimizing resource idle time and procedure waiting time in a single-OR environment where the surgery durations are uncertain and the decisions are the sequence of surgeries and their start times. He solves small problem instances that include two or three surgeries, and his numerical results reveal that the solution highly depends on the cost coefficients. Wang (1993) also considers a single-server (equivalently OR) appointment system where the processing times are assumed to have phase-type distribution. Exploiting the special structure of the problem, he is able to solve larger instances than Weiss and shows that constant interarrival times cannot guarantee optimality.

Denton and Gupta (2003) study the single-server appointment-scheduling problem where the service durations are stochastic and the sequence of customers is fixed. The objective is to determine appointment times for the customers to minimize the total expected cost of customer waiting time, server idle time, and tardiness with respect to the session length. They formulate this problem as a two-stage stochastic linear program, derive upper bounds that are independent of job duration distribution type, and solve the problem by using these bounds in a modified L-shaped algorithm that is based on successively partitioning the space of the random job durations. Denton et al. (2007) extend this study to investigate surgery sequencing and start-time decisions. They consider several different surgery sequences obtained with simple heuristic rules. Their computational results show that the performance of OR schedules are affected both by scheduled start times and sequencing decisions.

Denton et al. (2010) study the deterministic and stochastic versions of the surgery allocation problem in a multi-OR environment where the main aim is to minimize the total fixed cost of opening ORs and expected overtime cost. They focus only on the allocation decisions and do not consider the sequencing decisions within the ORs. For the stochastic version of the problem, they present both a two-stage SMIP with binary variables in the first stage and a robust formulation. To solve the problem, they develop valid inequalities that reduce symmetry and use lower and upper bounds on the optimal number of ORs to open each day. Moreover, they propose a simple and fast heuristic that performs reasonably well across many instances.

Only a few studies consider stochastic programming-based approaches to capture the stochastic behavior of surgery durations (Denton and Gupta 2003; Denton et al. 2007, 2010). These studies view surgery as a single activity; however, in practice a surgery is comprised of several activities. Depending on surgery-to-OR assignment decisions, some of these activities can be carried out simultaneously (in parallel), and parallelizing surgeries may improve the efficiency of resource usage significantly (Sandberg et al. 2005; Sokal et al. 2006, 2007; Marjamaa et al. 2009). To our knowledge, parallel surgery processing has not yet been incorporated into any optimization models. However, considering parallel surgery processing is essential to estimating the benefits of pooling OR capacity.

3.2. Our Contributions

This paper differs from the existing literature in a number of ways. The contributions made in this study are as follows.

- We model the stochastic multi-OR scheduling problem, integrating allocation, and sequencing decisions.
- We consider surgeons, as well as ORs, as resources.
- We provide a more realistic model of the surgery process by explicitly considering the preincision, incision, and postincision phases.
- Because our problem is unsolvable with standard techniques, we exploit several structural properties of our model. We also present a novel and widely applicable set of valid inequalities that are essential to solving large instances.
- We quantify the benefit of OR pooling and illustrate the impact of parallel surgery processing on the performance of surgery schedules.

4. Problem Definition and Mathematical Formulation

Our model considers daily decisions that include the number of ORs to open, surgery-to-OR assignment decisions, the sequence of surgeries within each OR, and the start time for each surgeon on the day of surgery. We formulate our model as a two-stage stochastic program with recourse (Dantzig 1955). In the first stage, the model determines the number of ORs to be opened, the assignment of surgeries to ORs, the sequence of surgeries within each OR, and start time for each surgeon. These decisions are made prior to the day of surgery (e.g., usually 24–48 hours in advance). Next, on the day of surgery, the actual surgery durations become known. Uncertainty in surgery durations is represented by a finite set of *scenarios* in the second stage. Each scenario is composed of collective random outcomes for the preincision, incision, and postincision durations of surgeries. Second-stage decisions include actual surgery completion times, surgeon idle times, and overtime in each OR. The objective of our model is to minimize total costs, including first-stage costs of opening ORs and expected second-stage costs of overtime and surgeon idle time. We use the following notation in our formulation.

Indices

i, j: surgery indices.
k: surgeon index.
q, r: OR indices.
ω: scenario index.
i_k: index of the first surgery of surgeon k.

Problem Instance-Related Parameters

n: total number of surgeries to be scheduled.
n_R: total number of available ORs.
n_S: total number of surgeons.
b_{ijk}: binary parameter denoting whether surgery i immediately precedes surgery j in surgeon k's surgery listing.
$\text{pre}_i(\omega)$: preincision duration of surgery i under scenario ω.
$p_i(\omega)$: incision duration of surgery i under scenario ω.
$\text{post}_i(\omega)$: postincision duration of surgery i under scenario ω.

Configuration or Environment-Related Parameters

L: session length for each OR.
c^f: daily fixed cost of opening an OR.

c^o: per-minute overtime cost of an OR.
c^S: per-minute idle time (waiting time) cost of a surgeon.
s^S: surgeon turnover time between two consecutive surgeries.
s^R: OR turnover time between two consecutive surgeries.

In our notation, $\omega \in \Omega$ represents the random outcome of the realized scenario. Given n surgeries, we obtain a random vector $\xi(\omega) = \{\text{pre}_1(\omega), \ldots, \text{pre}_n(\omega), p_1(\omega), \ldots, p_n(\omega), \text{post}_1(\omega), \ldots, \text{post}_n(\omega)\}$. We denote the finite support of $\xi(\omega)$ by Ξ, where $\Xi \in \mathbb{R}_+^{3n}$.

Decision Variables

x_r: binary decision variable denoting whether OR r is opened or not.
y_{ir}: binary decision variable denoting whether surgery i is allocated to OR r or not.
z_{ijr}: binary decision variable denoting whether surgery i precedes surgery j in OR r or not (defined for (i, j, r): $i \neq j$). Note that z_{ijr} does not denote immediate precedence but denotes general precedence relation between i and j. z_{ijr} is fixed to 0 if j precedes i in one of the surgeons' surgery listing.
t_k: start time for surgeon k.
$C_{ir}(\omega)$: completion time for surgery i in OR r under scenario ω.
$I_{ij}(\omega)$: surgeon idle (waiting) time between surgeries i and j under scenario ω (defined for (i, j): $\sum_{k=1}^{n_S} b_{ijk} = 1$; i.e., i immediately precedes j in one of the surgeons' surgery listing).
$I_k(\omega)$: idle time of surgeon k before his or her first surgery under scenario ω.
$O_r(\omega)$: overtime in OR r, with respect to session length L under scenario ω.

Note that while defining the parameters and decision variables, we use only one (or two, depending on the number of subscripts) of the indices to denote the sets. However, our definitions apply to other indices denoting the same set. For example, C_{*r} applies to subscripts i, j, and i_k.

Using the above notation, we formulate the model as follows:

$$\min \sum_{r=1}^{n_R} c^f x_r + \mathcal{Q}(x, y, z, t) \tag{1a}$$

$$\text{s.t.} \quad y_{ir} \le x_r \quad \forall i, r, \tag{1b}$$

$$\sum_{r=1}^{n_R} y_{ir} = 1 \quad \forall i, \tag{1c}$$

$$z_{ijr} + z_{jir} \le y_{ir} \quad \forall i, j > i, r, \tag{1d}$$

$$z_{ijr} + z_{jir} \le y_{jr} \quad \forall i, j > i, r, \tag{1e}$$

$$z_{ijr} + z_{jir} \ge y_{ir} + y_{jr} - 1 \quad \forall i, j > i, r, \tag{1f}$$

$$t_k \le L \quad \forall k, \tag{1g}$$

$$x_r, y_{ir}, z_{ijr} \in \{0, 1\} \quad \forall i, j \neq i, r, \tag{1h}$$

$$t_k \ge 0 \quad \forall k, \tag{1i}$$

where

$$\mathcal{Q}(x, y, z, t) = E_\xi [Q(x, y, z, t, \xi(\omega))] \tag{2}$$

is the *expected recourse function*, and

$$Q(x, y, z, t, \xi(\omega)) = \min \sum_{r=1}^{n_R} c^o O_r(\omega) + \sum_{(i,j): \sum_{k=1}^{n_S} b_{ijk} = 1} c^S I_{ij}(\omega) + \sum_{k=1}^{n_S} c^S I_k(\omega) \tag{3a}$$

s.t.

$$C_{ir}(\omega) \le M y_{ir} \quad \forall \omega, i, r, \tag{3b}$$

$$C_{jr}(\omega) \ge C_{ir}(\omega) + s^R + \text{pre}_j(\omega) + p_j(\omega) + \text{post}_j(\omega) - M(1 - z_{ijr}) \quad \forall \omega, i, j \neq i, r, \tag{3c}$$

$$\sum_{r=1}^{n_R} C_{i_k r}(\omega) = t_k + I_k(\omega) + \text{pre}_{i_k}(\omega) + p_{i_k}(\omega) + \text{post}_{i_k}(\omega) \quad \forall \omega, k, \tag{3d}$$

$$\sum_{r=1}^{n_R} C_{ir}(\omega) \ge t_k + \text{pre}_i(\omega) + p_i(\omega) + \text{post}_i(\omega) \quad \forall \omega, (i, k): \sum_{j=1}^{n} b_{jik} = 1, \tag{3e}$$

$$\sum_{r=1}^{n_R} C_{jr}(\omega) = \sum_{r=1}^{n_R} C_{ir}(\omega) - \text{post}_i(\omega) + s^S + p_j(\omega) + \text{post}_j(\omega) + I_{ij}(\omega) \quad \forall \omega, (i, j): \sum_{k=1}^{n_S} b_{ijk} = 1, \tag{3f}$$

$$O_r(\omega) \ge C_{ir}(\omega) - L \quad \forall \omega, i, r, \tag{3g}$$

$$C_{ir}(\omega), I_k(\omega), I_{ij}(\omega), O_r(\omega) \ge 0 \quad \forall \omega, i, j, r, k. \tag{3h}$$

The objective function (1a) is the sum of the first-stage cost and the expected second-stage cost over all scenarios. The first-stage cost is the fixed cost of opening ORs, and the second-stage costs are the sum of expected overtime costs and surgeon idle time costs. Note that the OR scheduling problem we consider in this study is a multicriteria problem, and each piece of the total operating cost defined by (1a) corresponds to a different performance measure.

Constraints (1b) and (1c) ensure that a surgery can be assigned to an OR only if it is opened and each surgery is assigned to exactly one OR, respectively. A precedence relation exists between two surgeries if and only if they are both assigned to the same OR and this is enforced by constraints (1d)–(1f). Constraint (1g) ensures that the starting time of each surgeon is no more than the session length. This constraint reflects an operationally meaningful assumption; if all of the surgeries of a surgeon are anticipated to be performed beyond the session length by using overtime, then it is more reasonable to schedule that particular surgeon's surgeries in another OR or on another day. As we have the upper bound L on the surgeon start times in the first stage, the surgery completion times in the second stage are also bounded. Constraints (1h) and (1i) define binary and nonnegativity restrictions for the first-stage decision variables, respectively.

The second-stage problem for a given x, y, z, t, and $\xi(\omega)$ is formulated explicitly by (3). The completion time of a surgery in an OR is 0 unless it is assigned to that OR, which is enforced by constraint (3b). Constraint (3c) defines the completion time of surgeries in ORs considering their precedence relation, processing times, and OR turnover time. The M parameter used in constraints (3b) and (3c) is an upper bound on the surgery completion times. Constraints (3d) and (3e) ensure that surgeries of a surgeon cannot be started before his or her arrival to the surgical suite. Constraint (3d) determines the idle time of the considered surgeon before his or her first surgery. Because the preincision of the first surgery needs to be started after the arrival of the surgeon, that portion is not included in the surgeon idle time as opposed to the preincision parts of the subsequent surgeries where the surgeon is considered to be idle unless he or she is performing the critical part of a surgery. Constraint (3f) provides the relation between surgery completion times, surgeon idle times, and the sequence of surgeries in surgeons' surgery listing. Constraint (3g) defines the overtime used in each OR. Constraints (3h) define nonnegativity restrictions for the second-stage decision variables.

Notice that we assume that the durations of all surgeries are realized at the beginning of the day of surgeries, and this is consistent with the limited recourse for schedule changes during the day (i.e., rescheduling of surgeries is not allowed). Although this can be considered as a limiting assumption for dynamic surgical environments where the surgeries are rescheduled during the day, we leave rescheduling for future research because it greatly complicates the problem.

It can be easily shown that the formulated stochastic multi-OR scheduling problem is NP-hard by reducing the bin-packing problem, which is known to be NP-hard, to a special case of our problem.

5. Solution Methods

The mathematical model we present is a two-stage SMIP with binary and continuous first-stage decision variables and purely continuous second-stage variables. We solve this SMIP by using the L-shaped method (Van Slyke and Wets 1969), which is an outer linearization approach in which a master problem (composed of the first-stage variables and constraints) is solved iteratively. Optimality and feasibility cuts, based on the solutions of the second-stage scenario subproblems, are used to approximate the recourse function (optimality cuts) and guarantee feasibility of the second-stage subproblems (feasibility cuts) throughout the iterations. The optimality cuts include a variable, θ, that represents the approximate recourse function, defining a progressively better lower bound on the recourse function at each iteration.

5.1. Antisymmetry Constraints

In this study, we consider the case (common in practice) in which the ORs are identical and therefore interchangeable. Given a solution, an equivalent solution can be obtained by swapping the set of surgeries assigned to any pair of ORs. Thus, our problem has complete symmetry with respect to ORs. While solving highly symmetric IP models, standard solution algorithms may need to explore many alternative symmetric solutions, which consumes too much computational time. Therefore, eliminating symmetric solutions while formulating and solving a problem may be beneficial (Sherali and Smith 2001, Margot 2002, Ostrowski et al. 2011). We add the following symmetry-breaking constraints, which are introduced by Denton et al. (2010), to the problem

$$x_r \geq x_{r+1} \quad r = 1, 2, \ldots, n_R - 1, \tag{4a}$$

$$\sum_{r=1}^{i} y_{ir} = 1 \quad i = 1, 2, \ldots, \min\{n, n_R\}, \tag{4b}$$

$$\sum_{q=r}^{\min\{i, n_R\}} y_{iq} \leq \sum_{j=r-1}^{i-1} y_{j,\,r-1} \quad \forall\, (i, r)\colon i \geq r. \tag{4c}$$

Constraint (4a) breaks the symmetry with respect to ORs by introducing an arbitrary ordering. Similarly, constraints (4b) and (4c) introduce a lexicographic order in terms of the indices of surgeries allocated to each OR. For example, if the first $i-1$ surgeries are assigned to the first $r-1$ ORs, then the ith surgery should be assigned to one of the first r ORs. Denton et al. (2010) observe that these constraints have a significant impact on the solution time for a stochastic version of the bin-packing problem.

5.2. Feasibility of the Second-Stage Problem

The extensive form of our two-stage recourse problem ensures feasible schedules, i.e., schedules that do not include cyclic surgery sequences or any other kind of infeasibilities. However, a decomposition method like the L-shaped method that solves the master and recourse problems separately may result in feasible first-stage solutions that are second-stage infeasible. This is because the completion time-related constraints (i.e., constraints (3b)–(3f)) are in the second stage. The standard L-shaped method (Van Slyke and Wets 1969) generates feasibility cuts to induce feasibility of first-stage solutions with respect to second-stage constraints. However, instead of generating feasibility cuts at each iteration of the L-shaped method (which may be very time consuming), we add the *induced constraints* introduced in Proposition 1 to the master problem a priori to induce *relatively complete recourse*.

Proposition 1. *A first-stage solution* (x, y, z, t) *is feasible for first- and second-stage problems if it satisfies* (1b)–(1i), (4), *and*

$$u_j \geq u_i + d - nd\left(1 - \sum_{r=1}^{n_R} z_{ijr}\right) \quad \forall i,\ j \neq i, \tag{5a}$$

$$u_j \geq u_i + d \quad \forall (i, j)\colon \sum_{k=1}^{n_S} b_{ijk} = 1, \tag{5b}$$

where u_i*s are nonnegative auxiliary first-stage decision variables and* d *is a positive finite scalar.*

By enforcing the difference of completion times of the surgeries that are scheduled within the same OR to be at least d, constraint set (5a) prevents infeasible schedules with respect to the sequence within an OR. In a similar way, constraint set (5b) ensures the feasibility of the constructed sequence with respect to surgeons across the ORs. As a result, constraints (5) ensure that z yields acyclic surgery sequences. Therefore, any first-stage feasible solution that also satisfies (5) is feasible for the second-stage problem under each scenario. Note that any positive finite scalar can be selected as d, and we choose $d = 1$ in our computational study.

5.3. Structure of Scenario Subproblems

Letting k denote the index of the surgeon who performs surgery i, the second-stage recourse problem can be solved in closed form as follows:

- If $y_{ir} = 0$, then $C_{ir}(\omega) = 0$.
- If $y_{ir} = 1$, then (i, r) pair falls into one of the following four categories and the corresponding $C_{ir}(\omega)$ takes a value accordingly:

1. If i is the first surgery in OR r and $i = i_k$, then

$$C_{ir}(\omega) = t_k + \text{pre}_i(\omega) + p_i(\omega) + \text{post}_i(\omega). \tag{6}$$

2. If i is the first surgery in OR r but $i \neq i_k$, then

$$C_{ir}(\omega) = \max\begin{cases} t_k + \text{pre}_i(\omega) + p_i(\omega) + \text{post}_i(\omega), \\ \displaystyle\sum_{j=1}^{n}\left[b_{jik}\left[\sum_{r=1}^{n_R} C_{jr}(\omega) - \text{post}_j(\omega)\right]\right] \\ \quad + s^S + p_i(\omega) + \text{post}_i(\omega). \end{cases} \tag{7}$$

3. If i is not the first surgery in OR r but $i = i_k$, then

$$C_{ir}(\omega) = \max\begin{cases} \max_j\{z_{jir} C_{jr}(\omega)\} + s^R + \text{pre}_i(\omega) \\ \quad + p_i(\omega) + \text{post}_i(\omega), \\ t_k + \text{pre}_i(\omega) + p_i(\omega) + \text{post}_i(\omega). \end{cases} \tag{8}$$

4. If i is not the first surgery in OR r and $i \neq i_k$, then

$$C_{ir}(\omega) = \max\begin{cases} t_k + \text{pre}_i(\omega) + p_i(\omega) + \text{post}_i(\omega), \\ \max_j\{z_{jir} C_{jr}(\omega)\} + s^R + \text{pre}_i(\omega) \\ \quad + p_i(\omega) + \text{post}_i(\omega), \\ \displaystyle\sum_{j=1}^{n}\left[b_{jik}\left[\sum_{r=1}^{n_R} C_{jr}(\omega) - \text{post}_j(\omega)\right]\right] \\ \quad + s^S + p_i(\omega) + \text{post}_i(\omega). \end{cases} \tag{9}$$

Given the values of the $C_{ir}(\omega)$ variables, the remaining decision variable values can be expressed as

$$I_{ij}(\omega) = \sum_{r=1}^{n_R} C_{jr}(\omega) - \sum_{r=1}^{n_R} C_{ir}(\omega) + \text{post}_i(\omega) - s^S - p_j(\omega)$$
$$-\text{post}_j(\omega) \quad \forall (i, j)\colon \textstyle\sum_{k=1}^{n_S} b_{ijk} = 1, \tag{10}$$

$$I_k(\omega) = \sum_{r=1}^{n_R} C_{i_k r}(\omega) - t_k - \text{pre}_{i_k}(\omega)$$
$$-p_{i_k}(\omega) - \text{post}_{i_k}(\omega) \quad \forall k, \tag{11}$$

$$O_r(\omega) = \max\left\{0, \max_i\{C_{ir}(\omega)\} - L\right\} \quad \forall r. \tag{12}$$

Using the above equations, we obtain the optimal solution to the primal subproblem. We use the optimal primal solution as the initial solution and solve the subproblem to get the dual solution so as to generate the optimality cuts.

5.4. Extended Master Problem Formulation

The following is an equivalent formulation of our problem:

$$\min \sum_{r=1}^{n_R} c^f x_r + \theta \tag{13}$$

$$\text{s.t.}\quad \theta \geq \mathcal{Q}(x, y, z, t), \tag{14}$$

$$(1\text{b})\text{–}(1\text{i}), (4), (5).$$

The standard L-shaped algorithm starts by solving the initial restricted master problem (RMP), which is

$$\min \sum_{r=1}^{n_R} c^f x_r + \theta$$

$$(1\text{b})\text{–}(1\text{i}), (4), (5).$$

A stopping criterion is used to determine if the RMP results in the minimum expected second-stage cost. If it does not, duality is employed to generate a corresponding optimality cut, which includes the first-stage variables and the linking variable θ. Iterations continue until the optimal solution is reached.

Our initial computational experiments revealed that the standard L-shaped algorithm fails to solve even small problem instances within a reasonable amount of time. The main reason is that the θ variable carries only limited information between first and second stages (Ruszczyński 1986, Kiwiel 1990, Smith et al. 2004). Because of this, the solutions generated by solving the RMP usually have high expected second-stage cost, and hence the lower and upper bounds converge to the optimal solution very slowly. To deal with this issue, we propose a novel way to strengthen the formulation by including a lower bounding inequality for θ in the first stage, based on the following proposition.

Proposition 2. *Let $(\hat{x}, \hat{y}, \hat{z}, \hat{t})$ and $Q(\hat{x}, \hat{y}, \hat{z}, \hat{t}, \bar{\xi}(\omega))$ be a feasible first-stage solution of our problem and the corresponding second-stage cost under the mean value scenario, respectively. Then,*

$$\theta \geq Q\big(\hat{x}, \hat{y}, \hat{z}, \hat{t}, \bar{\xi}(\omega)\big). \tag{15}$$

Proof. For any given feasible first-stage solution, the second-stage subproblems are feasible and bounded. Then, we have

$$\mathcal{Q}(\hat{x}, \hat{y}, \hat{z}, \hat{t}) \geq Q\big(\hat{x}, \hat{y}, \hat{z}, \hat{t}, \bar{\xi}(\omega)\big) \tag{16}$$

by Jensen's inequality (Jensen 1906). Moreover, we have

$$\theta \geq \mathcal{Q}(\hat{x}, \hat{y}, \hat{z}, \hat{t}) \tag{17}$$

because $\theta \geq \mathcal{Q}(x, y, z, t)$ is a part of our formulation (Equation (14)); (15) directly follows from (16) and (17). □

We observe that these cuts are broadly applicable to two-stage stochastic programs with recourse. We use valid inequalities based on Proposition 2 to speed up the convergence of the L-shaped algorithm. The following are additional parameters and auxiliary decision variables we use and the lower bounding inequality we propose.

Additional Parameters

$\overline{\text{pre}}_i$: expected preincision duration of surgery i.
$\bar{p}_i$: expected incision duration of surgery i.
$\overline{\text{post}}_i$: expected postincision duration of surgery i.

Auxiliary Decision Variables

C_{ir}: completion time for surgery i in OR r under the mean value scenario.
I_{ij}: surgeon idle time between surgeries i and j under the mean value scenario (defined for (i, j): $\sum_{k=1}^{n_S} b_{ijk} = 1$).
I_k: idle time of surgeon k before his or her first surgery under the mean value scenario.
O_r: overtime in OR r, with respect to session length L under the mean value scenario.

Proposition 3. *Let variables C_{ir}, I_{ij}, I_k, and O_r be defined by the following inequalities*:

$$C_{ir} \leq M y_{ir} \quad \forall\, i,\ r, \tag{18a}$$

$$C_{jr} \geq C_{ir} + s^R + \overline{\text{pre}}_j + \bar{p}_j + \overline{\text{post}}_j - M(1 - z_{ijr}) \quad \forall\, i,\ j \neq i, r, \tag{18b}$$

$$\sum_{r=1}^{n_R} C_{i_k r} = t_k + I_k + \overline{\text{pre}}_{i_k} + \bar{p}_{i_k} + \overline{\text{post}}_{i_k} \quad \forall k, \tag{18c}$$

$$\sum_{r=1}^{n_R} C_{ir} \geq t_k + \overline{\text{pre}}_i + \bar{p}_i + \overline{\text{post}}_i \quad \forall (i, k)\colon \sum_{j=1}^{n} b_{jik} = 1, \tag{18d}$$

$$\sum_{r=1}^{n_R} C_{jr} = \sum_{r=1}^{n_R} C_{ir} - \overline{\text{post}}_i + s^S + \bar{p}_j + \overline{\text{post}}_j + I_{ij} \quad \forall (i, j)\colon \sum_{k=1}^{n_S} b_{ijk} = 1, \tag{18e}$$

$$O_r \geq C_{ir} - L \quad \forall i, r, \tag{18f}$$

$$C_{ir}, I_k, I_{ij}, O_r \geq 0 \quad \forall i, j, r, k. \tag{18g}$$

Then,

$$\theta \geq \sum_{r=1}^{n_R} c^o O_r + \sum_{(i, j)\colon \sum_{k=1}^{n_S} b_{ijk} = 1} c^S I_{ij} + \sum_{k=1}^{n_S} c^S I_k \tag{19}$$

is a valid inequality for the RMP and hence can be added to the RMP together with (18).

Proof. This result directly follows from the validity of Proposition 2 for every feasible first-stage solution and the definition of mean value scenario, additional parameters, and auxiliary variables. □

Then, the initial RMP used in the L-shaped algorithm becomes the following *extended RMP* (*ERMP*):

$$\min \ \sum_{r=1}^{n_R} c^f x_r + \theta$$
$$\text{s.t.} \quad (1\text{b})\text{–}(1\text{i}), (4), (18), (19).$$

Note that constraints (18) ensure the second-stage feasibility of the first-stage solutions by eliminating the schedules that include cyclic surgery sequences, so we do not need to include the induced constraints (5) in the ERMP.

Valid inequalities in stochastic programming is explored by earlier work, including Laporte et al. (1994), Guan et al. (2006), Sanchez and Wood (2006), and Guan et al. (2009). Sanchez and Wood use inequalities based on Jensen (1906) within the simulation-based approach they propose for solving two-stage SMIPs. However, their method assumes a binary first stage and may require full enumeration of every feasible first-stage solution.

6. Computational Results

In this section, we first give some information on the data set we use to generate realistic problem instances for computational experiments. Next, we compare the performance of the algorithms we propose and discuss the value of capturing uncertainty. Finally, we estimate the value of OR pooling and illustrate the impact of parallel surgery processing.

6.1. Parameter Estimation

We use data from the Mayo Clinic's Division of General Thoracic Surgery at St. Marys Hospital in Rochester, MN. The department consists of six surgeons and their surgical residents, along with support staff (including nurses), and it performs more than 2,000 thoracic procedures per year. The surgeons provide comprehensive diagnosis and surgical care to adult patients with diseases of the lungs, trachea, esophagus, diaphragm, chest wall, and mediastinum. Thoracic surgery often consists of several separate subprocedures. Surgeons within the thoracic area perform surgeries every other day, usually with at least two staff surgeons working each day. Because the Mayo Clinic is an academic teaching institution, each staff surgeon may have multiple fellows assisting in the procedures, with the primary staff surgeon performing the critical part of each surgery.

Our parameter estimations are based on the historical data provided by the thoracic surgery department of the Mayo Clinic and our discussions with an anesthesiologist who works as an administrative director in the thoracic surgery department. The daily fixed cost of opening an OR, c^f, is estimated to be \$4,437. The session length, L, is nine hours per day for an OR. The overtime cost, c^o, is estimated to be \$12.37 per minute, which is 50% higher than the regular OR time cost. Because we could not directly estimate the cost of the surgeon idle time exactly, based on our discussion with the administrative director, we use two different levels of idle time cost to evaluate its effect. In the first case, we assume that scheduling 50 minutes of surgeon idle time is equivalent to opening another OR and incurring its fixed cost. For this case, surgeon idle time is \$88.74 per minute. In the second case, we assume that 250 minutes of idle time is equivalent to opening an OR and use \$17.748 per minute as the surgeon idling cost. We refer to these as high and low idle time costs, respectively.

The setup activities between two consecutive surgeries were reported to be completed within 30 minutes, which corresponds to OR turnover time in our formulation. Because it is reported to be very short, we assume that surgeon turnover time is 0 in our computational study. Our problem instances are based on 322 actual surgical days realized in the Division of General Thoracic Surgery at St. Marys Hospital. For each day, the following information is retrieved from the realized schedule and used as input:

- The number of surgeries, surgeons, and available ORs,
- The number and type of subprocedures in each surgery, and
- The ordered surgery listing of each surgeon.

For each surgical day, i.e., problem instance, we generate 500 different scenarios by sampling preincision, incision, and postincision durations for each surgery based on the number and type of the subprocedures included.

We estimated the probability distributions of the surgery durations from historical data. Thoracic surgeries involve several distinct subprocedures performed in sequence. Unfortunately, times for each subprocedure are not collected; only the start and stop times for the preincision, incision, and postincision were available. However, we are able to identify which subprocedures were performed during each surgery. We observe that the surgeries contain at most five combinations of the 21 most common subprocedures. The duration of a subprocedure is highly dependent on the complexity of the surgery, which is closely related to the number of subprocedures included. For example, if one surgery contains subprocedures 1, 5, and 6, and another surgery contains 1, 10, and 12, the time to carry out subprocedure 1 should be similar because they both include three subprocedures. To determine this relationship, we used a multiple regression model implementing a bootstrap method to estimate the probability distribution of the individual subprocedure durations.

The incision duration includes the critical portion of surgery, which is completed by the primary surgeon, as well as the noncritical portion of surgery. We used a discrete event simulation model (Huschka et al. 2007) to estimate the implied duration of the critical portion of surgery. The resources in our simulation model include ORs and surgeons, and the entities in the model were the patients. Similar to our stochastic programming model, the focus of the simulation model is also on the preincision, incision, and postincision aspects of the surgery.

We compared the results for various assumptions about the percentage of incision duration that comprises the critical portion of surgery. For our comparison, we considered the amount of overtime used to

Table 1 Size-Based Classification of Problem Instances for 322 Surgical Days

			Average				Maximum			
Set no.	n_{Bin}	No. of days	n	n_S	n_R	n_{Bin}	n	n_S	n_R	n_{Bin}
1	[1, 100]	177	3.95	1.76	2.64	41.16	7	3	4	99
2	[101, 200]	67	6.79	2.49	3.93	149.09	9	3	5	200
3	[201, 300]	46	8.04	2.43	4.39	234.74	9	3	6	280
4	[301, 400]	17	9.29	2.59	4.88	338.88	11	4	6	390
5	[401, 500]	6	10.17	2.83	5.17	436.83	11	3	6	485
6	[501, 600]	7	11.00	2.57	5.71	550.57	12	3	7	600
7	[601, 700]	2	11.00	3.00	6.00	609.00	11	3	6	612

complete the surgeries. If the overtime levels were close to the actual observations, we assumed that the estimates were reasonable. Based on this analysis, approximately 25% of the incision duration is estimated to be critical. Therefore, we decreased the incision duration to 25% of its initial value to estimate the duration of the critical portion of the surgery. We reallocated remaining time to the preincision and postincision durations evenly.

The problem size for each surgical day, i.e., problem instance, depends on the number of surgeries, surgeons, and available ORs. We classify the problem instances into seven sets based on the number of binary variables, which we denote as n_{Bin}. We present the number of problem instances, average and maximum number of surgeries (n), surgeons (n_S), available ORs (n_R), and binary variables (n_{Bin}) for each set in Table 1. More than 90% of the problem instances are included in the first four sets. The remaining 10% are considered as large instances. The largest instance, which is an instance in set 7, includes 612 binary variables, and it corresponds to a surgical day that involves 11 surgeries, three surgeons, and six ORs.

6.2. Computational Performance of the Proposed Algorithms

We analyze the performance of the standard L-shaped algorithm (Figure 4) with different master problem formulations: RMP and ERMP. The main drawback of the standard L-shaped algorithm is that it solves the master problem, which is a mixed-integer program (MIP), to optimality at each iteration. This requires significant computational effort that may not be productive at early iterations when few optimality cuts have been added to the master problem. In the second approach, we implement the L-shaped algorithm within a branch-and-cut framework (Figure 5), adding optimality cuts at each integer-feasible node. This approach solves the master problem only once, by adding the optimality cuts during branch

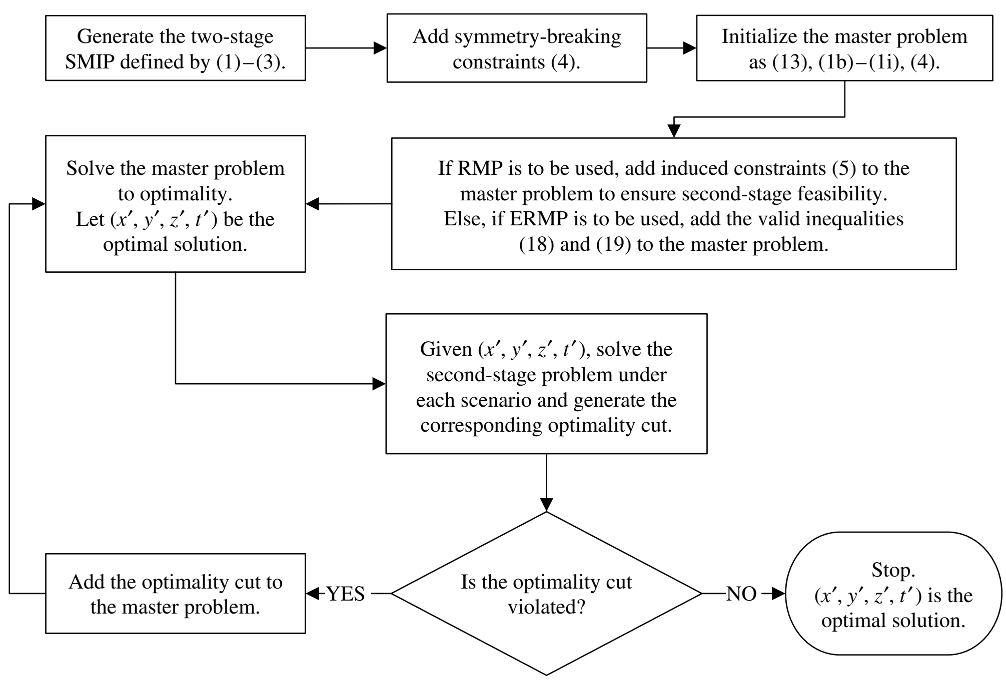

Figure 4 Flow Chart for Our Solution Method That Uses the L-Shaped Algorithm

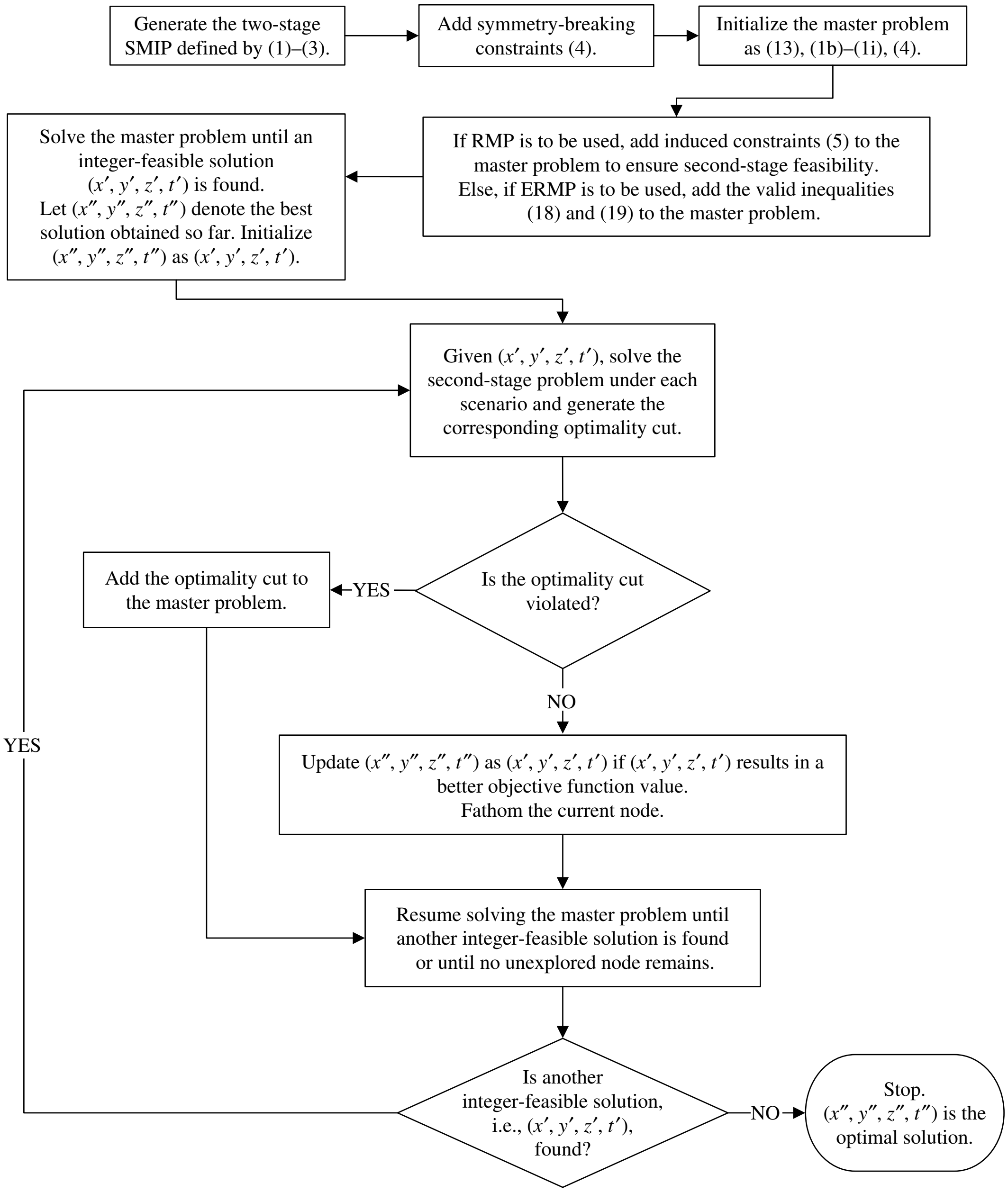

Figure 5 Flow Chart for Our Solution Method That Uses the L-Shaped-Based Branch-and-Cut Algorithm

and bound. We test our branch-and-cut approach with both of the master problem formulations, RMP and ERMP.

We coded our algorithms in Microsoft Visual Studio .NET 2003 using the CPLEX 11 callable library. We conducted our experiments on an Intel Core2 Duo PC with processors running at 3.17 GHz and 2 GB memory under Windows XP. To compare the computational performances of the proposed methods, we randomly choose 100 instances, i.e., surgical days, from the first three sets. We report the average and maximum solution times (in CPU seconds) and the number of iterations in Table 2. Sets 2 and 3 include problem instances that could not be solved in three hours by the algorithms with RMP formulation (these instances were solved within a reasonable amount of time using the ERMP formulation). For these unsolved instances, we consider the solution time as three hours, i.e., the computational time limit, when calculating the average solution time.

As can be observed from Table 2, the standard L-shaped algorithm with ERMP performs best. Regardless of the solution algorithm used, ERMP significantly outperforms RMP. Therefore, we conclude that adding valid inequalities (18) and (19) to the master problem improves the formulation considerably.

Table 2 Computational Performance of the L-Shaped and L-Shaped-Based Branch-and-Cut Algorithms

			L-shaped algorithm								L-shaped-based branch-and-cut algorithm			
			RMP				ERMP				RMP		ERMP	
			Solution time (CPU seconds)		No. of iterations		Solution time (CPU seconds)		No. of iterations		Solution time (CPU seconds)		Solution time (CPU seconds)	
Idle time cost level	Set no.	No. of instances	Average	Maximum	Average	Maximum	Average	Maximum	Average	Maximum	Average	Maximum	Average	Maximum
Low	1	60	8.76	60.30	69.23	338	1.51	14.95	14.32	82	12.81	77.33	4.55	34.94
	2	25	1,689.12	>10,800.00	1,025.80	4,226	20.16	123.48	22.48	52	776.13	6,141.42	49.48	249.28
	3	15	7,784.51	>10,800.00	2,396.53	3,642	308.23	1,701.73	38.73	84	7,158.44	>10,800.00	1,016.65	10,146.41
High	1	60	7.77	44.69	63.67	275	1.79	13.72	17.08	92	14.80	211.90	6.84	94.31
	2	25	968.23	>10,800.00	618.00	3,373	24.52	90.34	41.44	142	649.23	3,560.53	95.75	1,281.81
	3	15	5,923.05	>10,800.00	1,874.53	3,428	289.21	843.02	66.80	152	6,895.76	>10,800.00	424.68	1,449.48

Another conclusion from Table 2 is that the idle time cost level does not have an impact on the relative performance of the algorithms.

In Table 3, we report the solution times and number of iterations for 20 of the randomly selected 100 instances under the high idle time cost setting. When we look at the solution times of these problem instances, we observe that there are a couple of instances (2c and 3b) for which the L-shaped-based branch-and-cut algorithm with ERMP outperforms all other algorithms, although on average the standard L-shaped algorithm with ERMP performs best.

We conclude that the standard L-shaped algorithm with the ERMP formulation is superior, and hence we use it to solve the remaining instances of our problem. However, for larger instances that cannot be solved optimally within the three-hour time limit, we also generate a solution by using the L-shaped-based branch-and-cut algorithm with the ERMP formulation by imposing the same time limit. We use the best solution—i.e., the solution with the lower objective function value—generated by these two methods.

We solve the mean value problem and the stochastic problem for each surgical day under low and high surgeon-idling cost levels, and we report the average and maximum solution times in Table 4. For some of the larger instances in sets 5, 6, and 7, although we are able to solve the mean value problems within three hours, we are not able to solve the stochastic problems. The number of unsolved

Table 3 Computational Performance of the Proposed Algorithms for 20 Instances

	L-shaped algorithm				L-shaped-based branch-and-cut algorithm	
	RMP		ERMP		RMP	ERMP
Instance no.	Solution time (CPU seconds)	No. of iterations	Solution time (CPU seconds)	No. of iterations	Solution time (CPU seconds)	Solution time (CPU seconds)
1a	2.16	27	0.17	2	2.97	0.61
1b	7.83	82	1.70	15	20.08	94.31
1c	43.83	275	4.17	21	61.17	11.00
1d	11.47	71	0.39	2	13.83	2.75
1e	15.41	110	8.56	53	41.89	15.74
1f	1.41	32	0.75	17	3.17	1.31
1g	0.83	20	0.39	9	2.92	1.55
1h	0.234	11	0.312	13	0.688	1.109
1i	33.75	227	11.59	77	53.19	22.03
1j	0.97	35	0.47	13	1.50	1.36
2a	44.55	167	3.63	13	90.02	13.97
2b	167.90	441	21.50	57	178.49	41.41
2c	>10,800.00	3,373	90.34	32	2,112.92	45.00
2d	322.34	349	13.17	14	550.86	68.44
2e	506.20	754	13.16	21	350.47	41.08
3a	3,652.47	1,577	32.72	19	9,843.77	185.44
3b	>10,800.00	2,978	839.92	80	>10,800.00	716.06
3c	2,491.17	1,040	246.03	133	2,018.11	328.69
3d	1,930.78	1,037	35.22	22	5,440.00	322.14
3e	>10,800.00	3,428	97.98	13	>10,800.00	235.16

Table 4 Solution Times (in CPU Seconds) of Mean Value and Stochastic Problems

	Low idle time cost				High idle time cost			
	Mean value problem		Stochastic problem		Mean value problem		Stochastic problem	
Set no.	Average	Maximum	Average	Maximum	Average	Maximum	Average	Maximum
1	0.02	0.11	1.11	15.03	0.01	0.11	1.35	16.53
2	0.23	1.63	26.30	200.10	0.13	0.72	33.94	311.89
3	1.50	12.99	238.30	2,517.33	0.59	3.80	212.23	1,981.09
4	11.31	66.58	1,023.36	5,097.61	2.84	14.86	1,540.80	8,034.84
5	46.02	128.42	2,160.37	2,969.52	11.80	35.50	6,078.25	>10,800.00
6	196.13	687.18	5,241.69	>10,800.00	34.55	141.02	4,866.16	>10,800.00
7	126.78	151.83	9,447.07	>10,800.00	6.27	8.13	9,992.85	>10,800.00

Table 5 Percentage Gap Values for the Unsolved Instances

	Low idle time cost			High idle time cost		
		Gap between lower and upper bounds (%)			Gap between lower and upper bounds (%)	
Set no.	No. of unsolved instances	Average	Maximum	No. of unsolved instances	Average	Maximum
5	—	—	—	2	1.36	2.17
6	1	2.41	2.41	2	3.75	5.88
7	1	1.85	1.85	1	4.03	4.03

instances and the percentage optimality gap between the upper bound—the value of the best solution obtained within three hours—and the lower bound—the maximum of the bounds obtained by the standard L-shaped algorithm with ERMP and the L-shaped-based branch-and-cut algorithm with ERMP—are reported in Table 5. Of particular interest is the average percentage gap, which is below 3% and 5% for low and idle time costs, respectively.

6.3. Value of the Stochastic Solution

To assess the value of capturing uncertainty in surgery durations, we estimate the value of the stochastic solution (VSS), the difference between the optimal objective function value of the stochastic problem and the expected objective function value of the optimal solution of the mean value problem (Birge and Louveaux 1997). As for the instances whose stochastic problem formulations cannot be solved within the allowed time limit, we consider the value of the best solution obtained in our comparisons. We report the average and maximum improvement brought by solving the stochastic problem in Table 6. The average improvement when the idle time cost is high (low) is more than 4% (less than 1%) for most (all) of the data sets. Maximum VSS values in Table 6 imply that there are problem instances where the improvement is more than 9% and 28% when the idle time is low and high, respectively. We conclude that capturing the uncertainty is particularly important when the cost of idle time is high. Observing higher VSS values for high idle time costs is intuitive because having higher values of second-stage cost coefficients implies that the impact of a realized scenario would be more significant.

The total expected operating cost, which is the objective function in our formulation, is composed of three pieces, each of which is related to a different performance criterion. By solving the stochastic problem rather than the mean value problem, we are able to generate schedules with lower total expected operating costs. Because we are considering a multicriteria problem, a decrease in the objective function value does not necessarily imply that the schedule gets better in terms of all performance measures considered. Instead, it means that we are able to obtain a nondominated solution with lower objective function value. To see the impact of capturing uncertainty on the performance measures of our concern, we summarize the average number of open ORs, overtime per OR, and idle time per surgeon of the schedules generated by solving the mean value and stochastic problems

Table 6 Percentage Value of the Stochastic Solution for Each Problem Set

	Low idle time cost (%)		High idle time cost (%)	
Set no.	Average	Maximum	Average	Maximum
1	0.95	9.34	4.20	28.41
2	0.52	3.15	4.10	13.69
3	0.87	3.54	4.24	12.30
4	0.53	1.71	3.43	13.30
5	0.93	3.35	4.01	7.52
6	0.54	2.40	2.46	5.35
7	0.54	0.87	7.19	8.00

Table 7 Optimal Solution Statistics for the Stochastic and Mean Value Problems for Each Problem Set

		Low idle time cost		High idle time cost	
Set no.		Stochastic problem	Mean value problem	Stochastic problem	Mean value problem
Average no. of open ORs	1	2.07	2.06	2.34	2.34
	2	3.15	3.09	3.76	3.72
	3	3.80	3.80	4.04	4.04
	4	4.18	4.24	4.59	4.47
	5	4.50	4.50	4.83	4.83
	6	5.14	5.14	5.29	5.29
	7	5.00	5.00	6.00	5.00
Average overtime per OR (in minutes)	1	23.00	24.92	20.34	18.54
	2	49.39	54.08	42.89	40.21
	3	60.48	64.38	65.39	56.92
	4	53.85	54.67	53.47	53.03
	5	107.72	105.49	138.25	128.04
	6	74.74	71.35	89.81	81.89
	7	79.62	78.95	49.92	89.21
Average idle time per surgeon (in minutes)	1	85.31	89.33	57.29	60.31
	2	59.07	63.25	27.75	33.53
	3	40.61	40.27	21.53	28.05
	4	47.66	42.35	22.61	28.80
	5	61.23	66.55	26.45	33.32
	6	44.31	52.03	27.51	33.25
	7	27.44	30.89	3.97	22.83

in Table 7. For high idle time costs, the solutions to stochastic problems have lower values of expected idle time over all of the sets and higher values of overtime and numbers of open ORs for a majority of the sets. Therefore, for high idle time costs, we conclude that the total cost reduction achieved by solving the stochastic problem is mostly attributable to the decrease in the average idle time values. For low idle time costs, we are able to observe the multicriteria structure of the problem more explicitly. The improvements in the number of open ORs and overtime values play a significant role in the total cost reduction for sets 1–4, whereas the decrease in idle time still remains the only factor that lowers the objective function value for sets 5–7.

6.4. Value of OR Pooling

OR pooling, which is allowed in our model, occurs when the surgeries of different surgeons are allowed to be scheduled in the same OR. In this section, we quantify the benefit of OR pooling by comparing two implementation settings:

- Setting 1: Our original model, in which ORs are pooled as a shared resource.
- Setting 2: A restricted setting where OR pooling is not allowed.

In setting 2, we consider a modified version of our model that prevents the sharing of ORs among surgeons by using the following first-stage constraints:

$$y_{ir} + y_{jr} \le 1 \quad \forall\, r,\, (i, j > i)\colon \sum_{k=1}^{n_S} (\beta_{ijk} + \beta_{jik}) = 0, \tag{20}$$

where β_{ijk}, which can be directly obtained from b_{ijk}s, is a binary parameter denoting whether surgery i precedes surgery j in surgeon k's listing; $\beta_{ijk} = 1$ if there exists a sequence of surgeries of surgeon k that begins with surgery i and ends with surgery j, and every surgery in the sequence immediately precedes the next one according to surgeon k's listing. Then, constraint (20) ensures that surgeries i and j cannot be scheduled in the same OR if they are not operated by the same surgeon. This implies that the corresponding surgeons cannot share the same OR. Note that when OR pooling is not allowed, the OR scheduling problem has a feasible solution only if $n_R \ge n_S$. This is satisfied by all of our instances.

Table 8 Percentage Improvement Brought by OR Pooling

	Low idle time cost (%)		High idle time cost (%)	
Set no.	Average	Maximum	Average	Maximum
1	22.22	53.03	34.19	82.21
2	29.56	49.71	51.90	76.31
3	29.12	46.53	58.65	77.63
4	28.52	46.96	55.92	74.78
5	27.85	35.61	55.93	64.64
6	21.78	34.41	50.04	68.22
7	22.59	27.27	54.85	55.47

Table 9 Optimal Solution Statistics for Settings 1 and 2 for Each Problem Set

		Low idle time cost		High idle time cost	
Set no.		Setting 1	Setting 2	Setting 1	Setting 2
Average no. of open ORs	1	2.07	2.45	2.34	2.63
	2	3.15	3.70	3.76	3.93
	3	3.80	4.20	4.04	4.39
	4	4.18	4.76	4.59	4.88
	5	4.50	5.00	4.83	5.17
	6	5.14	5.29	5.29	5.71
	7	5.00	6.00	6.00	6.00
Average overtime per OR (in minutes)	1	23.00	22.01	20.34	20.83
	2	49.39	56.26	42.89	55.79
	3	60.48	53.88	65.39	55.10
	4	53.85	53.05	53.47	54.43
	5	107.72	109.37	138.25	110.76
	6	74.74	77.84	89.81	91.98
	7	79.62	26.13	49.92	26.13
Average idle time per surgeon (in minutes)	1	85.31	137.28	57.29	114.67
	2	59.07	172.63	27.75	154.03
	3	40.61	208.47	21.53	193.85
	4	47.66	199.43	22.61	191.97
	5	61.23	239.11	26.45	225.39
	6	44.31	210.99	27.51	176.04
	7	27.44	153.42	3.97	153.42

Low idle time cost

OR turnover time coefficient	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
2.0	2	2	2	2	2	3	3	3	3	3	3
1.8	1	2	2	2	2	3	3	3	3	3	3
1.6	1	2	2	2	2	3	3	3	3	3	3
1.4	1	2	2	2	2	3	3	3	3	3	3
1.2	1	2	2	2	2	3	3	3	3	3	3
1.0	1	1	2	2	2	2	3	3	3	3	3
0.8	1	1	2	2	2	2	3	3	3	3	3
0.6	1	1	2	2	2	2	3	3	3	3	3
0.4	1	1	1	2	2	2	3	3	3	3	3
0.2	1	1	1	2	2	2	3	3	3	3	3
0.0	1	1	1	2	2	2	2	3	3	3	3

Parallelizable portion coefficient

High idle time cost

OR turnover time coefficient	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
2.0	2	2	2	3	3	3	3	4	4	5	6
1.8	2	2	2	2	3	3	3	3	4	5	6
1.6	2	2	2	2	3	3	3	3	4	4	5
1.4	2	2	2	2	3	3	3	3	4	4	5
1.2	2	2	2	2	3	3	3	3	4	4	5
1.0	2	2	2	2	3	3	3	3	4	4	5
0.8	2	2	2	2	2	3	3	3	4	4	5
0.6	2	2	2	2	2	3	3	3	4	4	5
0.4	1	2	2	2	2	3	3	3	3	4	4
0.2	1	2	2	2	2	3	3	3	3	4	4
0.0	1	2	2	2	2	3	3	3	3	4	4

Parallelizable portion coefficient

Figure 6 Optimal Number of ORs to Open at Different Levels of OR Turnover Time and Parallelizable Portion of Surgeries

By comparing the optimal objective function values obtained by solving the problem under these two settings, we evaluate the percentage reduction in the expected total cost as a result of OR pooling. We summarize the percentage improvements brought by OR pooling in Table 8. In Table 9, we compare the average number of open ORs, overtime values, and idle time values of the schedules generated by settings 1 and 2.

We observe from Table 8 that the average benefit gained from OR pooling is more than 21% and 34% for low and high idle time costs, respectively. The results provided in Table 9 reveal that the substantial cost reduction achieved by OR pooling is mainly attributable to the decrease in the average number of open ORs. The decrease in the required number of ORs might result in significant savings because the initial investment needed to build and open a new OR is estimated between \$700,000 and \$9,000,000 (Greene 2004, 2006; Yee 2007; Zinn 2009).

6.5. Impact of Parallel Surgery Processing

The impact of parallel surgery processing is closely related with the duration of the parallelizable portion of the surgery (i.e., preincision and postincision durations) as well as the length of the OR turnover time. As the parallelizable portion and OR turnover time increase, the potential benefits of parallel surgery processing becomes higher; hence, opening more ORs becomes favorable. To demonstrate this, we consider a surgical day that includes six surgeries, one surgeon and six available ORs. We consider different levels of OR turnover time, in a range changing from zero to two times the original turnover time (which is 30 minutes). As for the parallelizable portion of the surgery, we consider a range from zero to one times the original duration. The original parallelizable portions of the surgeries are, on average, more than 80% of the total surgery duration in the considered example. We generate optimal schedules for the selected levels of OR turnover time and the parallelizable portion of surgeries for both low and high idle time costs. Figure 6 illustrates the number of open ORs in the optimal schedule. As the parallelizable portion or the OR turnover time increases, the optimal number of open ORs also increases. Moreover, for a given pair of OR turnover time levels and parallelizable portion levels, the optimal number of open ORs is higher when the surgeon idling cost is higher. This shows that the impact of parallel surgery processing becomes more significant as the surgeon idle time cost increases.

7. Conclusions

We consider the problem of scheduling surgeries with uncertain durations in a multi-OR environment. The decisions in our model are the number of ORs to open, the allocation of surgeries to ORs, the sequence of surgeries within each OR, and the times at which surgeons start their first surgery of the day. Our model minimizes the sum of the fixed cost of opening ORs, the overtime cost, and the surgeon idling cost. We formulate the problem as a two-stage SMIP, where OR opening, surgery allocation and sequencing, and start-time decisions are made in the first stage (prior to the day of surgeries), and the OR overtime and surgeon idle time values are realized in the second stage, after the actual surgery durations become known. We explicitly consider the different phases of the surgeries (preincision, incision, and postincision), which allows us to evaluate the impact of parallelization of a particular surgeon's surgeries.

We analyze the properties of our model and present a set of induced feasibility constraints and a set of new valid inequalities based on Jensen's inequality so as to increase its solvability. Our results show that adding the proposed valid inequalities decreases the solution times of the standard L-shaped and L-shaped-based branch-and-cut algorithms significantly. Our results also indicate that the L-shaped algorithm tends to perform better than the L-shaped-based branch-and-cut algorithm. We solve both the stochastic and mean value problems, and we estimate the value of capturing uncertainty in surgery durations by comparing the obtained solution value of the schedules. Our results reveal that the value of capturing uncertainty is particularly significant for high idle time costs (around 4% on average and as high as 28%).

We draw some important managerial insights from our numerical results. Examples, based on real data collected from the Mayo Clinic in Rochester, MN, illustrate that the potential benefits of parallel surgery processing increases; hence, opening more ORs becomes favorable as the OR turnover time and parallelizable portion of surgeries increases. We solve our problem under different resource usage schemes, and we observe from our computational results that OR pooling leads to total cost reductions between 21.78% and 58.65% on average. Thus, OR pooling can lead to substantial cost reduction in some cases.

Our comparison of different resource usage schemes is based on the total expected operating cost. As a result, our analysis is dependent on the specific cost coefficients that weight the multiple criteria in the objective function. We leave an explicit treatment of this multicriteria optimization problem for future work. However, we note that our model, methodological results, and general insights are relevant to most providers of surgical care.

Acknowledgments

This study was funded in part by Grants CMMI-0620573, CMMI-0620780, and CMMI-0844511 from the National Science Foundation, as well as a grant from the U.S. Department of Veterans Affairs. The authors thank Hari Balasubramanian for his valuable contributions during the earlier phases of their study. They are grateful for the insightful comments of Andrew Trapp, Osman Özaltın, and Nan Kong that helped them improve this manuscript.

References

Adan, I., J. Bekkers, N. Dellaert, J. Vissers, X. Yu. 2009. Patient mix optimisation and stochastic resource requirements: A case study in cardiothoracic surgery planning. *Health Care Management Sci.* **12**(2) 129–141.

Beliën, J., E. Demeulemeester. 2007. Building cyclic master surgery schedules with leveled resulting bed occupancy. *Eur. J. Oper. Res.* **176**(2) 1185–1204.

Birge, J. R., F. Louveaux. 1997. *Introduction to Stochastic Programming*. Springer, New York.

Blake, J. T., M. W. Carter. 1997. Surgical process scheduling: A structured review. *J. Soc. Health Systems* **5**(3) 17–30.

Cardoen, B., E. Demeulemeester, J. Beliën. 2010. Operating room planning and scheduling: A literature review. *Eur. J. Oper. Res.* **201**(3) 921–932.

Centers for Medicare and Medicaid Services. 2007. National health expenditures data. Retrived January 28, 2009, http://www.cms.hhs.gov/NationalHealthExpendData/downloads/tables.pdf.

Dantzig, G. B. 1955. Linear programming under uncertainty. *Management Sci.* **1**(3/4) 197–206.

Denton, B. T., D. Gupta. 2003. A sequential bounding approach for optimal appointment scheduling. *IIE Trans.* **35**(11) 1003–1016.

Denton, B. T., J. Viapiano, A. Vogl. 2007. Optimization of surgery sequencing and scheduling decisions under uncertainty. *Health Care Management Sci.* **10**(1) 13–24.

Denton, B. T., A. J. Miller, H. J. Balasubramanian, T. R. Huschka. 2010. Optimal allocation of surgery blocks to operating rooms under uncertainty. *Oper. Res.* **58**(4, Part 1) 802–816.

Dexter, F., J. Ledolter. 2005. Bayesian prediction bounds and comparisons of operating room times even for procedures with few or no historic data. *Anesthesiology* **103**(6) 1259–1267.

Dexter, F., A. Macario, F. Qian, R. D. Traub. 1999a. Forecasting surgical groups' total hours of elective cases for allocation of block time: Application of time series analysis to operating room management. *Anesthesiology* **91**(5) 1501–1508.

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

Dexter, F., M. Davis, C. B. Egger Halbeis, R. Marjamaa, J. Marty, C. McIntosh, Y. Nakata, K. N. Thenuwara, T. Sawa, M. Vigoda. 2006. Mean operating room times differ by 50% among hospitals in different countries for laparoscopic cholecystectomy and lung lobectomy. *J. Anesthesia* **20**(4) 319–322.

Erdogan, S. A., B. T. Denton. 2011. Surgery planning and scheduling. J. J. Cochran, L. A. Cox Jr., P. Keskinocak, J. P. Kharoufeh, J. C. Smith, eds. *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, New York.

Greene, J. 2004. St. Luke's builds state-of-the-art cardiovascular center. *OR Manager* **20**(9) 1–2.

Greene, J. 2006. OR renovation develops efficient flow. *OR Manager* **22**(8) 1–3.

Guan, Y., S. Ahmed, G. L. Nemhauser. 2009. Cutting planes for multistage stochastic integer programs. *Oper. Res.* **57**(2) 287–298.

Guan, Y., S. Ahmed, G. L. Nemhauser, A. J. Miller. 2006. A branch-and-cut algorithm for the stochastic uncapacitated lot-sizing problem. *Math. Programming* **105**(1) 55–84.

Gupta, D. 2007. Surgical suites' operations management. *Production Oper. Management* **16**(6) 689–700.

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

Healthcare Financial Management Association. 2003. Achieving operating room efficiency through process integration. *Healthcare Financ. Management* **57**(3, Supplement) 1–8.

Huschka, T. R., B. T. Denton, S. Gul, J. W. Fowler. 2007. Bi-criteria evaluation of an outpatient procedure center via simulation. *Proc. 2007 Winter Simulation Conf.*, IEEE Press, Piscataway, NJ, 1510–1518.

Jebali, A., A. B. H. Alouane, P. Ladet. 2006. Operating rooms scheduling. *Internat. J. Production Econom.* **99**(1/2) 52–62.

Jensen, J. L. W. V. 1906. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica* **30**(1) 175–193.

Kiwiel, K. C. 1990. Proximity control in bundle methods for convex nondifferentiable minimization. *Math. Programming* **46**(1–3) 105–122.

Laporte, G., F. V. Louveaux, L. van Hamme. 1994. Exact solution to a location problem with stochastic demands. *Transportation Sci.* **28**(2) 95–103.

Margot, F. 2002. Pruning by isomorphism in branch-and-cut. *Math. Programming Ser. A* **94**(1) 71–90.

Marjamaa, R. A., P. M. Torkki, E. J. Hirvensalo, O. A. Kirvelä. 2009. What is the best workflow for an operating room? A simulation study of five scenarios. *Health Care Management Sci.* **12**(2) 142–146.

Ostrowski, J., J. Linderoth, F. Rossi, S. Smriglio. 2011. Orbital branching. *Math. Programming* **126**(1) 147–178.

Ruszczyński, A. 1986. A regularized decomposition method for minimizing a sum of polyhedral functions. *Math. Programming* **35**(3) 309–333.

Sanchez, S. M., R. K. Wood. 2006. The "BEST" algorithm for solving stochastic mixed integer programs. *Proc. 2006 Winter Simulation Conf. Monterey, CA*, 765–773.

Sandberg, W. S., B. Daily, M. Egan, J. E. Stahl, J. M. Goldman, R. A. Wiklund, D. Rattner. 2005. Deliberate perioperative systems design improves operating room throughput. *Anesthesiology* **103**(2) 406–418.

Sherali, H. D., J. C. Smith. 2001. Improving discrete model representations via symmetry considerations. *Management Sci.* **47**(10) 1396–1407.

Smith, J. C., A. J. Schaefer, J. W. Yen. 2004. A stochastic integer programming approach to solving a synchronous optical network ring design problem. *Networks* **44**(1) 12–26.

Sokal, S. M., D. L. Craft, Y. Chang, W. S. Sandberg, D. L. Berger. 2006. Maximizing operating room and recovery room capacity in an era of constrained resources. *Arch. Surgery* **141**(4) 389–395.

Sokal, S. M., Y. Chang, D. L. Craft, W. S. Sandberg, P. F. Dunn, D. L. Berger. 2007. Surgeon profiling: A key to optimum operating room use. *Arch. Surgery* **142**(4) 365–370.

Strum, D. P., J. H. May, L. G. Vargas. 2000. Modeling the uncertainty of surgical procedure times: Comparison of log-normal and normal models. *Anesthesiology* **92**(4) 1160–1167.

Strum, D. P., J. H. May, A. R. Sampson, L. G. Vargas, W. E. Spangler. 2003. Estimating times of surgeries with two component procedures: Comparison of the lognormal and normal models. *Anesthesiology* **98**(1) 232–240.

Testi, A., E. Tanfani, G. Torre. 2007. A three-phase approach for operating theatre schedules. *Health Care Management Sci.* **10**(2) 163–172.

van Oostrum, J. M., M. Van Houdenhoven, J. L. Hurink, E. W. Hans, G. Wullink, G. Kazemier. 2008. A master surgical scheduling approach for cyclic scheduling in operating room departments. *OR Spectrum* **30**(2) 355–374.

Van Slyke, R. M., R. Wets. 1969. *L*-Shaped linear programs with applications to optimal control and stochastic programming. *SIAM J. Appl. Math.* **17**(4) 638–663.

Velásquez, R., M. T. Melo. 2005. A set packing approach for scheduling elective surgical procedures. H.-D. Haasis, H. Kopfer, J. Schönberger, eds. *Operations Research Proceedings 2005: Selected Papers*. Springer, Berlin, 425–430.

Wang, P. P. 1993. Static and dynamic scheduling of customer arrivals to a single-server system. *Naval Res. Logist.* **40**(3) 345–360.

Weiss, E. N. 1990. Models for determining estimated start times and case orderings in hospital operating rooms. *IIE Trans.* **22**(2) 143–150.

Yee, C. M. 2007. Scanning technology on surgery's cutting edge. *Star Tribune* (September 5), http://www.startribune.com/business/11244781.html.

Zinn, B. 2009. AMC unveils new vascular operating room. *Newsleader* (April 9), http://www.newsleader.com/article/20090409/NEWS01/904090344/-1/NEWS17.