SCHEDULING MULTIPLE OPERATING ROOMS UNDER UNCERTAINTY

by

Sakine Batun

B.S., Middle East Technical University, 2004

M.S., Middle East Technical University, 2006

Submitted to the Graduate Faculty of

the Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2011

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This dissertation was presented

by

Sakine Batun

It was defended on

November 18, 2011

and approved by

Andrew J. Schaefer, Ph.D., Professor, Department of Industrial Engineering

Brian T. Denton, Ph.D., Associate Professor, Department of Industrial and Systems Engineering, North Carolina State University

Oleg A. Prokopyev, Ph.D., Assistant Professor, Department of Industrial Engineering

Juan Pablo Vielma, Ph.D., Assistant Professor, Department of Industrial Engineering

Dissertation Director: Andrew J. Schaefer, Ph.D., Professor, Department of Industrial Engineering

SCHEDULING MULTIPLE OPERATING ROOMS UNDER UNCERTAINTY

Sakine Batun, PhD

University of Pittsburgh, 2011

Operating room (OR) scheduling is an important operational problem for most hospitals. Uncertainty in the surgery delivery process, the existence of multiple resources and competing performance criteria are among the important aspects of OR scheduling problems in practice. Considering these aspects, this dissertation focuses on developing and efficiently solving novel stochastic programming models for multi-OR scheduling problems under uncertainty in surgery durations.

We first consider a stochastic multi-OR scheduling problem with multiple surgeons where the daily scheduling decisions are made before the resolution of uncertainty. We formulate the problem as a two-stage stochastic mixed-integer program that minimizes the sum of the fixed cost of opening ORs and the expected overtime and surgeon idling cost. Decisions in our model include the number of ORs to open, the allocation of surgeries to ORs, the sequence of surgeries in each OR, and the start times for surgeons. Realistic-sized instances of our model are difficult or impossible to solve with standard stochastic programming techniques. Therefore, we exploit several structural properties of our model and describe a novel set of widely applicable valid inequalities to achieve computational advantages. We use our results to quantify the value of capturing uncertainty and the benefit of pooling ORs, and to demonstrate the impact of parallel surgery processing on surgery schedules.

We then consider a stochastic multi-OR scheduling problem where the initial schedule is revised at a prespecified rescheduling point during the surgical day. We formulate the problem as a three-stage stochastic mixed-integer program that minimizes the sum of the fixed cost of opening ORs and the expected overtime cost. The number of ORs to open and the allocation of surgeries to ORs are the first-, and the revisions on the allocation of surgeries to ORs are the second-stage decisions in our model. For our computational study, we consider a special case, which is a two-stage stochastic mixed-integer program, where rescheduling decisions are made under perfect information. We use stage-wise and scenario-wise decomposition methods to solve our model. By using our results, we estimate the value of rescheduling, and illustrate the impact of different surgery sequencing rules on this value.

Keywords: operating room scheduling, operating room rescheduling, operating room pooling, parallel surgery processing, multiple operating rooms, two-stage stochastic mixed-integer programming, multi-stage stochastic mixed-integer programming.

TABLE OF CONTENTS

1.0 INTRODUCTION 1
1.1 Background Information on Operating Room Scheduling 2
1.2 Contributions of the Dissertation 7
1.3 Organization of the Dissertation 8
1.4 Acknowledgment 8
2.0 LITERATURE REVIEW 9
2.1 Operating Room Scheduling 9
2.2 Stochastic Programming 12
2.2.1 Two-Stage Stochastic Programs 12
2.2.2 Multi-Stage Stochastic Programs 15
2.2.3 Value of Perfect Information and the Stochastic Solution 18
3.0 OPERATING ROOM POOLING AND PARALLEL SURGERY PROCESSING UNDER UNCERTAINTY 20
3.1 Motivation and Contributions 20
3.2 Problem Definition and Mathematical Formulation 22
3.3 Solution Methods 26
3.3.1 Anti-Symmetry Constraints 27
3.3.2 Feasibility of the Second-Stage Problem 27
3.3.3 Structure of Scenario Subproblems 28
3.3.4 Extended Master Problem Formulation 30
3.4 Computational Results 33
3.4.1 Generation of Problem Instances 33

3.4.2 Computational Performance of the Proposed Algorithms 35
3.4.3 Value of the Stochastic Solution 43
3.4.4 Value of OR Pooling 44
3.4.5 Impact of Parallel Surgery Processing 48
3.5 Conclusions 48
4.0 OPERATING ROOM RESCHEDULING UNDER UNCERTAINTY . 51
4.1 Motivation and Contributions 51
4.2 Problem Definition and Mathematical Formulation 53
4.3 Rescheduling Under Perfect Information 60
4.4 Solution Methods 65
4.4.1 The Integer L-Shaped Algorithm for 2S-ORRP 65
4.4.2 A Progressive Hedging-Based Heuristic for 2S-ORRP 66
4.5 Computational Results 71
4.5.1 Generation of Problem Instances 71
4.5.2 Computational Performance of the Proposed Algorithms 72
4.5.3 Value of Rescheduling 75
4.5.4 Impact of the Overtime Cost Level 78
4.5.5 Impact of the Surgery Sequence 78
4.6 Conclusions 80
5.0 CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS 82
5.1 Summary and Conclusions 82
5.2 Future Research Directions 84
BIBLIOGRAPHY 86

LIST OF TABLES

3.1 Size-based classification of problem instances for 322 surgical days. 36

3.2 Computational performance of the L-shaped and L-shaped based branch-and-cut algorithms. 40

3.3 Computational performance of the proposed algorithms for 20 instances. . . . 41

3.4 Solution times (in CPU seconds) of mean value and stochastic problems. . . . 42

3.5 Percentage gap values for the unsolved instances. 42

3.6 Percentage value of the stochastic solution for each problem set. 43

3.7 Optimal solution statistics for the stochastic and mean value problems for each problem set. 45

3.8 Percentage improvement brought by OR pooling. 46

3.9 Optimal solution statistics for Setting 1 and Setting 2 for each problem set. . 47

4.1 Important characteristics of the problems considered in this study. 64

4.2 Size-based classification of 2S-ORRP instances for 322 surgical days. 72

4.3 Solution times (in CPU seconds) and the percentage optimality gap values for the integer L-shaped algorithm. 73

4.4 Solution times (in CPU seconds) and the objective values of the best solutions returned by the PH-based heuristic under different parameter settings for a problem instance whose optimal objective function value is 16993.34. 74

4.5 Solution times (in CPU seconds) and the percentage gap values for the PH-based heuristic. 75

4.6 Percentage value of modeling uncertainty and rescheduling decisions. 77

4.7 Percentage value of modeling uncertainty and rescheduling decisions at different overtime cost levels for the problem instances in Set 2. 79
4.8 Percentage value of rescheduling decisions under different sequencing rules for the problem instances in Set 2. 80

LIST OF FIGURES

1.1 A feasible surgery schedule illustrating three surgeons sharing two ORs. . . . 4

1.2 An illustration of overtime and surgeon idle time under different scenarios. . . 6

1.3 Parallel surgery processing of two surgeries across two ORs. 6

2.1 A tree of six scenarios over three periods. 16

3.1 Flow chart for our solution method that uses the L-shaped algorithm. 37

3.2 Flow chart for our solution method that uses the L-shaped based branch-and-cut algorithm. 38

3.3 Optimal number of ORs to open at different levels of OR turnover time and parallelizable portion of surgeries. 49

4.1 Scenario tree for the OR rescheduling problem where the initial schedule is revised at T_R. 54

4.2 Important time points during a surgical day. 56

4.3 Rescheduling under perfect information. 61

1.0 INTRODUCTION

Health care expenditures in the United States exceeded $2.4 trillion in 2009, accounting for around 17% of the Gross Domestic Product [14]. Hospital expenditures represent approximately a third of this total amount [14], and surgery generates more than 40% of a hospital's total expenses and revenues [28, 44]. A recent joint study by the National Academy of Engineering and the Institute of Medicine [61] highlights the importance of health care and engineering partnership, and indicates scheduling in health care delivery systems as one of the areas with significant research opportunities.

Besides being an important operational problem, designing surgery schedules is also a challenging problem due to several factors including its combinatorial nature, uncertainty involved in the surgery delivery process, and the existence of many different and expensive resources and multiple performance criteria.

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. However, both the stochastic nature of the surgery delivery process and the existence of multiple ORs in the surgical environment need to be considered in order to develop approaches for the practical planning and scheduling problems in health care delivery systems. In this dissertation, we develop novel stochastic programming models for the optimal design of surgery schedules across multiple ORs of a surgical ward under uncertainty, and we show how our models can be used for long-term planning. Since the OR is typically the bottleneck in the overall process, we make it the central focus of our study. We consider identical and therefore interchangeable ORs (which is common in practice in a surgical ward). However, our models can easily be extended for surgical environments with nonidentical ORs by introducing some eligibility constraints and modifying or completely removing the symmetry-breaking constraints accordingly.

1.1 BACKGROUND INFORMATION ON OPERATING ROOM SCHEDULING

The overall surgery delivery process, which is also known as *peri-operative services*, is comprised of a variety of activities that are performed through *pre-operative*, *intra-operative* and *post-operative* stages. The pre-operative stage begins with the surgery decision and continues until the patient's arrival to the OR. It typically includes all preparations prior to the surgery such as physical examinations, medical tests, and administrative work. The intra-operative stage includes all activities performed in the OR and ends with the patient's transfer to the recovery area. The last phase, which is the post-operative stage, includes recovery and follow-up periods.

Surgeries may be performed on an *inpatient* or *outpatient* basis. In an inpatient setting, the patient is admitted to the hospital either on or prior to the day of the surgery, and stays in the hospital until the completion of the recovery period. If the surgery occurs on an outpatient basis, admission, surgery and discharge take place on the same day. Surgeries may be either *elective* or *non-elective* (*urgent* or *emergent*). Elective surgeries are planned in advance, whereas non-elective surgeries typically arise unexpectedly during the day and they need to be added to the existing schedule. This dissertation is mainly focused on the scheduling problems in the intra-operative stage of the delivery of inpatient elective surgeries.

Whether a surgery is elective or non-elective, or operated in an inpatient or outpatient setting, many features of the OR environment are the same. The entire area that includes the ORs, intake and recovery areas, and a storage area for the equipment/material is called the *surgical suite* or the *OR suite*. A surgery delivery system is composed of many different resources including a surgical suite, equipment/material resources, and human resources such as surgeons, nurses, and anesthesiologists.

The daily fixed cost of opening an OR is significant due to the cost of OR staff, and staffing of supporting upstream (intake) and downstream (recovery) areas. Typically, ORs have planned session lengths of 8-9 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/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 1.1 illustrates important aspects of OR scheduling with a simple deterministic example. There are eleven surgeries to be performed by three surgeons, and the size of the surgery blocks denote the lengths of the surgeries. For example, Surgeon 1 has five surgeries, of which the third is the longest. There are two identical ORs, and the size of an OR block represents the daily session length (e.g. 8 hours). A feasible schedule is illustrated where surgeries of Surgeon 2 are scheduled in OR 1, and surgeries of Surgeon 1 and Surgeon 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 [22, 27, 78, 79]. When surgeries are scheduled based on expected values

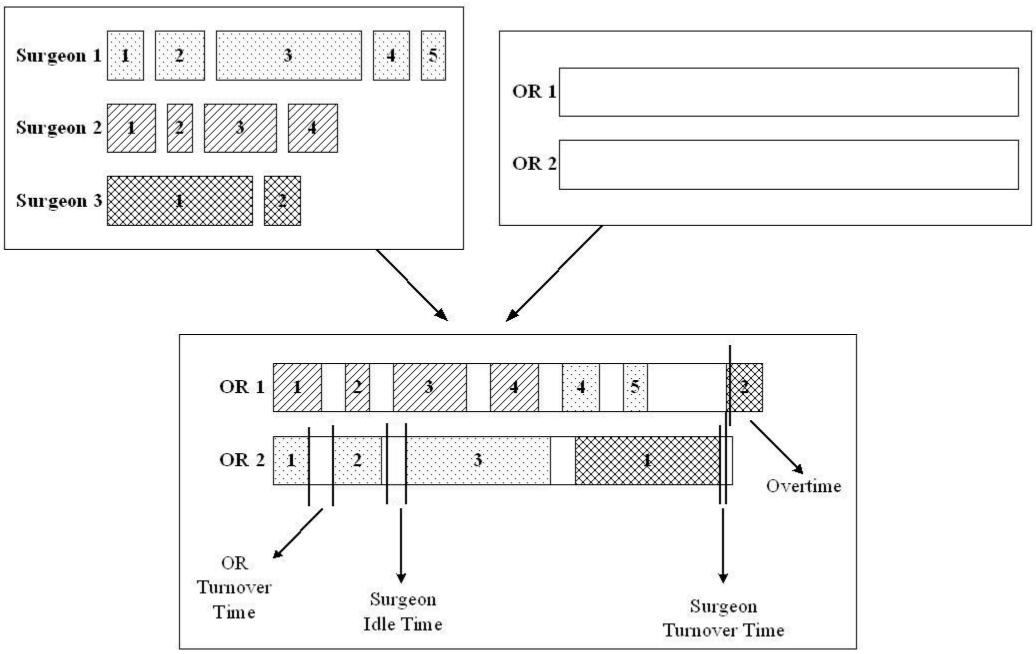

Figure 1.1: A feasible surgery schedule illustrating three surgeons sharing two ORs.

of surgery durations (as is often done in practice), high expected overtime and surgeon idle time may occur [17], which is illustrated in Figure 1.2. 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 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 *pre-incision*, *incision* and *post-incision*. 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 pre-incision phase includes positioning the patient on the OR bed and initiating anesthesia, and post-incision phase includes closing the incision. Since 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, with the attending 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/she is in the surgical ward but not performing the critical portion of a surgery. Figure 1.3 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/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 pre-incision phase of surgery 2 is completed, then the incision phase of surgery 2 is delayed until the surgeon becomes available.

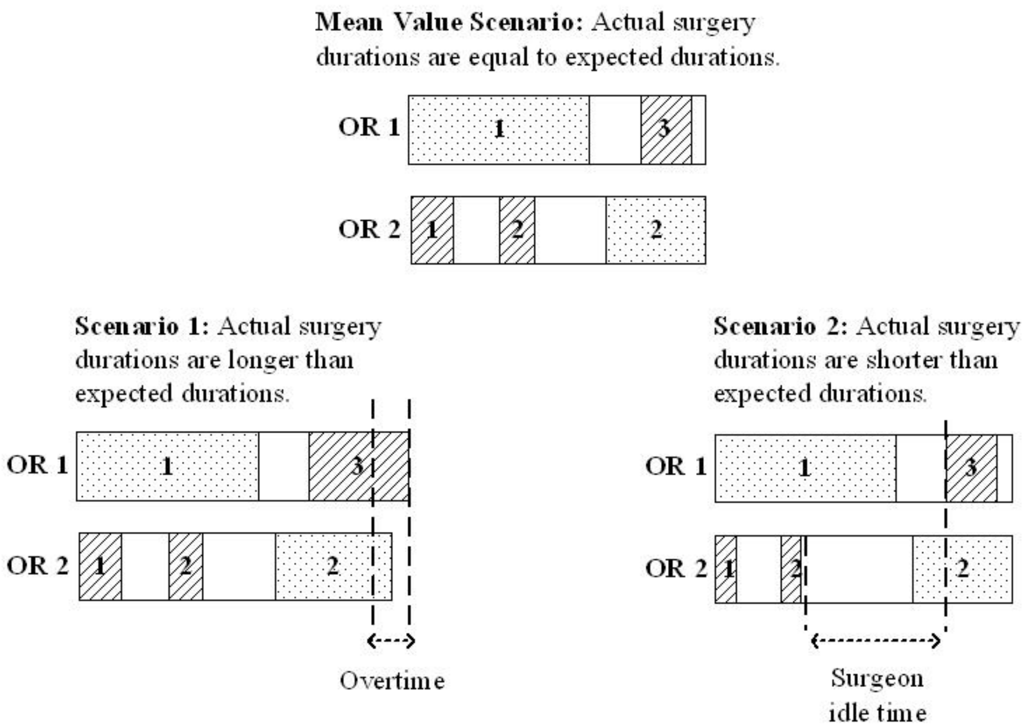

Figure 1.2: An illustration of overtime and surgeon idle time under different scenarios.

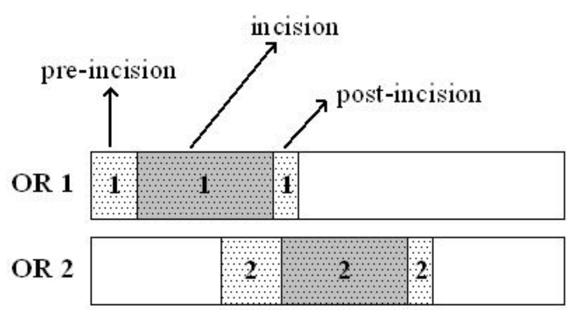

Figure 1.3: Parallel surgery processing of two surgeries across two ORs.

1.2 CONTRIBUTIONS OF THE DISSERTATION

We first consider a multi-OR scheduling problem with multiple surgeons where the surgery durations are uncertain. We formulate a two-stage stochastic mixed-integer program (SMIP) for the problem. 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 different resource usage schemes such as parallel surgery processing and operating room pooling. 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 add symmetry elimination constraints to the first stage. 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 [48]. We perform a series of computational experiments to test our proposed methods, to illustrate the impact of parallel surgery processing, and to quantify the potential benefit of pooling ORs as a shared resource among surgeons.

We then explore a stochastic multi-OR scheduling problem where surgery-to-OR allocation decisions are allowed to be revised during the day. We formulate the problem as a three-stage SMIP, where the initial scheduling decisions and the rescheduling decisions are made in the first and second stages, respectively. We consider a special case of our model, which is a two-stage SMIP, where the surgeries are rescheduled under perfect information in the second stage. We solve this problem with stage-wise and scenario-wise decomposition methods. We estimate the potential benefit of rescheduling and the portion of this benefit which is attributable to the adaptive/anticipative decisions. We also investigate the impact of using different surgery sequencing rules within ORs on the value of rescheduling.

1.3 ORGANIZATION OF THE DISSERTATION

The remainder of this dissertation is organized as follows. In Chapter 2, we review the relevant literature on OR scheduling and stochastic programming, which are the application area and the methodological domain considered in this study. In Chapters 3 and 4, we describe our models, solution methodologies, and present our results and insights for stochastic multi-OR scheduling and rescheduling problems, respectively. We discuss conclusions and highlight future research directions in Chapter 5.

1.4 ACKNOWLEDGMENT

Much of the content in Chapters 1, 2, and 3 originally appeared in Batun et al. [4], and is reproduced with kind permission from The Institute for Operations Research and the Management Sciences: S. Batun, B. T. Denton, T. R. Huschka, and A. J. Schaefer, "Operating Room Pooling and Parallel Surgery Processing Under Uncertainty," *INFORMS Journal on Computing*, 23 (2), pp. 220-237, 2011.

2.0 LITERATURE REVIEW

2.1 OPERATING ROOM SCHEDULING

Deterministic and stochastic optimization models [6, 17, 19, 20, 30, 32, 33, 47, 54, 64, 65, 81, 85, 89], queueing models [42, 84, 90, 93, 95], simulation models [18, 24, 31, 39, 46, 59] and heuristic approaches [7, 19, 24, 39, 54] have all been widely used to investigate OR scheduling. In our review, we focus on those studies which are directly related to multi-OR scheduling or which consider stochastic programming models for OR scheduling. More extensive reviews are found in [11, 12, 29, 38, 40, 41, 60].

Velásquez and Melo [85] study a 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. [47] 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. Fei et al. [33] consider a similar hierarchical approach to solve a deterministic multi-OR scheduling problem over a weekly planning horizon where the objective is to minimize the OR idle time and overtime costs. They first solve a set-partitioning problem by using a column-generation-based heuristic to assign a date for each surgery, then solve the daily scheduling problem by using a hybrid genetic algorithm. Liu et al. [54] propose a heuristic algorithm to solve large instances of the problem defined in [33].

Testi et al. [81] 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 Adan et al. [1], Beliën and Demeulemeester [7], van Oostrum et al. [82].

Weiss [89] 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 [87] considers a single server (equivalently OR) appointment system where the processing times are assumed to be exponentially distributed. 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 [17] 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 in order 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 the distribution of job duration, 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. Begen and Queyranne [6] consider the single server appointment scheduling problem under the assumption that service durations are independent and discrete. They show that the objective function, which is the expected total underage and overage costs, is L-convex under reasonable conditions on cost coefficients. Based on this property, they also provide a polynomial time algorithm for the problem.

Denton et al. [19] extend the single server appointment scheduling problem [17] to investigate the impact of 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 effected both by scheduled start times and sequencing decisions.

Erdogan and Denton [30] consider the dynamic appointment scheduling problem where service durations and the number of customers to be scheduled on a particular day are uncertain. They formulate the problem as a multi-stage stochastic linear program where stages are defined by customer appointment requests, and the objective is to minimize the expected cost of overtime and customer waiting time. At each stage, the requested appointment is scheduled by determining the length of the time slot to be allocated. They analyze the structural properties of the model and utilize decomposition-based algorithms to solve the problem instances generated based on real data.

Denton et al. [20] 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.

Despite the vast amount of literature on OR scheduling, there is still a lack of studies on stochastic multi-OR scheduling problems. However, stochasticity in the surgery delivery process and the existence of multiple ORs in surgical suites are important aspects that need to be captured while formulating and solving the practical problems seen in surgery delivery systems.

2.2 STOCHASTIC PROGRAMMING

Stochastic programming is a branch of mathematical programming that provides a framework for modeling and solving optimization problems with random parameters. The evolution of information over time, which plays a critical role in decision making under uncertainty, is explicitly considered through *decision stages* in stochastic programs. In this section, we briefly introduce the types of stochastic programs and the related solution methods that are most relevant to this dissertation. For more comprehensive information on stochastic programming, we refer to Birge and Louveaux [10] and Kall and Wallace [49].

2.2.1 Two-Stage Stochastic Programs

A two-stage stochastic program [5, 16], which is the most widely studied type of stochastic program, is composed of first and second stages that take place before and after the resolution of uncertainty, respectively. The *first-stage decisions* ($x \in \mathbf{X}$) are made a priori, before the availability of complete information on the random parameters. Given the first-stage decisions, the *second-stage decisions* ($y(\omega) \in \mathbf{Y}$), also called *recourse decisions*, are made after the random scenario (indexed by $\omega \in \Omega$) is realized and the vector of random parameters ($\xi(\omega) \in \Xi$) become known. A two-stage stochastic program, which minimizes the first-stage cost plus the expected second-stage cost, can be formulated in the *deterministic equivalent form* as follows:

$$\min \quad z = c^T x + \mathcal{Q}(x) \tag{2.1a}$$

s.t.

$$Ax = b, \tag{2.1b}$$

$$x \in \mathbf{X} \tag{2.1c}$$

where

$$\mathcal{Q}(x) = E_\xi\left[Q(x, \xi(\omega))\right] \tag{2.2}$$

is the *expected recourse function*, and for every scenario ω,

$$Q(x, \xi(\omega)) = \min\{q(\omega)y(\omega) | W(\omega)y(\omega) = h(\omega) - T(\omega)x, y(\omega) \in \mathbf{Y}\} \tag{2.3}$$

defines the *recourse function*.

In the above formulation, $c^T \in R^{n_1}$, $A \in R^{n_1 \times m_1}$, $b \in R^{m_1}$, $T(\omega) \in R^{n_1 \times m_2}$, $W(\omega) \in R^{n_2 \times m_2}$ and $h(\omega) \in R^{m_2}$ are real-valued matrices and vectors. The matrices $T(\omega) \in R^{n_1 \times m_2}$ and $W(\omega) \in R^{n_2 \times m_2}$ define the second-stage constraints under scenario ω; they are called the *technology* and *recourse matrices*, respectively. A stochastic program is said to have *fixed recourse*, if the recourse matrix is not scenario-dependent (i.e., $W(\omega) = W \quad \forall \omega$). The *(relatively) complete recourse* property of a two-stage stochastic program implies the feasibility of the second-stage problem for any given (feasible) first-stage solution.

If sets $\mathbf{X}$ and $\mathbf{Y}$ impose only nonnegativity restrictions on x and $y(\omega)$, then (2.1)-(2.3) is a two-stage stochastic linear program (SLP). Otherwise, if a subset of the first- and/or second-stage variables are integers, then the resulting formulation is a two-stage stochastic mixed-integer program (SMIP).

Letting π_ω be the realization probability of scenario ω, (2.1)-(2.3) can be reformulated in the *extensive form* as follows:

$$\min \quad z = c^T x + \sum_{\omega \in \Omega} \pi_\omega q(\omega) y(\omega) \tag{2.4a}$$

s.t.

$$Ax = b, \tag{2.4b}$$

$$T(\omega)x + W(\omega)y(\omega) = h(\omega) \qquad \forall \omega, \tag{2.4c}$$

$$x \in \mathbf{X}, \tag{2.4d}$$

$$y(\omega) \in \mathbf{Y} \qquad \forall \omega. \tag{2.4e}$$

Any solution approach for linear programs (LPs) or integer programs (IPs) can be used to solve the extensive form of a stochastic program when there are few possible scenario realizations. However, the size of a stochastic program grows, and hence its solvability with standard solution methods decreases, with the increasing number of scenarios. Therefore, when there exist a large number of scenarios (which is common in many practical problems), solution methods must exploit the structural properties of stochastic programs.

Given a first-stage solution, the scenario-dependent second-stage problems can be solved independently. Owing to this so-called *block separability property* of the recourse, decomposition methods are very efficient for solving stochastic programs [45, 67, 83]. One of the most commonly used stage-wise decomposition methods for two-stage SLPs is the *L-shaped algorithm* [83], which is an extension of Benders decomposition [8] for stochastic programs. The L-shaped algorithm is an outer linearization approach in which a master problem (composed of the first-stage variables and constraints) is solved iteratively until the optimal solution is found. At each iteration, given the first-stage solution obtained by solving the master problem, *second-stage scenario subproblems* are solved and the optimal values of dual variables are used to generate *optimality* and *feasibility cuts*. These cuts are added to the master problem to approximate the expected recourse function (using optimality cuts) and guarantee the feasibility of the second-stage subproblems (using feasibility cuts) throughout the iterations. The optimality cuts include a surrogate variable, θ, that represents the approximate expected recourse function, defining a progressively better lower bound on the expected recourse function at each iteration.

Although the L-shaped algorithm was originally proposed to solve two-stage SLPs, it can also be employed to solve a two-stage SMIP if the integrality restrictions are only on the first-stage variables. Two-stage SMIPs with integrality restrictions on the second-stage variables can be solved using the *integer L-shaped algorithm* [52] if the problem has relatively complete recourse and the first-stage is pure binary. The integer L-shaped algorithm solves the master problem within a branch-and-cut framework where optimality cuts are added as necessary at each integer feasible node. The optimality cut at a considered node is generated using the corresponding value of the expected second-stage cost, and a pre-calculated lower bound on the expected recourse function. Therefore, unlike in the standard L-shaped algorithm, the construction of the optimality cuts in the integer L-shaped algorithm does not require the linearity of the second-stage subproblems. Due to the practical importance of two-stage stochastic programs with integer variables, there are several other methods developed to overcome the computational challenge of solving these problems [2, 13, 43, 51, 70, 71, 72, 73].

2.2.2 Multi-Stage Stochastic Programs

Many problems in practice involves sequential decision making over time. Multi-stage stochastic programs can be used to formulate such problems where uncertainty is resolved in multiple stages. The decisions at any stage are made by considering the information revealed up until that stage, and the uncertainty inherent in the successive stages. Evolution of information in a multi-stage stochastic program can be shown on a tree structure where each scenario in every stage is represented by a node. The scenario tree of a three-stage stochastic program with three second-stage scenarios, and six third-stages scenarios is given in Figure 2.1.

A multi-stage stochastic program can be formulated as follows:

$$\min \quad z = c_1^T(\omega_1)x_1(\omega_1) + \mathcal{Q}_1(x_1(\omega_1)) \tag{2.5a}$$

s.t.

$$W_1(\omega_1)x_1(\omega_1) = h_1(\omega_1), \tag{2.5b}$$

$$x_1(\omega_1) \in \mathbf{X}_1 \tag{2.5c}$$

where

$$\mathcal{Q}_t(x_t(\omega_t)) = E_{\xi_{t+1}|\xi_t}\left[Q_t(x_t(\omega_t), \xi_{t+1}(\omega_{t+1}))\right] \tag{2.6}$$

and

$$\begin{aligned} Q_t(x_t(\omega_t), \xi_{t+1}(\omega_{t+1})) = \min\{ & c_{t+1}^T(\omega_{t+1})x_{t+1}(\omega_{t+1}) + \mathcal{Q}_{t+1}(x_{t+1}(\omega_{t+1}))| \\ & W_{t+1}(\omega_{t+1})x_{t+1}(\omega_{t+1}) = h_{t+1}(\omega_{t+1}) - T_{t+1}(\omega_{t+1})x_t(\omega_t), \\ & x_{t+1}(\omega_{t+1}) \in \mathbf{X}_{t+1}\} \end{aligned} \tag{2.7a}$$

for $t = 1, 2, ..., T-1$ where t is the stage index and T is the number of stages.

In the above formulation, $c_t^T(\omega_t), T_t(\omega_t), W_t(\omega_t)$, and $h_t(\omega_t)$ are real-valued matrices and vectors of conformable size. It is assumed that ω_1, and hence $c_1^T(\omega_1), T_1(\omega_1), W_1(\omega_1)$, are

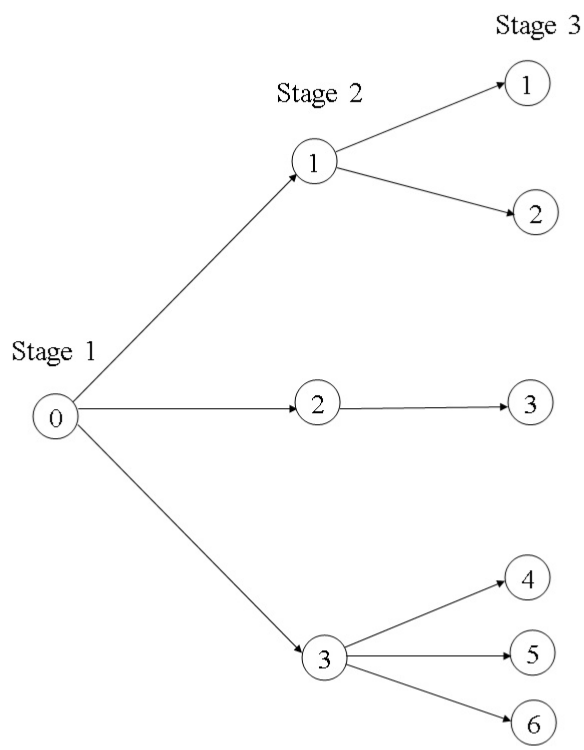

Figure 2.1: A tree of six scenarios over three periods.

known whereas the parameters in the future stages are random variables. (2.5)-(2.7) is a multi-stage SLP if all variables are continuous, a multi-stage SMIP otherwise.

The *nested L-shaped algorithm* [9, 55], a generalization of the L-shaped algorithm to multiple stages, is one of the commonly used solution methods for multi-stage SLPs. Throughout the iterations of the nested L-shaped algorithm, a subproblem is considered for each scenario in every stage. The primal solution of a subproblem is passed to the descendant nodes, and the dual solutions of the subproblems at the descendant nodes are passed back to the ancestor node to be used in the cut generation. Cuts are added to approximate the expected cost over the remaining stages ($\mathcal{Q}_t(x_t(\omega_t))$) and to ensure feasibility of the solution ($x_t(\omega_t)$) in the descendant scenarios. The algorithm iterates between the stages and carries information from one scenario node to another until the optimal solution is found. When a subproblem in stage t is solved, the algorithm can either move back to stage $t-1$ and solve the subproblem at the ancestor node after adding the cuts, or move on to stage $t+1$ and solve the subproblems at the descendant nodes. There are different strategies used to determine the order in which subproblems are solved in the nested L-shaped algorithm. One of the widely used strategies is the *fast-forward-fast-back* approach [91], according to which the algorithm explores all subproblems in a stage (starting from the first stage) and moves on to the next stage by passing the primal solution until it detects an infeasible subproblem or reaches the last stage, and then moves backwards by adding cuts to the ancestor nodes until it reaches the first stage. This cycle repeats until no new cuts are generated, which indicates that the optimal solution is found.

The *progressive hedging algorithm* (PHA) [66] is another widely used solution method for multi-stage SLPs. It provides an iterative framework where an augmented Lagrangian relaxation is applied to the explicitly modeled *nonanticipativity constraints*. This framework facilitates the decomposition of a multi-stage SLP into *scenario subproblems* (which are deterministic LPs) that can be solved independently. An aggregated policy is obtained at each iteration by using the scenario subproblem solutions. A penalty term associated with the deviation from the aggregated policy is included in the objective function of the scenario subproblems to progressively enforce the nonanticipativity constraints. The algorithm iterates until the optimal solution is found.

Although the convergence of the PHA to the optimal solution is guaranteed only for problems with continuous variables, progressive hedging-based (PH-based) heuristic methods are observed to be very efficient in obtaining good solutions to SMIPs [15, 88]. As can be seen in more recent work, scenario-wise decomposition approach is also used to develop branch-and-price methods for multi-stage SMIPs [56, 63].

2.2.3 Value of Perfect Information and the Stochastic Solution

For one particular scenario ω, the deterministic version of the two-stage stochastic program described by (2.1)-(2.3) can be written as:

$$\min \quad z(x, \xi(\omega)) = c^T x + q(\omega) y(\omega) \tag{2.8a}$$

$$\text{s.t.}$$

$$Ax = b, \tag{2.8b}$$

$$T(\omega)x + W(\omega)y(\omega) = h(\omega), \tag{2.8c}$$

$$x \in \mathbf{X}, \tag{2.8d}$$

$$y(\omega) \in \mathbf{Y}. \tag{2.8e}$$

If perfect information were available (i.e., if the random scenario was realized before the decision maker sets x), the optimal solution could be found by solving (2.8) for the already known scenario. The expected value of the solution under perfect information, known as the *wait-and-see* solution, is:

$$z_{WS} = E_\xi \left[\min z(x, \xi(\omega)) \right]. \tag{2.9}$$

The *expected value* or *mean value* problem is the deterministic version of the stochastic program where the random parameters ($\xi(\omega)$) are replaced with their expected values ($E[\xi(\omega)]$). The optimal solution of the expected value problem is called the *expected value solution* (x_{EV}). The expected value of using x_{EV} is:

$$z_{EEV} = E_\xi \left[\min z(x_{EV}, \xi(\omega)) \right]. \tag{2.10}$$

Letting z_{SP} denote the objective function value of the optimal solution of (2.1)-(2.3), we have the following relation between z_{WS}, z_{EEV} and z_{SP} [57]:

$$z_{WS} \leq z_{SP} \leq z_{EEV}. \tag{2.11}$$

There are two important measures used to estimate the value of modeling uncertainty and incorporating it into mathematical programs.

- The *expected value of perfect information* (EVPI) is the expected benefit that could be gained by using perfect information (instead of using the solution of a stochastic program), and hence is the maximum amount a decision maker would be willing to pay in return for perfect information. It is formally defined as:

$$EVPI = z_{SP} - z_{WS}. \tag{2.12}$$

- The *value of the stochastic solution* (VSS) is the expected benefit of formulating and solving the problem as a stochastic program rather than using the solution of the corresponding expected value model. It is formally defined as:

$$VSS = z_{EVV} - z_{SP}. \tag{2.13}$$

3.0 OPERATING ROOM POOLING AND PARALLEL SURGERY PROCESSING UNDER UNCERTAINTY

3.1 MOTIVATION AND CONTRIBUTIONS

The majority of studies in the OR scheduling literature have focused on either single-OR models or deterministic multi-OR models. Only a few articles consider stochastic-programming-based approaches to capture the stochastic behavior of surgery durations [17, 19, 20].

In this chapter, we explore a stochastic multi-OR scheduling problem with multiple surgeons and identical ORs. By considering surgeons (as well as ORs) and introducing additional decisions to be made, we extend the stochastic surgery scheduling problem studied by Denton et al. [20]. We formulate and solve the problem as a two-stage SMIP to minimize total expected operating cost given that scheduling decisions are made before the resolution of uncertainty in surgery durations. To increase the problem solvability, particularly for the large-sized instances, we exploit a number of structural properties of our model, and propose a novel set of valid inequalities. Our extensive computational study based on data provided by a large health care provider reveals both the computational strength of our proposed methods and some important managerial insights.

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 [23, 24]. 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 [25, 33, 54]. 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.

Earlier studies that consider stochastic-programming based approaches for the OR scheduling problem view surgery as a single activity; however, in practice a surgery is comprised of several activities (pre-incision, incision, and post-incision phases). Depending on surgery-to-OR assignment decisions, some of these activities can be carried out simultaneously (in parallel) due to the availability of multiple ORs, and assistance of other surgeons in addition to the attending staff surgeon (e.g. surgery fellows). Parallelizing surgeries may improve the efficiency of resource usage significantly [59, 69, 76, 77]. To our knowledge, parallel surgery processing has not yet been incorporated into any optimization models. Our model explicitly considers the parallelizable nature of surgery, which is an important aspect that must be taken into account to accurately estimate the benefits of OR pooling.

Our work presented in this chapter differs from the existing literature in a number of ways, and its main contributions can be summarized 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 pre-incision, incision, and post-incision phases.
- As 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.

The remainder of this chapter is organized as follows. In Sections 3.2 and 3.3, we provide the formulation of our model, discuss its structural properties, and present our solution methods. In Section 3.4, 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 Section 3.5.

3.2 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 [16]. 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 pre-incision, incision and post-incision 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.

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.

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.

$pre_i(\omega)$: pre-incision duration of surgery i under scenario ω.

$p_i(\omega)$: incision duration of surgery i under scenario ω.

$post_i(\omega)$: post-incision duration of surgery i under scenario ω.

In our notation, $\omega \in \Omega$ represents the random outcome of the realized scenario. Given n surgeries, we obtain a random vector $\xi(\omega) = \{pre_1(\omega), ..., pre_n(\omega), p_1(\omega), ..., p_n(\omega), post_1(\omega), ..., post_n(\omega)\}$. We denote the finite support of $\xi(\omega)$ by Ξ where $\Xi \in \mathbb{R}_+^{3n}$.

First-Stage 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 zero if j precedes i in one of the surgeons' surgery listing.

t_k: start time for surgeon k.

Second-Stage Decision Variables

$C_{ir}(\omega)$: 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/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:

$$\min \sum_{r=1}^{n_R} c^f x_r + \mathcal{Q}(x, y, z, t) \tag{3.1a}$$

s.t.

$$y_{ir} \leq x_r \qquad \forall i, r, \tag{3.1b}$$

$$\sum_{r=1}^{n_R} y_{ir} = 1 \qquad \forall i, \tag{3.1c}$$

$$z_{ijr} + z_{jir} \leq y_{ir} \qquad \forall i, j > i, r, \tag{3.1d}$$

$$z_{ijr} + z_{jir} \leq y_{jr} \qquad \forall i, j > i, r, \tag{3.1e}$$

$$z_{ijr} + z_{jir} \geq y_{ir} + y_{jr} - 1 \qquad \forall i, j > i, r, \tag{3.1f}$$

$$t_k \leq L \qquad \forall k, \tag{3.1g}$$

$$x_r, y_{ir}, z_{ijr} \in \{0, 1\} \qquad \forall i, j \neq i, r, \tag{3.1h}$$

$$t_k \geq 0 \qquad \forall k, \tag{3.1i}$$

where

$$\mathcal{Q}(x, y, z, t) = E_\xi \left[Q(x, y, z, t, \xi(\omega)) \right] \tag{3.2}$$

is the expected recourse function and

$$Q(x, y, z, t, \xi(\omega)) = \min \sum_{r=1}^{n_R} c^o O_r(\omega) + \sum_{(i,j):\sum_{k=1}^{n_S} b_{ijk}=1} c^S I_{ij}(\omega) + \sum_{k=1}^{n_S} c^S I_k(\omega) \tag{3.3a}$$

s.t.

$$C_{ir}(\omega) \leq M y_{ir} \qquad \forall i, r, \tag{3.3b}$$

$$C_{jr}(\omega) \geq C_{ir}(\omega) + s^R + pre_j(\omega) + p_j(\omega) + post_j(\omega) - M(1 - z_{ijr}) \qquad \forall i, j \neq i, r, \tag{3.3c}$$

$$\sum_{r=1}^{n_R} C_{i_k r}(\omega) = t_k + I_k(\omega) + pre_{i_k}(\omega) + p_{i_k}(\omega) + post_{i_k}(\omega) \qquad \forall k, \tag{3.3d}$$

$$\sum_{r=1}^{n_R} C_{ir}(\omega) \geq t_k + pre_i(\omega) + p_i(\omega) + post_i(\omega) \qquad \forall (i, k) : \sum_{j=1}^{n} b_{jik} = 1, \tag{3.3e}$$

$$\sum_{r=1}^{n_R} C_{jr}(\omega) = \sum_{r=1}^{n_R} C_{ir}(\omega) - post_i(\omega) + s^S + p_j(\omega) + post_j(\omega) + I_{ij}(\omega)$$
$$\forall (i, j) : \sum_{k=1}^{n_S} b_{ijk} = 1, \tag{3.3f}$$

$$O_r(\omega) \geq C_{ir}(\omega) - L \qquad \forall i, r, \tag{3.3g}$$

$$C_{ir}(\omega), I_k(\omega), I_{ij}(\omega), O_r(\omega) \geq 0 \qquad \forall i, j, r, k. \tag{3.3h}$$

The objective function (3.1a) 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 here is a multi-criteria problem, and each piece of the total operating cost defined by (3.1a) corresponds to a different performance measure.

Constraints (3.1b) and (3.1c) 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 (3.1d)-(3.1f). Constraint (3.1g) 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 (3.1h) and (3.1i) define binary and nonnegativity restrictions for the first-stage decision variables.

The second-stage problem for a given x, y, z, t and $\xi(\omega)$ is formulated explicitly by (3.3). The completion time of a surgery in an OR is 0 unless it is assigned to that OR, which is enforced by constraint (3.3b). Constraint (3.3c) defines the completion time of surgeries in ORs considering their precedence relation, processing times and OR turnover time. The M parameter used in constraints (3.3b) and (3.3c) is an upper bound on the surgery completion times. Constraints (3.3d) and (3.3e) ensure that surgeries of a surgeon cannot be started before his/her arrival to the surgical suite. Constraint (3.3d) determines the idle time of the considered surgeon before his/her first surgery. Since the pre-incision 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 pre-incision parts of the subsequent surgeries where the surgeon is considered to be idle unless he/she is performing the critical part of a surgery. Constraint (3.3f) provides the relation between surgery completion times, surgeon idle times and the sequence of surgeries in surgeons' surgery listing. Constraint (3.3g) defines the overtime used in each OR. Constraints (3.3h) define nonnegativity restrictions for the second-stage decision variables.

Notice that we assume 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).

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.

3.3 SOLUTION METHODS

The mathematical model we present is a two-stage SMIP with binary and continuous first-stage decision variables, and continuous second-stage variables. We solve this SMIP by using the L-shaped algorithm [83], which is briefly described in Section 2.2.1.

3.3.1 Anti-Symmetry Constraints

Given a solution, an equivalent solution can be obtained by swapping the set of surgeries assigned to any pair of ORs since we consider surgical environments with identical 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 [58, 62, 74]. We add the following symmetry-breaking constraints, which are applied by Denton et al. [20] in the context of OR scheduling, to the problem:

$$x_r \geq x_{r+1} \qquad \forall r < n_R, \tag{3.4a}$$

$$\sum_{r=1}^{i} y_{ir} = 1 \qquad \forall i \leq min\{n, n_R\}, \tag{3.4b}$$

$$\sum_{q=r}^{min\{i,n_R\}} y_{iq} \leq \sum_{j=r-1}^{i-1} y_{j,r-1} \qquad \forall (i,r) : i \geq r > 1. \tag{3.4c}$$

Constraint (3.4a) breaks the symmetry with respect to ORs by introducing an arbitrary ordering. Similarly, constraints (3.4b) and (3.4c) 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 i^{th} surgery should be assigned to one of the first r ORs. Denton et al. [20] observe that these constraints have a significant impact on the solution time for a stochastic version of the bin packing problem.

3.3.2 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 algorithm that solves the master and recourse problems separately may result in feasible first-stage solutions that are second-stage infeasible. This is due to the fact that the completion time related constraints (i.e., constraints (3.3b)-(3.3f)) are in the second stage. The standard L-shaped algorithm [83] 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 algorithm (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 (3.1b)-(3.1i), (3.4), and*

$$u_j \geq u_i + d - nd(1 - \sum_{r=1}^{n_R} z_{ijr}) \qquad \forall i, j \neq i, \tag{3.5a}$$

$$u_j \geq u_i + d \qquad \forall (i, j) : \sum_{k=1}^{n_S} b_{ijk} = 1, \tag{3.5b}$$

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 (3.5a) prevents infeasible schedules with respect to the sequence within an OR. In a similar way, constraint set (3.5b) ensures the feasibility of the constructed sequence with respect to surgeons across the ORs. As a result, constraints (3.5) ensure that z yields acyclic surgery sequences. Therefore, any first-stage feasible solution that also satisfies (3.5) 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.

3.3.3 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

$$C_{ir}(\omega) = t_k + pre_i(\omega) + p_i(\omega) + post_i(\omega). \tag{3.6}$$

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

$$C_{ir}(\omega) = max \begin{cases} t_k + pre_i(\omega) + p_i(\omega) + post_i(\omega) \\ \sum_{j=1}^{n} [b_{jik} [\sum_{r=1}^{n_R} C_{jr}(\omega) - post_j(\omega)]] + s^S + p_i(\omega) + post_i(\omega). \end{cases} \tag{3.7}$$

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

$$C_{ir}(\omega) = max \begin{cases} max_j \{z_{jir} C_{jr}(\omega)\} + s^R + pre_i(\omega) + p_i(\omega) + post_i(\omega) \\ t_k + pre_i(\omega) + p_i(\omega) + post_i(\omega). \end{cases} \tag{3.8}$$

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

$$C_{ir}(\omega) = max \begin{cases} t_k + pre_i(\omega) + p_i(\omega) + post_i(\omega) \\ max_j \{z_{jir} C_{jr}(\omega)\} + s^R + pre_i(\omega) + p_i(\omega) + post_i(\omega) \\ \sum_{j=1}^{n} [b_{jik} [\sum_{r=1}^{n_R} C_{jr}(\omega) - post_j(\omega)]] + s^S + p_i(\omega) + post_i(\omega). \end{cases} \tag{3.9}$$

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

$$I_{ij}(\omega) = \sum_{r=1}^{n_R} C_{jr}(\omega) - \sum_{r=1}^{n_R} C_{ir}(\omega) + post_i(\omega) - s^S - p_j(\omega) - post_j(\omega) \qquad \forall (i,j) : \sum_{k=1}^{n_S} b_{ijk} = 1, \tag{3.10}$$

$$I_k(\omega) = \sum_{r=1}^{n_R} C_{i_k r}(\omega) - t_k - pre_{i_k}(\omega) - p_{i_k}(\omega) - post_{i_k}(\omega) \qquad \forall k, \tag{3.11}$$

$$O_r(\omega) = \max\{0, \max_i \{C_{ir}(\omega)\} - L\} \qquad \forall r. \tag{3.12}$$

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.

3.3.4 Extended Master Problem Formulation

The following is an equivalent formulation of our problem:

$$\min \sum_{r=1}^{n_R} c^f x_r + \theta \tag{3.13}$$

s.t.

$$\theta \geq \mathcal{Q}(x, y, z, t), \tag{3.14}$$

$$\text{(3.1b)-(3.1i),(3.4),(3.5).}$$

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

$$\min \sum_{r=1}^{n_R} c^f x_r + \theta \tag{3.13}$$

s.t.

$$\text{(3.1b)-(3.1i),(3.4),(3.5).}$$

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 L-shaped algorithm with the enhancements described in Sections 3.3.1-3.3.3 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 [50, 67, 75]. 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. In order 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,*

$$\theta \geq Q(\hat{x}, \hat{y}, \hat{z}, \hat{t}, \bar{\xi}(\omega)). \tag{3.15}$$

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

$$\mathcal{Q}(\hat{x}, \hat{y}, \hat{z}, \hat{t}) \geq Q(\hat{x}, \hat{y}, \hat{z}, \hat{t}, \bar{\xi}(\omega)), \tag{3.16}$$

by Jensen's inequality [48]. Moreover, we have

$$\theta \geq \mathcal{Q}(\hat{x}, \hat{y}, \hat{z}, \hat{t}) \tag{3.17}$$

since $\theta \geq \mathcal{Q}(x, y, z, t)$ is a part of our formulation (equation (3.14)). (3.15) directly follows from (3.16) and (3.17). □

We observe that these cuts are broadly applicable to two-stage stochastic programs with recourse. We use valid inequalities based on Proposition 2 in order 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

$\overline{pre}_i$: expected pre-incision duration of surgery i

$\overline{p}_i$: expected incision duration of surgery i

$\overline{post}_i$: expected post-incision 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/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, O_r be defined by the following inequalities:*

$$C_{ir} \leq M y_{ir} \qquad \forall i, r, \tag{3.18a}$$

$$C_{jr} \geq C_{ir} + s^R + \overline{pre}_j + \overline{p}_j + \overline{post}_j - M(1 - z_{ijr}) \qquad \forall i, j \neq i, r, \tag{3.18b}$$

$$\sum_{r=1}^{n_R} C_{i_k r} = t_k + I_k + \overline{pre}_{i_k} + \overline{p}_{i_k} + \overline{post}_{i_k} \qquad \forall k, \tag{3.18c}$$

$$\sum_{r=1}^{n_R} C_{ir} \geq t_k + \overline{pre}_i + \overline{p}_i + \overline{post}_i \qquad \forall (i,k) : \sum_{j=1}^{n} b_{jik} = 1, \tag{3.18d}$$

$$\sum_{r=1}^{n_R} C_{jr} = \sum_{r=1}^{n_R} C_{ir} - \overline{post}_i + s^S + \overline{p}_j + \overline{post}_j + I_{ij} \qquad \forall (i,j) : \sum_{k=1}^{n_S} b_{ijk} = 1, \tag{3.18e}$$

$$O_r \geq C_{ir} - L \qquad \forall i, r, \tag{3.18f}$$

$$C_{ir}, I_k, I_{ij}, O_r \geq 0 \qquad \forall i, j, r, k. \tag{3.18g}$$

Then,

$$\theta \geq \sum_{r=1}^{n_R} c^o O_r + \sum_{(i,j): \sum_{k=1}^{n_S} b_{ijk} = 1} c^S I_{ij} + \sum_{k=1}^{n_S} c^S I_k \tag{3.19}$$

is a valid inequality.

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)*:

$$\min \sum_{r=1}^{n_R} c^f x_r + \theta \tag{3.13}$$

s.t.

(3.1b)-(3.1i),(3.4),(3.18),(3.19).

Note that constraints (3.18) 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 (3.5) in the ERMP.

Valid inequalities in stochastic programming is explored by earlier work, including [36, 37, 53, 68]. Sanchez and Wood [68] use Jensen's based inequalities within the simulation-based approach they propose for solving two-stage SMIPs. However, their method assumes binary first stage and may require full enumeration of every feasible first-stage solution.

3.4 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.

3.4.1 Generation of Problem Instances

We use data from the Mayo Clinic 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 performs more than 2000 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 sub-procedures. Surgeons within the thoracic area perform surgeries every other day, usually with at least two staff surgeons working each day. Since Mayo Clinic is an academic teaching institution, each staff surgeon may have multiple fellows assisting in the procedures with the attending 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 Mayo Clinic and our discussions with an anesthesiologist who is working as an administrative director in the thoracic surgery department. The daily fixed cost of opening an OR, c^f, is estimated to be \$4437. The session length, L, is 9 hours/day for an OR. The overtime cost, c^o, is estimated to be \$12.37/minute, which is 50% higher than the regular OR time cost. As 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/minute. In the second case, we assume that 250 minutes of idle time is equivalent to opening an OR, and use $17.748/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. As 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 sub-procedures in each surgery,
- The ordered surgery listing of each surgeon.

For each surgical day, i.e., problem instance, we generated 500 different scenarios by sampling pre-incision, incision and post-incision durations for each surgery based on the number and type of the sub-procedures included.

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

The incision duration includes the critical portion of surgery, which is completed by the attending surgeon, as well as the non-critical portion of surgery. We used a discrete event simulation model [46] 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 are the patients. Similar to our stochastic programming model, the focus of the simulation model is also on the pre-incision, incision, and post-incision 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 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 pre-incision and post-incision 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 7 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 3.1. 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, 3 surgeons and 6 ORs.

3.4.2 Computational Performance of the Proposed Algorithms

We analyze the performance of the standard L-shaped algorithm (Figure 3.1) 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

Table 3.1: Size-based classification of problem instances for 322 surgical days.

Set No	Number of Instances	Average				Maximum			
		n	n_S	n_R	n_{Bin}	n	n_S	n_R	n_{Bin}
1	177	3.95	1.76	2.64	41.16	7	3	4	99
2	67	6.79	2.49	3.93	149.09	9	3	5	200
3	46	8.04	2.43	4.39	234.74	9	3	6	280
4	17	9.29	2.59	4.88	338.88	11	4	6	390
5	6	10.17	2.83	5.17	436.83	11	3	6	485
6	7	11.00	2.57	5.71	550.57	12	3	7	600
7	2	11.00	3.00	6.00	609.00	11	3	6	612

(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 3.2), adding optimality cuts at each integer feasible node. This approach solves the master problem only once, by adding the optimality cuts during branch 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 CPLEX 11 callable library. We conducted our experiments on 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 3.2. Set 2 and Set 3 include problem instances that could not be solved in 3 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 3 hours, i.e., the computational time limit, when calculating the average solution time.

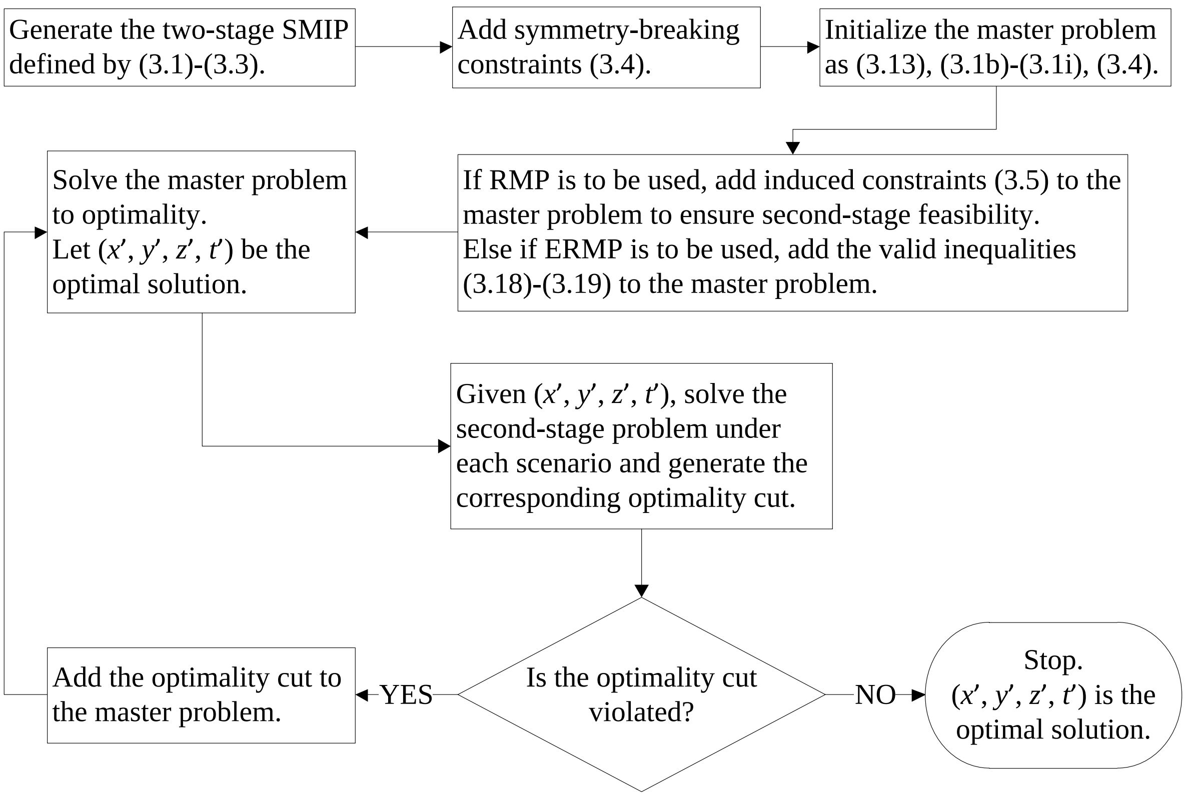

Figure 3.1: Flow chart for our solution method that uses the L-shaped algorithm.

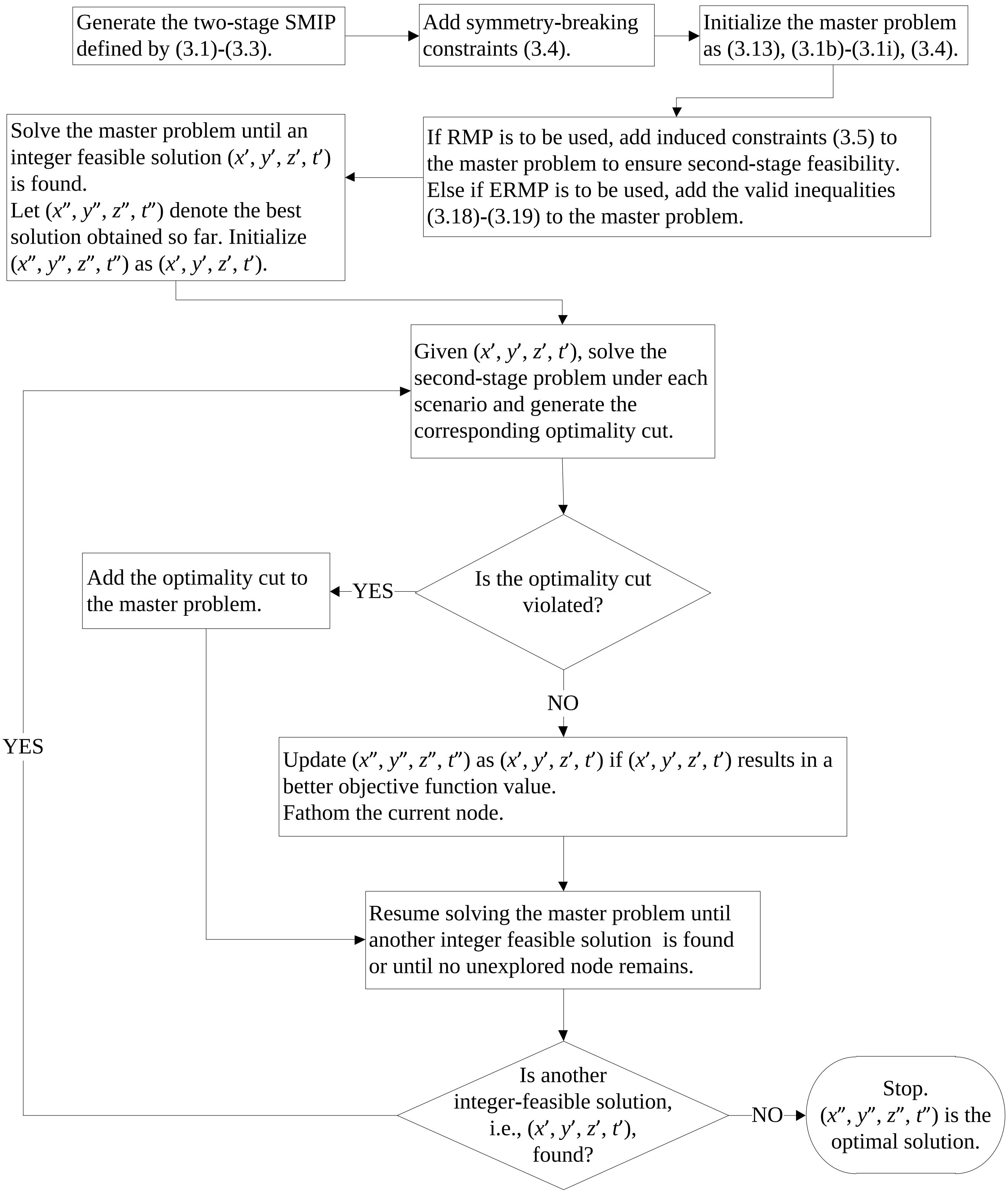

Figure 3.2: Flow chart for our solution method that uses the L-shaped based branch-and-cut algorithm.

As can be observed from Table 3.2, 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 (3.18)-(3.19) to the master problem improves the formulation considerably. Another conclusion from Table 3.2 is that the idle time cost level does not have an impact on the relative performance of the algorithms.

In Table 3.3, 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, though on the 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 employ it to solve the remaining instances of our problem. However, for larger instances that cannot be solved optimally within the 3-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 report the average and maximum solution times in Table 3.4. For some of the larger instances in Sets 5, 6 and 7, although we are able to solve the mean value problems within 3 hours, we are not able to solve the stochastic problems. The number of unsolved instances and the percentage optimality gap between the upper bound, i.e., the value of the best solution obtained within 3 hours, and the lower bound, i.e., 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 3.5. Of particular interest is the average percentage gap, which is below 3% and 5% for low and idle time costs, respectively.

Table 3.2: Computational performance of the L-shaped and L-shaped based branch-and-cut algorithms.

Idle Time Cost Level	Set No.	Number of Instances	L-Shaped Algorithm								L-Shaped Based Branch-and-Cut Algorithm			
			RMP				ERMP				RMP		ERMP	
			Solution Time (CPU Seconds)		Number of Iterations		Solution Time (CPU Seconds)		Number of Iterations		Solution Time (CPU Seconds)		Solution Time (CPU Seconds)	
			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	1689.12	>10800.00	1025.80	4226	20.16	123.48	22.48	52	776.13	6141.42	49.48	249.28
	3	15	7784.51	>10800.00	2396.53	3642	308.23	1701.73	38.73	84	7158.44	>10800.00	1016.65	10146.41
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	>10800.00	618.00	3373	24.52	90.34	41.44	142	649.23	3560.53	95.75	1281.81
	3	15	5923.05	>10800.00	1874.53	3428	289.21	843.02	66.80	152	6895.76	>10800.00	424.68	1449.48

Table 3.3: Computational performance of the proposed algorithms for 20 instances.

Instance No.	L-Shaped Algorithm				L-Shaped Based Branch-and-Cut Algorithm	
	RMP		ERMP		RMP	ERMP
	Solution Time (CPU Seconds)	Number of Iterations	Solution Time (CPU Seconds)	Number 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	0.23	11	0.31	13	0.69	1.11
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	>10800.00	3373	90.34	32	2112.92	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	3652.47	1577	32.72	19	9843.77	185.44
3b	>10800.00	2978	839.92	80	>10800.00	716.06
3c	2491.17	1040	246.03	133	2018.11	328.69
3d	1930.78	1037	35.22	22	5440.00	322.14
3e	>10800.00	3428	97.98	13	>10800.00	235.16

Table 3.4: Solution times (in CPU seconds) of mean value and stochastic problems.

Set No.	Low Idle Time Cost				High Idle Time Cost			
	Mean Value Problem		Stochastic Problem		Mean Value Problem		Stochastic Problem	
	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	2517.33	0.59	3.80	212.23	1981.09
4	11.31	66.58	1023.36	5097.61	2.84	14.86	1540.80	8034.84
5	46.02	128.42	2160.37	2969.52	11.80	35.50	6078.25	>10800.00
6	196.13	687.18	5241.69	>10800.00	34.55	141.02	4866.16	>10800.00
7	126.78	151.83	9447.07	>10800.00	6.27	8.13	9992.85	>10800.00

Table 3.5: Percentage gap values for the unsolved instances.

Set No.	Low Idle Time Cost			High Idle Time Cost		
	Number of Unsolved Instances	Percentage Gap Between Lower and Upper Bounds		Number of Unsolved Instances	Percentage Gap Between Lower and Upper Bounds	
		Average	Maximum		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%

Table 3.6: Percentage value of the stochastic solution for each problem set.

Set	Low Idle Time Cost		High Idle Time Cost	
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%

3.4.3 Value of the Stochastic Solution

In order 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 [10]. 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 3.6. 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 3.6 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 since 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 components, 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. Since we are considering a multi-criteria 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 non-dominated 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 in Table 3.7. 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 number 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 multi-criteria 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 to be the only factor that lowers the objective function value for Sets 5-7.

3.4.4 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 which prevents the sharing of ORs among surgeons by using the following first-stage constraints:

$$y_{ir} + y_{jr} \leq 1 \quad \forall r, (i, j > i) : \sum_{k=1}^{n_S} (\beta_{ijk} + \beta_{jik}) = 0, \qquad (3.20)$$

Table 3.7: Optimal solution statistics for the stochastic and mean value problems for each problem set.

	Set No.	Low Idle Time Cost: Stochastic Problem	Low Idle Time Cost: Mean Value Problem	High Idle Time Cost: Stochastic Problem	High Idle Time Cost: Mean Value Problem
Average Number 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

Table 3.8: Percentage improvement brought by OR pooling.

Set No.	Low Idle Time Cost		High Idle Time Cost	
	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%

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 (3.20) 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 $n_R \geq n_S$. This is satisfied by all of our instances.

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 due to OR pooling. We summarize the percentage improvements brought by OR pooling in Table 3.8. In Table 3.9, we compare the average number of open ORs, overtime and idle time values of the schedules generated by Setting 1 and Setting 2.

We observe from Table 3.8 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 3.9 reveal that the substantial cost reduction achieved by OR pooling is mainly attributable

Table 3.9: Optimal solution statistics for Setting 1 and Setting 2 for each problem set.

	Set No.	Low Idle Time Cost		High Idle Time Cost	
		Setting 1	Setting 2	Setting 1	Setting 2
Average Number 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

to the decrease in the average number of open ORs. The decrease in the required number of ORs might result in significant savings since the initial investment needed to build/open a new OR is estimated between $700,000 and $9,000,000 [34, 35, 92, 94].

3.4.5 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., pre-incision and post-incision 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. In order to demonstrate this, we consider a surgical day that includes 6 surgeries, 1 surgeon and 6 available ORs. We consider different levels of OR turnover time, in a range changing from 0 to 2 times of the original turnover time (which is 30 minutes). As for the parallelizable portion of the surgery, we consider a range from 0 to 1 times of the original duration. The original parallelizable portion 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 3.3 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 level and parallelizable portion level, 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.

3.5 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

Figure 3.3: Optimal number of ORs to open at different levels of OR turnover time and parallelizable portion of surgeries.

Low Idle Time Cost

OR Turnover Time Coefficient \ Parallelizable Portion 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

High Idle Time Cost

OR Turnover Time Coefficient \ Parallelizable Portion 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

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 (pre-incision, incision and post-incision), 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 decrease 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 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 data collected from 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/or parallelizable portion of surgeries increase. 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 multi-criteria optimization problem for future work. However, we note that our model, methodological results, and general insights are relevant to most providers of surgical care.

4.0 OPERATING ROOM RESCHEDULING UNDER UNCERTAINTY

4.1 MOTIVATION AND CONTRIBUTIONS

A surgical suite is a dynamic environment, and making day of surgery adjustments on OR schedules is inevitable. Revising OR schedules during the surgical day in order to mitigate the impact of unexpected events is very common in practice [3, 26, 86]. Therefore, a realistic model of the surgery delivery process should include both anticipative (proactive) and adaptive (reactive) decisions in a stochastic framework. The anticipative decisions can be used to design the baseline schedule for the surgical day, whereas the adaptive decisions can be used to determine the rescheduling actions to be taken based on revealed information (i.e., resolved uncertainty).

Despite the importance of the adaptive decisions on the day of surgery, there are only few studies that consider rescheduling activities. Dexter [21] proposes a statistical method to predict whether moving the last surgery of the day to another OR would reduce the overtime labor costs. Wachtel and Dexter [86] present several interventions including rescheduling to reduce tardiness in surgical suites. Stuart et al. [80] study a single-OR rescheduling problem where the objective is minimizing the cancelation of elective surgeries, maximizing the throughput of emergent surgeries, and minimizing deviation from the original schedule. They treat these objectives by combining them into a single objective function, and they propose an approach where a deterministic model is solved at the completion of each surgery and the schedule is revised accordingly. The focus of these earlier studies is only on the adaptive decisions. However, the existence of the adaptive decisions may influence how the anticipative decisions are made. Therefore, integrating both types of decisions is essential to developing more accurate models of the process.

In this chapter, we explore OR scheduling problem under uncertainty where surgeries are allowed to be rescheduled at a prespecified time point during the day. Permitting rescheduling greatly complicates the problem. Therefore, our approaches do not consider surgeon related constraints and performance measures. Instead, we choose to focus only on ORs since OR-related decisions are the most important decisions from a financial perspective. Thus, the problem investigated in this chapter is an extension of the stochastic surgery allocation problem considered by Denton et al. [20] to the case where the allocation decisions are allowed to be revised during the day.

We formulate the problem as a three-stage SMIP, where rescheduling activities are modeled as the recourse decisions in the second-stage. A special case of our model, where the rescheduling decisions are made under perfect information, is a two-stage SMIP. We solve this two-stage SMIP by using the integer L-shaped algorithm and a PH-based heuristic method, and draw important conclusions from our numerical results.

The main contributions of our work presented in this chapter can be summarized as follows:

- We model the stochastic multi-OR rescheduling problem, integrating anticipative and adaptive decisions.
- We estimate the potential benefit of rescheduling, and the value of anticipative and adaptive decisions under different problem settings including outpatient and inpatient surgical environments.
- We explore the impact of using different surgery sequencing rules on the value of rescheduling.

The remainder of this chapter is organized as follows. In Sections 4.2 and 4.3, we describe our models. In Section 4.4, we present our solution methods. Finally, we discuss our computational results and summarize our findings in Sections 4.5 and 4.6, respectively.

4.2 PROBLEM DEFINITION AND MATHEMATICAL FORMULATION

Surgery allocation decisions can be revised upon the completion of each surgery in practice. We consider the simplest case where rescheduling decisions are made only once during the day. Considering this setting, our model is composed of three stages (Figure 4.1). As in the models introduced in [20] and Chapter 3 of this dissertation, the first stage takes place before the surgical day starts, i.e., before the resolution of uncertainty in surgery durations. The number of ORs to open and surgery-to-OR assignments are determined in the first stage, which is the same set of first-stage decisions as in the stochastic OR allocation model presented in [20]. The second stage takes place at a *rescheduling point*, where some of the surgeries are completed or partially completed depending on the resolved uncertainty in surgery durations. Based on this additional information revealed in the second stage, rescheduling decisions are made: allocation of the remaining surgeries to ORs is revised. Uncertainty in the durations of the remaining surgeries is resolved, and OR overtime levels become known in the third stage. The overall objective is to minimize the total fixed cost of opening ORs and the expected overtime cost.

Although the actual rescheduling point can be at any time, we refer to the period before/after the rescheduling point as *morning/afternoon*. Reallocation of the remaining surgeries to the open ORs is the only set of rescheduling decisions, which are recourse decisions made in the second-stage. Therefore, we use *reallocation, rescheduling,* and *recourse* interchangeably when referring to the second-stage decisions or the time point at which these decisions are made.

The state of the system, i.e., which surgeries are completed or partially completed, at the rescheduling point for a realized scenario depends on the sequencing decisions as well as the allocation decisions. We assume that surgeries in each OR are performed in an order based on a predetermined sequence.

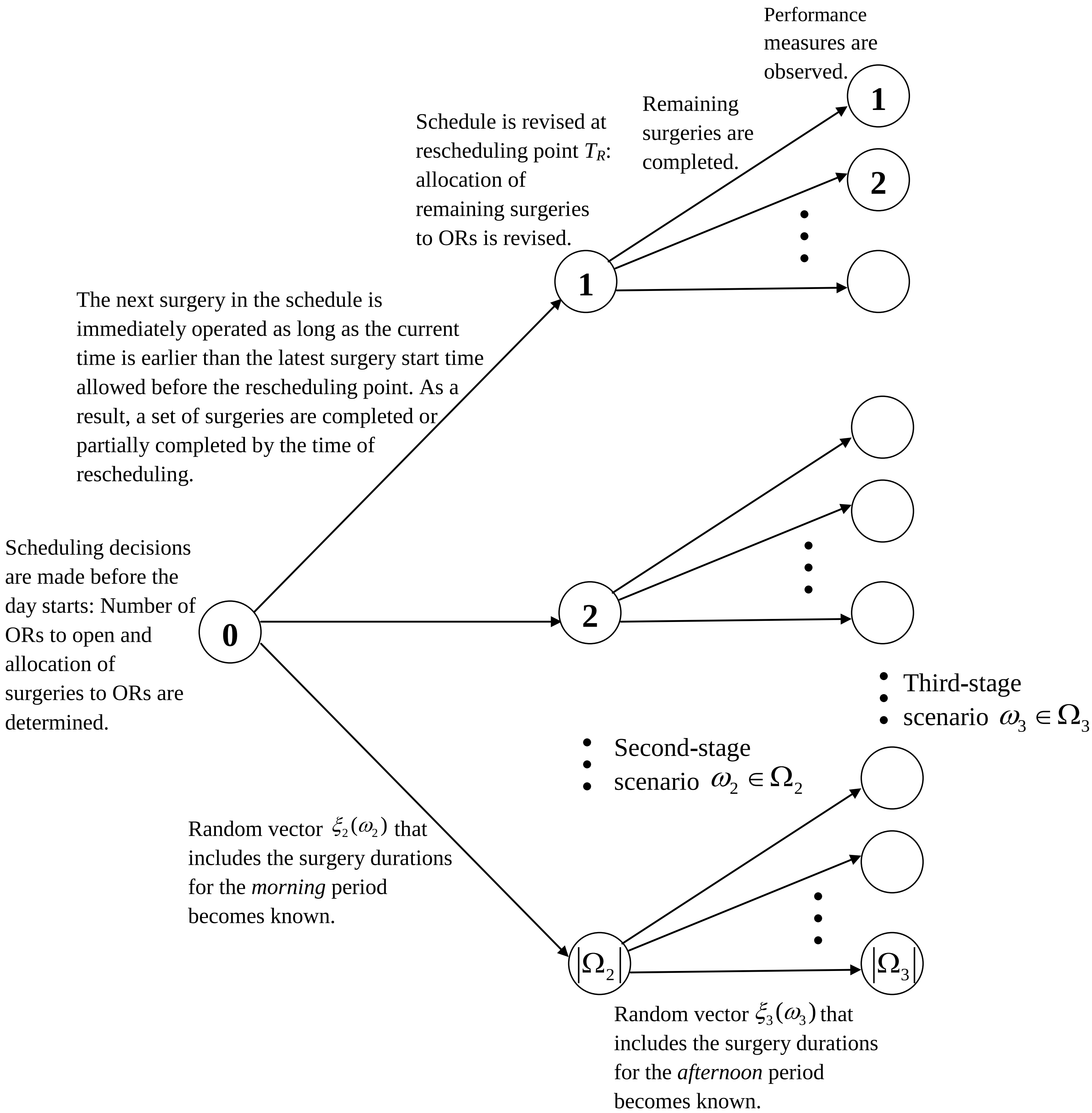

Figure 4.1: Scenario tree for the OR rescheduling problem where the initial schedule is revised at T_R.

We use the following notation in our formulation:

Indices

i, j: Surgery indices.

q, r: OR indices.

ω_2, ω_3: Scenario indices for second- and third-stage scenarios, respectively.

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.

T_R: Length of the portion of the surgical day before the rescheduling point. We assume that the surgical day starts at time 0, and T_R denotes the time of rescheduling.

T_L: Latest start time allowed for the surgeries operated in the morning such that $T_L \leq T_R$. Before T_L, the next surgery in the sequence immediately starts after the completion of the previous one in each OR. After T_L, on the other hand, none of the remaining surgeries start until the schedule is revised. T_L is very close to the rescheduling point T_R, and reallocating the immediately succeeding surgeries might be more beneficial instead of operating them in their current ORs. We assume that the probability of having a surgery starting exactly at T_L is negligibly small.

T_E: Earliest start time allowed for the surgeries operated in the afternoon such that $T_E \geq T_R$. Rescheduling decisions are not made instantaneously. Even though we consider the state of the system at T_R to reschedule the surgeries, constructing a revised schedule would take some time which we denote by $T_E - T_R$. Therefore, the earliest time that the revised schedule can be implemented is T_E (Figure 4.2).

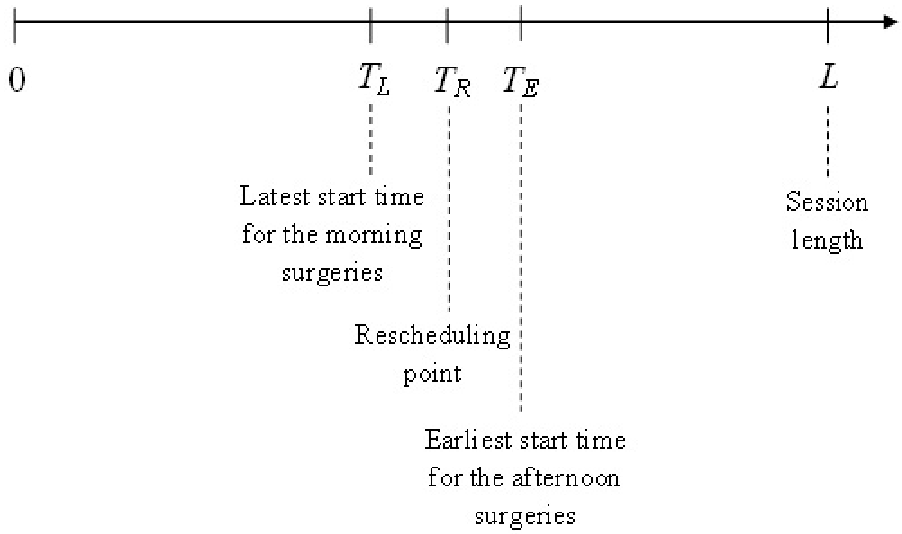

Figure 4.2: Important time points during a surgical day.

Problem Instance Related Parameters

n:	Total number of surgeries to be scheduled.
n_R:	Total number of available ORs.
b_{ij}:	A binary parameter which denotes that surgery i should be operated before surgery j if they are allocated to the same OR. The value of this parameter is calculated according to a predetermined sequencing rule.
$p_i^M(\omega_2)$:	Duration of surgery i under second-stage scenario ω_2 if it starts in the morning.
$p_i^A(\omega_3)$:	Duration of surgery i under third-stage scenario ω_3 if it starts in the afternoon.

We define $\omega_2 \in \Omega_2$ and $\omega_3 \in \Omega_3$ to be the indices of second- and third-stage scenarios, respectively. Because the third stage is the last stage in our formulation, ω_3 is also the index of the complete scenario for the surgical day. Given n surgeries, we obtain random vectors $\xi_2(\omega_2) = \{p_1^M(\omega_2), ..., p_n^M(\omega_2)\}$ and $\xi_3(\omega_3) = \{p_1^A(\omega_3), ..., p_n^A(\omega_3)\}$. We denote the finite support of these vectors by $\Xi_2 \in \mathbb{R}_+^n$ and $\Xi_3 \in \mathbb{R}_+^n$, respectively.

Each third-stage scenario emanates from an ancestor second-stage scenario. Letting $\omega_2(\omega_3)$ denote the ancestor scenario of ω_3, the duration of surgery i under scenario ω_3 is:

- $p_i^M(\omega_2(\omega_3))$ if it starts before T_L,
- $p_i^A(\omega_3)$ if it starts after T_E.

As implied by the definition, we assume that duration of a surgery becomes known (in other words, uncertainty is resolved) as soon as it starts. Considering the different phases of surgery would increase the accuracy of an OR scheduling model only when surgeons are also taken into account among the scheduled resources. Therefore, instead of considering pre-incision, incision, post-incision phases separately as in Chapter 3, we only consider the total surgery duration (p_i^M and p_i^A). Moreover, rather than explicitly modeling the OR turnover time, we adjust the surgery durations to account for the setup and cleaning activities.

First-Stage 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 in the initial schedule.

Second-Stage Decision Variables

$C_{ir}(\omega_2)$: Completion time for surgery i in OR r under second-stage scenario ω_2 according to the initial schedule.

$y_{ir}^M(\omega_2)$: Binary decision variable denoting whether surgery i is allocated to OR r in the initial schedule, and its start time is before T_L under second-stage scenario ω_2.

$y_{ir}^A(\omega_2)$: Binary decision variable denoting whether surgery i is allocated to OR r in the revised schedule under second-stage scenario (ω_2).

Third-Stage Decision Variables

$O_r(\omega_3)$: Overtime in OR r, with respect to session length L under scenario ω_3.

Let $x, y, C(\omega_2), y^A(\omega_2), y^M(\omega_2), O(\omega_3)$ denote the vector or matrix form of the decision variables defined above.

Using the above notation we formulate the model as follows:

$$\min \sum_{r=1}^{n_R} c^f x_r + \mathcal{Q}_1(x,y) \tag{4.1a}$$

s.t.

$$y_{ir} \leq x_r \qquad \forall i, r, \tag{4.1b}$$

$$\sum_{r=1}^{n_R} y_{ir} = 1 \qquad \forall i, \tag{4.1c}$$

$$x_r \geq x_{r+1} \qquad \forall r < n_R, \tag{4.1d}$$

$$\sum_{r=1}^{i} y_{ir} = 1 \qquad \forall i \leq min\{n, n_R\}, \tag{4.1e}$$

$$\sum_{q=r}^{min\{i,n_R\}} y_{iq} \leq \sum_{j=r-1}^{i-1} y_{j,r-1} \qquad \forall (i,r) : i \geq r > 1, \tag{4.1f}$$

$$x_r, y_{ir} \in \{0,1\} \qquad \forall i, r, \tag{4.1g}$$

where

$$\mathcal{Q}_1(x,y) = E_{\xi_2}\left[Q_1(x,y,\xi_2(\omega_2))\right] \tag{4.2}$$

and

$$Q_1(x,y,\xi_2(\omega_2)) = \min \mathcal{Q}_2(y^M(\omega_2), y^A(\omega_2)) \tag{4.3a}$$

s.t.

$$C_{ir}(\omega_2) \leq M y_{ir} \qquad \forall i, r, \tag{4.3b}$$

$$C_{ir}(\omega_2) \geq \sum_{j=1}^{n} b_{ji} p_j^M(\omega_2) y_{jr} + p_i^M(\omega_2) y_{ir} - M(1 - y_{ir}) \qquad \forall i, r, \tag{4.3c}$$

$$C_{ir}(\omega_2) \leq \sum_{j=1}^{n} b_{ji} p_j^M(\omega_2) y_{jr} + p_i^M(\omega_2) y_{ir} \qquad \forall i, r, \tag{4.3d}$$

$$T_L - \left[\sum_{r=1}^{n_R} C_{ir}(\omega_2) - p_i^M(\omega_2)\right] \leq T_L \sum_{r=1}^{n_R} y_{ir}^M(\omega_2) \qquad \forall i, \tag{4.3e}$$

$$\left[\sum_{r=1}^{n_R} C_{ir}(\omega_2) - p_i^M(\omega_2)\right] - T_L \leq M\left[1 - \sum_{r=1}^{n_R} y_{ir}^M(\omega_2)\right] \qquad \forall i, \tag{4.3f}$$

$$y_{ir}^M(\omega_2) \leq y_{ir} \qquad \forall i, r, \tag{4.3g}$$

$$\sum_{r=1}^{n_R} y_{ir}^M(\omega_2) + \sum_{r=1}^{n_R} y_{ir}^A(\omega_2) = 1 \qquad \forall i, \tag{4.3h}$$

$$y_{ir}^A(\omega_2) \le x_r \qquad \forall i, r, \tag{4.3i}$$

$$y_{ir}^M(\omega_2), y_{ir}^A(\omega_2) \in \{0, 1\} \qquad \forall i, r, \tag{4.3j}$$

$$C_{ir}(\omega_2) \ge 0 \qquad \forall i, r, \tag{4.3k}$$

where

$$\mathcal{Q}_2(y^M(\omega_2), y^A(\omega_2)) = E_{\xi_3|\xi_2}\left[Q_2(y^M(\omega_2), y^A(\omega_2), \xi_3(\omega_3))\right] \tag{4.4}$$

and

$$Q_2(y^M(\omega_2), y^A(\omega_2), \xi_3(\omega_3)) = \min \sum_{r=1}^{n_R} c^o O_r(\omega_3) \tag{4.5a}$$

s.t.

$$O_r(\omega_3) \ge \sum_{i=1}^{n} p_i^M(\omega_2) y_{ir}^M(\omega_2) + \sum_{i=1}^{n} p_i^A(\omega_3) y_{ir}^A(\omega_2) - L \qquad \forall r, \tag{4.5b}$$

$$O_r(\omega_3) \ge T_E + \sum_{i=1}^{n} p_i^A(\omega_3) y_{ir}^A(\omega_2) - L \qquad \forall r, \tag{4.5c}$$

$$O_r(\omega_3) \ge 0 \qquad \forall r. \tag{4.5d}$$

The objective function (4.1a) is the sum of the first-stage cost and the expected cost of second and third stages over all scenarios. The fixed cost of opening ORs is incurred in the first stage, and OR overtime costs are incurred in the third stage.

Constraints (4.1b) and (4.1c) 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. Since we consider the case where the ORs are identical and therefore interchangeable, our problem has complete symmetry with respect to ORs. Constraints (4.1d)-(4.1e) are symmetry-breaking constraints introduced by Denton et al. [20]. Constraints (4.1g) define binary restrictions for the first-stage decision variables.

The second-stage problem for a given x, y and $\xi_2(\omega_2)$ is formulated explicitly by (4.3). The completion time of a surgery in an OR is 0 unless it is assigned to that OR, which is enforced by constraint (4.3b). If surgery i is assigned to OR r (i.e., $y_{ir} = 1$), then $C_{ir}(\omega_2)$

should be equal to the total duration of surgery i and the preceding surgeries in OR r. This is ensured by constraints (4.3c) and (4.3d). Constraints (4.3e), (4.3f) and (4.3g) ensure that $y_{ir}^M(\omega_2) = 1$ if and only if $y_{ir} = 1$ and the start time of surgery i under second-stage scenario is before T_L according to the initial schedule. The M parameter used in constraint sets (4.3b), (4.3c), and (4.3f) is an upper bound on the surgery completion times (and hence an upper bound on the surgery start times as well). If a surgery does not start before T_L under second-stage scenario ω_2 (i.e., $\sum_{r=1}^{n_R} y_{ir}^M(\omega_2) = 0$), then the allocation of this surgery to the open ORs is among the rescheduling decisions and this is imposed by constraint set (4.3h). Constraints (4.3i) guarantee that allocation of a surgery to an OR is among the viable rescheduling decisions only if the considered OR is open. Constraints (4.3j) and (4.3k) define the binary and nonnegativity restrictions for the second-stage decision variables.

For a given set of rescheduling decision variables $y^M(\omega_2)$ and $y^A(\omega_2)$, the third-stage model (4.5) calculates the minimum overtime under scenario ω_3. We refer to the model (4.1)-(4.5) as the three-stage OR rescheduling problem (3S-ORRP).

4.3 RESCHEDULING UNDER PERFECT INFORMATION

Due to the complexities involved in accurately capturing the uncertainty and solving the problem in a three-stage setting, we consider a special case of 3S-ORRP, where each second-stage scenario has one descendant scenario (i.e., third-stage scenario) and $p_i^A(\omega_3) = p_i^M(\omega_2(\omega_3))$ $\forall \omega_3$. This leads to a two-stage setting where we assume that the surgery durations become known once the day starts and the rescheduling decisions are made under perfect information. In this setting, the allocation decisions are made before the day starts, and revised at the rescheduling point for the remaining surgeries (Figure 4.3). Due to the assumption that the uncertainty in surgery durations is completely resolved before the rescheduling point, this special case represents a relaxation of the real problem and gives a lower bound on the minimum total expected cost that can be attained by making rescheduling decisions before the complete resolution of uncertainty. Therefore, the difference between the expected costs of the solutions of the two-stage scheduling models with and without rescheduling activities provides an estimate of the upper bound on the value of rescheduling.

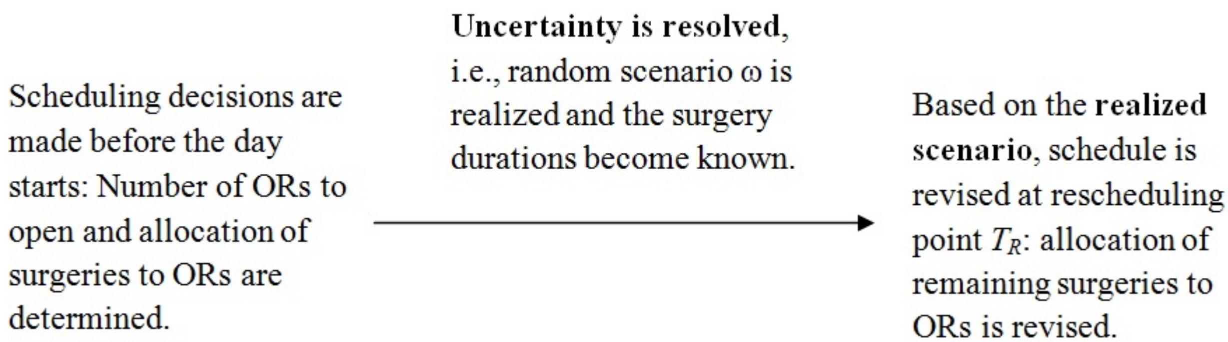

Figure 4.3: Rescheduling under perfect information.

The two-stage rescheduling model we describe in this section includes only one set of scenarios for the random parameters instead of the two different sets of scenarios (i.e., morning and afternoon durations) introduced in 3S-ORRP. Combining the second and third stages of 3S-ORRP and letting $p_i(\omega)$ denote the random duration of surgery i where ω is the scenario index, we have the following two-stage SMIP as a special case of 3S-ORRP. We refer to this model as the two-stage OR rescheduling problem (2S-ORRP). The formulation is as follows:

$$\min \sum_{r=1}^{n_R} c^f x_r + \mathcal{Q}(x,y) \tag{4.6a}$$

s.t.

$$y_{ir} \leq x_r \qquad \forall i, r, \tag{4.6b}$$

$$\sum_{r=1}^{n_R} y_{ir} = 1 \qquad \forall i, \tag{4.6c}$$

$$x_r \geq x_{r+1} \qquad \forall r < n_R, \tag{4.6d}$$

$$\sum_{r=1}^{i} y_{ir} = 1 \qquad \forall i \leq min\{n, n_R\}, \tag{4.6e}$$

$$\sum_{q=r}^{min\{i,n_R\}} y_{iq} \leq \sum_{j=r-1}^{i-1} y_{j,r-1} \qquad \forall (i,r) : i \geq r > 1, \tag{4.6f}$$

$$x_r, y_{ir} \in \{0,1\} \qquad \forall i,r, \tag{4.6g}$$

where

$$\mathcal{Q}(x,y) = E_\xi \left[Q(x,y,\xi(\omega)) \right] \tag{4.7}$$

and

$$Q(x,y,\xi(\omega)) = \min \sum_{r=1}^{n_R} c^o O_r(\omega) \tag{4.8a}$$

s.t.

$$C_{ir}(\omega) \leq M y_{ir} \qquad \forall i,r, \tag{4.8b}$$

$$C_{ir}(\omega) \geq \sum_{j=1}^{n} b_{ji} p_j(\omega) y_{jr} + p_i(\omega) y_{ir} - M(1 - y_{ir}) \qquad \forall i,r, \tag{4.8c}$$

$$C_{ir}(\omega) \leq \sum_{j=1}^{n} b_{ji} p_j(\omega) y_{jr} + p_i(\omega) y_{ir} \qquad \forall i,r, \tag{4.8d}$$

$$T_L - \left[\sum_{r=1}^{n_R} C_{ir}(\omega) - p_i(\omega) \right] \leq T_L \sum_{r=1}^{n_R} y_{ir}^M(\omega) \qquad \forall i, \tag{4.8e}$$

$$\left[\sum_{r=1}^{n_R} C_{ir}(\omega) - p_i(\omega) \right] - T_L \leq M \left[1 - \sum_{r=1}^{n_R} y_{ir}^M(\omega) \right] \qquad \forall i, \tag{4.8f}$$

$$y_{ir}^M(\omega) \leq y_{ir} \qquad \forall i,r, \tag{4.8g}$$

$$\sum_{r=1}^{n_R} y_{ir}^M(\omega) + \sum_{r=1}^{n_R} y_{ir}^A(\omega) = 1 \qquad \forall i, \tag{4.8h}$$

$$y_{ir}^A(\omega) \leq x_r \qquad \forall i,r, \tag{4.8i}$$

$$O_r(\omega) \geq \sum_{i=1}^{n} p_i(\omega) y_{ir}^M(\omega) + \sum_{i=1}^{n} p_i(\omega) y_{ir}^A(\omega) - L \qquad \forall r, \tag{4.8j}$$

$$O_r(\omega) \geq T_E + \sum_{i=1}^{n} p_i(\omega) y_{ir}^A(\omega) - L \qquad \forall r, \tag{4.8k}$$

$$y_{ir}^M(\omega), y_{ir}^A(\omega) \in \{0,1\} \qquad \forall i,r, \tag{4.8l}$$

$$C_{ir}(\omega), O_r(\omega) \geq 0 \qquad \forall i,r. \tag{4.8m}$$

The stochastic OR allocation problem studied by Denton et al. [20] is a special case of 2S-ORRP where rescheduling is not permitted (i.e., $y^M_{ir}(\omega) + y^A_{ir}(\omega) = y_{ir}\ \forall i, r, \omega$). As we refer to this stochastic OR allocation problem at various places in the remainder of this dissertation, we now present the corresponding formulation using our notation.

$$\min \sum_{r=1}^{n_R} c^f x_r + \mathcal{Q}^D(x, y) \tag{4.9a}$$

s.t.

$$y_{ir} \leq x_r \qquad \forall i, r, \tag{4.9b}$$

$$\sum_{r=1}^{n_R} y_{ir} = 1 \qquad \forall i, \tag{4.9c}$$

$$x_r \geq x_{r+1} \qquad \forall r < n_R, \tag{4.9d}$$

$$\sum_{r=1}^{i} y_{ir} = 1 \qquad \forall i \leq min\{n, n_R\}, \tag{4.9e}$$

$$\sum_{q=r}^{min\{i,n_R\}} y_{iq} \leq \sum_{j=r-1}^{i-1} y_{j,r-1} \qquad \forall (i, r) : i \geq r > 1, \tag{4.9f}$$

$$x_r, y_{ir} \in \{0, 1\} \qquad \forall i, r, \tag{4.9g}$$

where

$$\mathcal{Q}^D(x, y) = E_\xi \left[Q^D(x, y, \xi(\omega)) \right] \tag{4.10}$$

and

$$Q^D(x, y, \xi(\omega)) = \min \sum_{r=1}^{n_R} c^o O_r(\omega) \tag{4.11a}$$

s.t.

$$O_r(\omega) \geq \sum_{i=1}^{n} p_i(\omega) y_{ir}(\omega) - L \qquad \forall r, \tag{4.11b}$$

$$O_r(\omega) \geq 0 \qquad \forall i, r. \tag{4.11c}$$

We refer to the model (4.9)-(4.11) as the two-stage OR scheduling problem (2S-ORSP).

Table 4.1: Important characteristics of the problems considered in this study.

	Number of Stages	Resources	Scheduling Decisions	Rescheduling Decisions
2S-ORSP	2	ORs	Number of ORs to open Surgery-to-OR allocation	-
2S-ORSP′	2	ORs Surgeons	Number of ORs to open Surgery-to-OR allocation Surgery sequencing Start times for surgeons	-
3S-ORRP	3	ORs	Number of ORs to open Surgery-to-OR allocation	Surgery-to-OR reallocation
2S-ORRP	2	ORs	Number of ORs to open Surgery-to-OR allocation	Surgery-to-OR reallocation

If there were no uncertainty in surgery durations, scheduling decisions could be made under perfect information, and the rescheduling activities would be unnecessary. Therefore, solving the deterministic versions of 2S-ORSP and 2S-ORRP (i.e., under a particular scenario) results in obtaining the same optimal objective function value, and the optimal first-stage solution of 2S-ORSP is also an optimal first-stage solution of 2S-ORRP under a deterministic setting.

Let 2S-ORSP′ denote the problem investigated in Chapter 3, which is an extension of 2S-ORSP to the case where surgeons are also treated as resources, and surgery sequencing decisions and start times for surgeons are also among the decisions to be made. Table 4.1 summarizes the problem setting, resources, and decisions considered in 2S-ORSP, 2S-ORSP′, 3S-ORRP, and 2S-ORRP.

2S-ORRP is a special case of 3S-ORRP where rescheduling decisions are made under perfect information. 2S-ORSP is a special case of 2S-ORRP where rescheduling is not permitted, and also a special case of 2S-ORSP′ where surgeons are not among the considered resources.

4.4 SOLUTION METHODS

We use the integer L-shaped algorithm [52] and a PH-based heuristic [15, 66] to solve 2S-ORRP, which is a two-stage SMIP with pure binary first-stage and mixed-integer second-stage, and relatively complete recourse. We describe these solution methods in Sections 4.4.1 and 4.4.2.

4.4.1 The Integer L-Shaped Algorithm for 2S-ORRP

The integer L-shaped algorithm is a stage-wise decomposition method where a master problem is solved within a branch-and-cut framework, and optimality cuts are added at every integer feasible node. For 2S-ORRP, the algorithm starts by solving the following initial RMP:

$$\min \sum_{r=1}^{n_R} c^f x_r + \theta \tag{4.12}$$

s.t.

(4.6b)-(4.6g).

Letting x^N and y^N be the first-stage solution represented by the integer-feasible node N of the branch-and-cut tree, the optimality cut generated at this node is defined as:

$$\begin{aligned}\theta \geq & (\mathcal{Q}(x^N, y^N) - L_{\mathcal{Q}}) \left[\left(\sum_{r \in S_X^N} x_r - \sum_{r \notin S_X^N} x_r \right) + \left(\sum_{(i,r) \in S_Y^N} y_{ir} - \sum_{(i,r) \notin S_Y^N} y_{ir} \right) \right] \\ & - (\mathcal{Q}(x^N, y^N) - L_{\mathcal{Q}})(|S_X^N| + |S_Y^N| - 1) + L_{\mathcal{Q}},\end{aligned} \tag{4.13}$$

where $\mathcal{Q}(x^N, y^N)$ is the corresponding expected second-stage value, $S_X^N = \{r | x_r^N = 1\}$, $S_Y^N = \{(i,r) | y_{ir}^N = 1\}$, and $L_{\mathcal{Q}}$ is a finite value satisfying $L_{\mathcal{Q}} \leq \min_{x,y}\{\mathcal{Q}(x,y) | (4.6b)\text{-}(4.6g)\}$. In our particular implementation, we use $L_{\mathcal{Q}}$ as the continuous relaxation of $\min_{x,y}\{\mathcal{Q}(x,y) | (4.6b)\text{–}(4.6g)\}$.

Since there are a finite number of first-stage feasible solutions, and the set of cuts (4.13) generated for all first-stage feasible solutions describe a valid set of optimality cuts [52], the integer L-shaped algorithm finds the optimal solution in a finite number of steps [52].

Any continuous L-shaped optimality cut [83] provides a lower bound on $\mathcal{Q}(x, y)$ [52]. Therefore, besides (4.13), we also generate the standard L-shaped cuts (by using the dual solution of the linear relaxation of the second-stage problem) in our implementation.

Note that the recourse function is not convex in ξ due to the existence of integer variables in the second stage and the scenario-dependent recourse matrix. Therefore, the Jensen's based valid inequalities introduced in Chapter 3 are not applicable to 2S-ORRP.

For a given first-stage feasible solution, $C_{ir}(\omega)$ variables for any scenario ω are completely described by (4.8b)-(4.8d), and hence can be computed without solving the second-stage subproblem. Considering the computed $C_{ir}(\omega)$ values and the integrality requirements, $y^M_{ir}(\omega)$ variables for any scenario ω are completely described by (4.8e)-(4.8g), and therefore, they also can be obtained without solving the second-stage subproblem. When calculating the expected second-stage cost at integer-feasible node N, we first calculate the values of these variables and then solve the second-stage subproblems given these values. As a result, the second-stage problem becomes easier to solve, which enhances the integer L-shaped algorithm.

4.4.2 A Progressive Hedging-Based Heuristic for 2S-ORRP

The PHA [66] is an iterative method, where scenario subproblems are solved independently at each iteration, and nonanticipativity constraints are enforced progressively throughout the iterations. In order to solve large instances of 2S-ORRP, we employ a PH-based heuristic [15].

Considering the previously defined notation, and letting π_ω be the realization probability of scenario ω, and $x_r(\omega)$ and $y_{ir}(\omega)$ be the first-stage variables under scenario ω, the extensive form of 2S-ORRP can be formulated as:

$$\min \sum_{\omega \in \Omega} \pi_\omega \left[\sum_{r=1}^{n_R} c^f x_r(\omega) + \sum_{r=1}^{n_R} c^o O_r(\omega) \right] \tag{4.14a}$$

s.t.

$$y_{ir}(\omega) \leq x_r(\omega) \qquad \forall i, r, \omega, \tag{4.14b}$$

$$\sum_{r=1}^{n_R} y_{ir}(\omega) = 1 \qquad \forall i, \omega, \tag{4.14c}$$

$$x_r(\omega) \geq x_{r+1}(\omega) \qquad \forall r < n_R, \omega, \tag{4.14d}$$

$$\sum_{r=1}^{i} y_{ir}(\omega) = 1 \qquad \forall i \leq min\{n, n_R\}, \omega, \tag{4.14e}$$

$$\sum_{q=r}^{min\{i,n_R\}} y_{iq}(\omega) \leq \sum_{j=r-1}^{i-1} y_{j,r-1}(\omega) \qquad \forall (i,r), \omega : i \geq r > 1, \tag{4.14f}$$

$$C_{ir}(\omega) \leq M y_{ir}(\omega) \qquad \forall i, r, \omega, \tag{4.14g}$$

$$C_{ir}(\omega) \geq \sum_{j=1}^{n} b_{ji} p_j(\omega) y_{jr}(\omega) + p_i(\omega) y_{ir}(\omega) - M(1 - y_{ir}(\omega)) \qquad \forall i, r, \omega, \tag{4.14h}$$

$$C_{ir}(\omega) \leq \sum_{j=1}^{n} b_{ji} p_j(\omega) y_{jr}(\omega) + p_i(\omega) y_{ir}(\omega) \qquad \forall i, r, \omega, \tag{4.14i}$$

$$T_L - \left[\sum_{r=1}^{n_R} C_{ir}(\omega) - p_i(\omega)\right] \leq T_L \sum_{r=1}^{n_R} y_{ir}^M(\omega) \qquad \forall i, \omega, \tag{4.14j}$$

$$\left[\sum_{r=1}^{n_R} C_{ir}(\omega) - p_i(\omega)\right] - T_L \leq M \left[1 - \sum_{r=1}^{n_R} y_{ir}^M(\omega)\right] \qquad \forall i, \omega, \tag{4.14k}$$

$$y_{ir}^M(\omega) \leq y_{ir}(\omega) \qquad \forall i, r, \omega, \tag{4.14l}$$

$$\sum_{r=1}^{n_R} y_{ir}^M(\omega) + \sum_{r=1}^{n_R} y_{ir}^A(\omega) = 1 \qquad \forall i, \omega, \tag{4.14m}$$

$$y_{ir}^A(\omega) \leq x_r(\omega) \qquad \forall i, r, \omega, \tag{4.14n}$$

$$O_r(\omega) \geq \sum_{i=1}^{n} p_i(\omega) y_{ir}^M(\omega) + \sum_{i=1}^{n} p_i(\omega) y_{ir}^A(\omega) - L \qquad \forall r, \omega, \tag{4.14o}$$

$$O_r(\omega) \geq T_E + \sum_{i=1}^{n} p_i(\omega) y_{ir}^A(\omega) - L \qquad \forall r, \omega, \tag{4.14p}$$

$$x_r(\omega) = x_r \qquad \forall r, \omega, \tag{4.14q}$$

$$y_{ir}(\omega) = y_{ir} \qquad \forall i, r, \omega, \tag{4.14r}$$

$$x_r, y_{ir}, x_r(\omega), y_{ir}(\omega), y_{ir}^M(\omega), y_{ir}^A(\omega) \in \{0, 1\} \qquad \forall i, r, \omega, \tag{4.14s}$$

$$C_{ir}(\omega), O_r(\omega) \geq 0 \qquad \forall i, r, \omega. \tag{4.14t}$$

In the above formulation, (4.14q)-(4.14r) are the nonanticipativity constraints, and they ensure that the same first-stage decisions are made under each scenario. If these constraints are relaxed by using an augmented Lagrangian strategy (as proposed by Rockafellar and Wets [66]), the objective function of the problem becomes:

$$\min \sum_{\omega\in\Omega} \pi_\omega \Bigg[\sum_{r=1}^{n_R} c^f x_r(\omega) + \sum_{r=1}^{n_R} c^o O_r(\omega) + \sum_{r=1}^{n_R} \lambda^X_{r\omega}(x_r(\omega) - x_r) + \sum_{i=1}^{n}\sum_{r=1}^{n_R} \lambda^Y_{ir\omega}(y_{ir}(\omega) - y_{ir})$$
$$+ \frac{1}{2}\sum_{r=1}^{n_R} \rho(x_r(\omega) - x_r)^2 + \frac{1}{2}\sum_{i=1}^{n}\sum_{r=1}^{n_R} \rho(y_{ir}(\omega) - y_{ir})^2 \Bigg], \tag{4.15}$$

where $\lambda^X_{r\omega}$ and $\lambda^Y_{ir\omega}$ are Lagrangian multipliers for the relaxed constraints (4.14q)-(4.14r), and ρ is a positive penalty coefficient.

Since x_r is a binary variable, $x_r^2 = x_r$. Such a substitution is also valid for $y_{ir}, x_r(\omega)$ and $y_{ir}(\omega)$. Therefore, expanding and rearranging the quadratic terms in (4.15), the objective function of the relaxed problem can be rewritten as:

$$\min \sum_{\omega\in\Omega} \pi_\omega \Bigg[\sum_{r=1}^{n_R} (c^f + \lambda^X_{r\omega} - \rho x_r + \frac{\rho}{2}) x_r(\omega) + \sum_{i=1}^{n}\sum_{r=1}^{n_R} (\lambda^Y_{ir\omega} - \rho y_{ir} + \frac{\rho}{2}) y_{ir}(\omega)$$
$$+ \sum_{r=1}^{n_R} c^o O_r(\omega) - \sum_{r=1}^{n_R} (\lambda^X_{r\omega} - \frac{\rho}{2}) x_r - \sum_{i=1}^{n}\sum_{r=1}^{n_R} (\lambda^Y_{ir\omega} - \frac{\rho}{2}) y_{ir} \Bigg]. \tag{4.16}$$

For a given first-stage solution $(\hat{x}, \hat{y})$, the objective function (4.16), and hence the relaxed problem becomes scenario-separable. Given $(\hat{x}, \hat{y})$, the scenario subproblem under scenario ω is:

$$\min \sum_{r=1}^{n_R} (c^f + \lambda^X_{r\omega} - \rho \hat{x}_r + \frac{\rho}{2}) x_r(\omega) + \sum_{i=1}^{n}\sum_{r=1}^{n_R} (\lambda^Y_{ir\omega} - \rho \hat{y}_{ir} + \frac{\rho}{2}) y_{ir}(\omega)$$
$$+ \sum_{r=1}^{n_R} c^o O_r(\omega) - \sum_{r=1}^{n_R} (\lambda^X_{r\omega} - \frac{\rho}{2}) \hat{x}_r - \sum_{i=1}^{n}\sum_{r=1}^{n_R} (\lambda^Y_{ir\omega} - \frac{\rho}{2}) \hat{y}_{ir} \tag{4.17a}$$

s.t.

$$y_{ir}(\omega) \le x_r(\omega) \qquad \forall i, r, \tag{4.17b}$$

$$\sum_{r=1}^{n_R} y_{ir}(\omega) = 1 \qquad \forall i, \tag{4.17c}$$

$$x_r(\omega) \ge x_{r+1}(\omega) \qquad \forall r < n_R, \tag{4.17d}$$

$$\sum_{r=1}^{i} y_{ir}(\omega) = 1 \qquad \forall i \le min\{n, n_R\}, \tag{4.17e}$$

$$\sum_{q=r}^{min\{i,n_R\}} y_{iq}(\omega) \le \sum_{j=r-1}^{i-1} y_{j,r-1}(\omega) \qquad \forall (i, r) : i \ge r > 1, \tag{4.17f}$$

$$C_{ir}(\omega) \le M y_{ir}(\omega) \qquad \forall i, r, \tag{4.17g}$$

$$C_{ir}(\omega) \geq \sum_{j=1}^{n} b_{ji} p_j(\omega) y_{jr}(\omega) + p_i(\omega) y_{ir}(\omega) - M(1 - y_{ir}(\omega)) \qquad \forall i, r, \tag{4.17h}$$

$$C_{ir}(\omega) \leq \sum_{j=1}^{n} b_{ji} p_j(\omega) y_{jr}(\omega) + p_i(\omega) y_{ir}(\omega) \qquad \forall i, r, \tag{4.17i}$$

$$T_L - \left[\sum_{r=1}^{n_R} C_{ir}(\omega) - p_i(\omega) \right] \leq T_L \sum_{r=1}^{n_R} y_{ir}^M(\omega) \qquad \forall i, \tag{4.17j}$$

$$\left[\sum_{r=1}^{n_R} C_{ir}(\omega) - p_i(\omega) \right] - T_L \leq M \left[1 - \sum_{r=1}^{n_R} y_{ir}^M(\omega) \right] \qquad \forall i, \tag{4.17k}$$

$$y_{ir}^M(\omega) \leq y_{ir}(\omega) \qquad \forall i, r, \tag{4.17l}$$

$$\sum_{r=1}^{n_R} y_{ir}^M(\omega) + \sum_{r=1}^{n_R} y_{ir}^A(\omega) = 1 \qquad \forall i, \tag{4.17m}$$

$$y_{ir}^A(\omega) \leq x_r(\omega) \qquad \forall i, r, \tag{4.17n}$$

$$O_r(\omega) \geq \sum_{i=1}^{n} p_i(\omega) y_{ir}^M(\omega) + \sum_{i=1}^{n} p_i(\omega) y_{ir}^A(\omega) - L \qquad \forall r, \tag{4.17o}$$

$$O_r(\omega) \geq T_E + \sum_{i=1}^{n} p_i(\omega) y_{ir}^A(\omega) - L \qquad \forall r, \tag{4.17p}$$

$$x_r(\omega), y_{ir}(\omega), y_{ir}^M(\omega), y_{ir}^A(\omega) \in \{0, 1\} \qquad \forall i, r, \tag{4.17q}$$

$$C_{ir}(\omega), O_r(\omega) \geq 0 \qquad \forall i, r. \tag{4.17r}$$

At each iteration of the PHA [66], scenario subproblems are solved, and an *aggregate solution* $(\hat{x}, \hat{y})$ is generated by using the optimal solutions of scenario subproblems. The aggregate solution $(\hat{x}, \hat{y})$ is then used to construct the scenario subproblems at the succeeding iteration. Based on the deviation of scenario subproblem solutions from the aggregate solution, the Lagrangian multipliers and the penalty parameter are updated at each iteration to enforce nonanticipativity in a progressive manner. A stepwise description of the PHA for 2S-ORRP is presented below.

- **Step 1.**
  - **1a.** Solve the scenario subproblem (4.17) for each ω by changing the objective function (4.17a) as:

$$\min \sum_{r=1}^{n_R} c^f x_r(\omega) + \sum_{r=1}^{n_R} c^o O_r(\omega). \tag{4.18}$$

- **1b.** Let $(x^*(\omega), y^*(\omega))$ be the optimal solution under scenario ω, and initialize the aggregate solution $(\hat{x}, \hat{y})$ as:

$$\hat{x}_r = \sum_{\omega \in \Omega} \pi_\omega x_r^*(\omega) \qquad \forall r, \tag{4.19}$$

$$\hat{y}_{ir} = \sum_{\omega \in \Omega} \pi_\omega y_{ir}^*(\omega) \qquad \forall i, r. \tag{4.20}$$

- **1c.** If $(x^*(\omega), y^*(\omega)) = (\hat{x}, \hat{y})$ for all ω (i.e., the nonanticipativity constraints are satisfied), $(\hat{x}, \hat{y})$ is an optimal solution of 2S-ORRP. Otherwise, go to Step 2 after initializing the multipliers and the penalty parameter as:

$$\lambda_{r\omega}^X = 0 \qquad \forall r, \omega, \tag{4.21}$$

$$\lambda_{ir\omega}^Y = 0 \qquad \forall i, r, \omega, \tag{4.22}$$

$$\rho = \rho_0, \tag{4.23}$$

where ρ_0 is a constant value such that $\rho_0 > 0$.

- **Step 2.**
  - **2a.** Considering the aggregate solution $(\hat{x}, \hat{y})$, and the current values of $\lambda_{r\omega}^X, \lambda_{ir\omega}^Y$, and ρ, solve the scenario subproblem (4.17) for each ω.
  - **2b.** Let $(x^*(\omega), y^*(\omega))$ be the optimal solution under scenario ω, and update the aggregate solution $(\hat{x}, \hat{y})$ by using (4.19) and (4.20).
  - **2c.** If $(x^*(\omega), y^*(\omega)) = (\hat{x}, \hat{y})$ for all ω, $(\hat{x}, \hat{y})$ is the best solution obtained by the PHA. Otherwise, go to Step 2a after updating the multipliers and the penalty parameter as:

$$\lambda_{r\omega}^X = \lambda_{r\omega}^X + \rho(x_r^*(\omega) - \hat{x}_r) \qquad \forall r, \omega, \tag{4.24}$$

$$\lambda_{ir\omega}^Y = \lambda_{ir\omega}^Y + \rho(y_{ir}^*(\omega) - \hat{y}_{ir}) \qquad \forall i, r, \omega, \tag{4.25}$$

$$\rho = \alpha\rho, \tag{4.26}$$

where α is a constant value such that $\alpha \geq 1$.

Convergence of the PHA to the optimal solution is not guaranteed for the problems with integer variables. Therefore, we use the algorithm as a heuristic method to solve 2S-ORRP (i.e., to get an upper bound on the optimal objective function value of 2S-ORRP).

Note that the aggregate solution $(\hat{x}, \hat{y})$ is not necessarily feasible at every iteration. Obtaining an upper bound on the optimal objective function value by constructing a feasible solution from $(\hat{x}, \hat{y})$ at each iteration enhances the algorithm [15]. We use the following rounding procedure to obtain a feasible solution from $(\hat{x}, \hat{y})$:

- Let $(\hat{x}^F, \hat{y}^F)$ denote the feasible solution to be constructed from $(\hat{x}, \hat{y})$. Initialize every component of $(\hat{x}^F, \hat{y}^F)$ to 0.
- For every surgery i: Let r' be $\min_r\{\arg\max_r(\hat{y}_{ir})\}$. Update $\hat{y}^F_{ir'}$ as $\hat{y}^F_{ir'} = 1$.
- For every OR r: If $\sum_{i=1}^{n} \hat{y}^F_{ir} > 0$, then update $\hat{x}^F_r$ as $\hat{x}^F_r = 1$.

At each iteration, we update the best available solution if $(\hat{x}^F, \hat{y}^F)$ provides a lower objective function value.

4.5 COMPUTATIONAL RESULTS

4.5.1 Generation of Problem Instances

The problem instances for our computational study in this chapter are generated based on the data set and parameter estimation described in Section 3.4.1. As the focus of 2S-ORRP is ORs, we do not use surgeon-related parameters (such as the surgeon idle time cost, surgeon turnover time or the surgery sequence based on surgeons' surgery listings). Moreover, 2S-ORRP does not explicitly include the OR turnover time. Instead, when generating the random scenarios, we increase the total duration (the sum of pre-incision, incision, and post-incision durations) of each surgery by 30 minutes to account for the OR turnover time. As in Chapter 3, we consider L=9 hours/day, and a cost structure where the overtime cost is 50% higher than the regular OR time cost (i.e., 6 hours of overtime is equivalent in cost to opening a new OR). In Section 4.5.4, we investigate the impact of different overtime cost levels.

Table 4.2: Size-based classification of 2S-ORRP instances for 322 surgical days.

Set No	Number of Instances	Average				Maximum			
		n	n_R	n^1_{Bin}	n^2_{Bin}	n	n_R	n^1_{Bin}	n^2_{Bin}
1	116	3.23	2.26	9.68	7422.41	6	3	15	12000
2	119	6.01	3.60	25.19	21596.64	9	4	32	28000
3	87	8.64	4.70	45.46	40758.62	12	7	77	70000

We consider a setting where T_R=4 hours. Since the surgery durations become known once the day starts (i.e., 2S-ORRP is a two-stage problem), there is no incentive in using a smaller T_L value or a greater T_E value than T_R. Therefore, we set $T_L = T_R$ and $T_E = T_R$.

We assume that surgeries within each OR are performed in the order of decreasing expected processing time (i.e., longest expected processing time (LEPT) first), and we set the values of b_{ij} parameters accordingly. In Section 4.5.5, we explore the impact of using other sequencing rules including shortest expected processing time (SEPT) first, smallest variance (SVar) first, largest variance (LVar) first, smallest coefficient of variation (SCoV) first, and largest coefficient of variation (LCoV) first.

The number of binary variables in an instance of 2S-ORRP, and hence the problem size, depends on the number of surgeries, available ORs, and the total number of scenarios (which is 500 in our implementation). We classify our 322 problem instances, each of which corresponds to a surgical day, into 3 sets based on their size. Table 4.2 summarizes the number of instances, average and maximum number of surgeries (n), available ORs (n_R), and first- and second-stage binary variables (n^1_{Bin} and n^2_{Bin}) for each set.

4.5.2 Computational Performance of the Proposed Algorithms

We coded our algorithms in Microsoft Visual Studio 2010 using CPLEX 12 callable library, and we executed our experiments on Intel Core2 Duo PC with processors running at 3.17 GHz and 2 GB memory under Windows 7. We impose a 3-hour time limit on both algorithms.

Table 4.3: Solution times (in CPU seconds) and the percentage optimality gap values for the integer L-shaped algorithm.

Set No.	Solution Time		Number of Unsolved Instances	Percentage Optimality Gap for the Unsolved Instances (Based on the Lower Bound of the Integer L-Shaped Algorithm)		Percentage Optimality Gap for the Unsolved Instances (Based on the Best Available Lower Bound)	
	Average	Maximum		Average	Maximum	Average	Maximum
1	10.32	87.42	-	-	-	-	-
2	1114.11	>10800.00	1	0.80%	0.80%	0.37%	0.37%
3	8631.26	>10800.00	58	30.71%	210.61%	0.83%	14.41%

We report the performance measures associated with the integer L-shaped algorithm including the average and maximum solution times, number of unsolved instances, and the percentage optimality gap for the unsolved instances in Table 4.3. For the unsolved instances, we consider the solution time as 3 hours when calculating the average solution time. Although the algorithm performs well for the small and moderate-sized instances, it cannot solve most of the large instances (i.e., instances in Set 3) within the allowed time limit. The average percentage gap for the unsolved 58 instances in Set 3 is 30.71%, and the maximum gap is 210.61%. Besides the percentage optimality gap values based on the lower bound returned by the integer L-shaped algorithm, we also report the gap values based on the best available lower bound, which is the maximum of the integer L-shaped lower bound and the expected value of the wait-and-see solution. The percentage optimality gap with respect to the best available lower bound for the unsolved instances is 0.83% on average and as much as 14.41%, which indicates that the performance of the integer L-shaped algorithm in terms of solution quality is remarkably well on average, but not for every instance.

Because the percentage optimality gap achieved by the integer L-shaped algorithm is not acceptably low for every instance, we employ the PH-based heuristic described in Section 4.4.2 to obtain solutions for the instances in Set 3. Our initial experiments indicate that using a smaller penalty coefficient earlier in the algorithm, and increasing it throughout the

Table 4.4: Solution times (in CPU seconds) and the objective values of the best solutions returned by the PH-based heuristic under different parameter settings for a problem instance whose optimal objective function value is 16993.34.

ρ_0	α	Solution Time	Objective Value
5000	1	32.34	17495.73
2500	1	>10800.00	16993.34
1000	1	>10800.00	16993.46
2	1	>10800.00	16993.46
2	5	1306.95	16995.78
2	10	164.03	16995.78

iterations facilitates obtaining a good solution in a reasonable amount of time (see Table 4.4). Considering this, we set the parameters in the PH-based heuristic as $\rho_0 = 2$ and $\alpha = 10$. We report the solution time- and solution quality-related measures for the algorithm in Table 4.5. To assess the quality of the solution, we use the average gap between the objective values of the best solutions (i.e., upper bounds) obtained by the PH-based heuristic and the integer L-shaped algorithm. For 3 of the instances (out of 87 in Set 3), the PH-based heuristic terminated because of the computational time limit. The maximum solution time is reported accordingly. The minimum percentage gap is -9.31%, which indicates that there are problem instances for which the upper bound provided by the PH-based heuristic is significantly better than that of the integer L-shaped algorithm. The average gap is positive (0.09%), yet very small, which makes the PH-based heuristic an appealing method for the large instances when its superiority in terms of the solution time is also taken into account.

To have a better understanding of the PH-based heuristic solution quality, we also report the percentage gap values by classifying the instances in Set 3 to Sets 3a and 3b. Our classification is based on whether or not an instance can be solved optimally by the integer L-shaped algorithm within the allowed time limit. Set 3a includes the 29 instances for which the optimal solution is available, and Set 3b includes the remaining 58 instances for which

Table 4.5: Solution times (in CPU seconds) and the percentage gap values for the PH-based heuristic.

Set No.	Solution Time		Percentage Gap between the Upper Bounds of the Integer L-Shaped Algorithm and the PH-Based Heuristic		
	Average	Maximum	Minimum	Average	Maximum
3	1245.64	>10800.00	-9.31%	0.09%	1.43%
3a	-		0.00%	0.14%	1.43%
3b	-		-9.31%	0.07%	1.38%

only an upper bound is available. The average gap across the instances in Set 3a, 0.14%, reveals that the PH-based heuristic solution performs slightly worse than the optimal solution on average. The average gap for Set 3b, 0.07%, indicates that the upper bounds obtained by the two algorithms are very close to each other.

4.5.3 Value of Rescheduling

Let z_{EEV} and z_{WS} be the expected value of the mean value problem solution and the wait-and-see solution for 2S-ORSP. Moreover, let z_{ORSP} and z_{ORRP} be the optimal objective function value of 2S-ORSP and 2S-ORRP, respectively. The following relation exists between these values:

$$z_{EEV} \geq z_{ORSP} \geq z_{ORRP} \geq z_{WS}. \tag{4.27}$$

Capturing the uncertainty in surgery durations reduces the minimum total expected cost from z_{EEV} to z_{ORSP}, and the difference between these two values corresponds to the VSS for 2S-ORSP. The expected cost could be reduced further, from z_{ORSP} to z_{ORRP}, by rescheduling. We refer to the difference between z_{ORSP} to z_{ORRP} as the value of rescheduling. If perfect information about surgery durations were available when the initial scheduling decisions are made, then the expected cost would be z_{WS}. Therefore, z_{WS} is a lower bound on both z_{ORSP} and z_{ORRP}, and the difference between z_{ORSP} and z_{WS} (i.e., EVPI for 2S-ORSP) is an upper bound on the value of rescheduling. A high EVPI for 2S-ORSP indicates that rescheduling, which is introduced in 2S-ORRP, might achieve significant cost reductions.

It is possible to decompose the value of rescheduling into two components: the value of adaptive decisions and the value of anticipative decisions. Let (x_{ORSP}, y_{ORSP}) be the optimal solution of 2S-ORSP, and z'_{ORRP} be the optimal objective function value of 2S-ORRP given that the first-stage solution is (x_{ORSP}, y_{ORSP}). Then, we have:

$$z_{ORSP} \geq z'_{ORRP} \geq z_{ORRP}. \tag{4.28}$$

The expected cost can be reduced from z_{ORSP} to z'_{ORRP} by making adaptive decisions. A further reduction from z'_{ORRP} to z_{ORRP} can be achieved through the integration of adaptive and anticipative decisions (i.e., by using 2S-ORRP). Therefore, we refer to $z_{ORSP} - z'_{ORRP}$ and $z'_{ORRP} - z_{ORRP}$ as the value of adaptive decisions and the value of anticipative decisions, respectively.

We summarize the percentage VSS and EVPI for 2S-ORSP, the percentage value of rescheduling, and the percentage value of adaptive and anticipative decisions for our problem instances in Table 4.6. For the instances whose 2S-ORRP formulations cannot be solved within the allowed time limit, we consider the best available upper bound (which is the minimum of the upper bounds returned by the integer L-shaped algorithm and the PH-based heuristic, and z'_{ORRP}) as z_{ORRP} in our comparisons to obtain the lower bounds on the value of rescheduling and the value of anticipative decisions. To estimate the upper bounds on these values, we use the best available lower bound (which is the maximum of z_{WS} and the lower bound returned by the integer L-shaped algorithm) as z_{ORRP} in our comparisons.

Both VSS (ranging between 0.03%-0.37%) and EVPI (ranging between 0.57%-1.61%) are considerably small for our problem instances (particularly for the smaller instances) on average. As expected based on the EVPI results, we also observe that the value of rescheduling (ranging between 0.20%-1.27%) is not high on average. However, the average value of rescheduling compared to the average VSS is significant. The value of adaptive decisions is notably higher than that of anticipative decisions, which indicates that a large portion of

Table 4.6: Percentage value of modeling uncertainty and rescheduling decisions.

Set No.	VSS		EVPI		Value of Rescheduling		Value of Adaptive Decisions		Value of Anticipative Decisions	
	Average	Maximum	Average	Maximum	Average	Maximum	Average	Maximum	Average	Maximum
1	0.03%	1.01%	0.57%	3.80%	0.20%	3.26%	0.18%	3.26%	0.03%	0.76%
2	0.25%	3.32%	1.21%	4.05%	0.64%	2.97%	0.53%	2.97%	0.11%	1.65%
3	0.37%	2.28%	1.61%	4.67%	[0.89%,1.27%]	[3.01%,4.67%]	0.78%	3.01%	[0.11%,0.50%]	[1.36%,3.26%]

the value of rescheduling is attributable to taking reactive actions rather than designing the initial schedule by taking the availability of the rescheduling opportunity into account.

4.5.4 Impact of the Overtime Cost Level

The overtime cost, c^o, highly depends on the surgical environment, and it is typically higher for an outpatient setting. In order to explore the impact of c^o on the value of rescheduling, we solve the problem instances in Set 2 under the following additional cost structures [20]:

- Using 2 hours of overtime is equivalent to opening a new OR.
- Using 0.5 hours of overtime is equivalent to opening a new OR.

We refer to our original parameter setting, and these two additional settings as the low, medium, and high overtime cost levels, respectively. We report the uncertainty- and rescheduling-related performance measures at these overtime cost levels in Table 4.7. The value of rescheduling becomes higher as the overtime cost increases. For the high overtime cost level, it is 4.15% on average and as much as 19.86%. Based on the results summarized in Table 4.7, we conclude that rescheduling is particularly important in outpatient surgical environments.

4.5.5 Impact of the Surgery Sequence

Our computational results presented in the earlier sections are based on the experiments that employ the LEPT first rule to sequence the surgeries within each OR. In order to investigate the impact of the surgery sequence on the value of rescheduling, we solve the problem instances in Set 2 under the SEPT, SVar, LVar, SCoV, and LCoV first sequencing rules. For these different settings, we report the performance measures associated with the rescheduling decisions in Table 4.8. As can be observed from Table 4.8, the value of rescheduling highly depends on the surgery sequence. The highest (lowest) average value is attained under the LEPT (SEPT) first sequencing rule, which is an intuitive result as performing longer thus fewer surgeries earlier in the day leaves more surgeries to be rescheduled at the time of

Table 4.7: Percentage value of modeling uncertainty and rescheduling decisions at different overtime cost levels for the problem instances in Set 2.

Overtime Cost Level	VSS		EVPI		Value of Rescheduling		Value of Adaptive Decisions		Value of Anticipative Decisions	
	Average	Maximum	Average	Maximum	Average	Maximum	Average	Maximum	Average	Maximum
Low	0.25%	3.32%	1.21%	4.05%	0.64%	2.97%	0.53%	2.97%	0.11%	1.65%
Medium	0.60%	3.88%	3.70%	12.24%	2.17%	8.15%	1.76%	8.15%	0.41%	4.85%
High	3.14%	22.26%	7.11%	26.78%	4.15%	19.86%	3.39%	19.86%	0.80%	5.78%

Table 4.8: Percentage value of rescheduling decisions under different sequencing rules for the problem instances in Set 2.

Sequencing Rule	Value of Rescheduling		Value of Adaptive Decisions		Value of Anticipative Decisions	
	Average	Maximum	Average	Maximum	Average	Maximum
SEPT	0.35%	2.49%	0.20%	1.69%	0.16%	1.51%
LEPT	0.64%	2.97%	0.53%	2.97%	0.11%	1.65%
SVar	0.43%	2.52%	0.24%	2.30%	0.19%	1.52%
LVar	0.57%	2.97%	0.43%	2.97%	0.14%	2.06%
SCoV	0.53%	2.75%	0.33%	2.30%	0.20%	2.48%
LCoV	0.54%	2.97%	0.37%	2.97%	0.17%	2.06%

rescheduling. LVar first rule achieves a higher value of rescheduling than SVar first does, whereas using SCoV first and LCoV first yields very similar results.

4.6 CONCLUSIONS

We consider the stochastic multi-OR surgery scheduling problem where the allocation of surgeries to ORs is revised at a predetermined rescheduling point during the surgical day. We formulate the problem as a three-stage SMIP that treats the initial scheduling and the day of surgery rescheduling decisions as the first- and second-stage decisions, respectively. The objective of the model is to minimize the sum of the fixed cost of opening ORs incurred in the first-stage and the expected cost of overtime incurred in the third-stage. Due to the difficulty in accurately modeling the resolution of uncertainty in surgery durations in a three-stage setting and efficiently solving the resulting three-stage model, we consider a special case where the rescheduling decisions in the second-stage are made under perfect information and hence the problem can be formulated as a two-stage SMIP.

We employ the integer L-shaped algorithm, which is a stage-wise decomposition method, to solve small and moderate-sized instances of our problem. Although the algorithm performs well for those instances, it fails to solve most of the large instances within a reasonable amount of time. Therefore, we use a PH-based heuristic, which is a scenario-wise decomposition method, to find a near optimal solution for the large instances. Our results indicate that the performance of the PH-based heuristic in terms of solution quality (i.e., the optimality gap) is slightly worse than that of the integer L-shaped algorithm. However, it is significantly superior when the comparison criterion is solution time.

By using our model, we estimate the value of rescheduling, and the value of adaptive and anticipative decisions. Our numerical results indicate that rescheduling can bring notable cost reductions when VSS is high, which typically is the case under high overtime cost levels (i.e., in outpatient surgical environments). Although the value of rescheduling for the moderate-sized instances is 0.64% on average with a maximum of 2.97% when the overtime cost level is low, its average and maximum value is 4.15% and 19.86% under the high overtime cost level. Our results reveal that a significant portion of the value of rescheduling is attributable to the adaptive decisions rather than the anticipative decisions. We also observe that the value of rescheduling depends on the surgery sequence within ORs, and the LEPT sequence (among several sequencing rules considered) yields the best results.

5.0 CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

This dissertation focuses on developing realistic models of the surgery scheduling process common to many health care providers and efficiently solving those models. The majority of earlier studies consider either deterministic multi-OR or stochastic single-OR environments. However, both uncertainty and the existence of multiple ORs in surgical suites are important aspects of OR scheduling problems in practice. Recognizing this fact, we develop novel stochastic programming models for multi-OR scheduling problems where surgery durations are uncertain, increase the solvability of our models by exploiting their structural properties, and draw valuable managerial insights from our extensive computational study.

5.1 SUMMARY AND CONCLUSIONS

In Chapter 3, we formulated the multi-OR scheduling problem as a two-stage SMIP where the scheduling decisions (the number of ORs to open, surgery allocation and sequencing, and start times for surgeons) are made in the first stage, before the resolution of uncertainty. We treated both ORs and surgeons as resources, and explicitly considered the OR and surgeon turnover times, and pre-incision, incision, and post-incision phases of surgeries. The objective of our model is to minimize the total expected operating cost, which is the sum of the fixed cost opening ORs, the expected overtime cost and the expected surgeon idling cost. By using our model, we estimated the value of capturing uncertainty (i.e., VSS) and the value of using ORs as a common shared resource (as opposed to assigning them to surgeons), and we illustrated the impact of parallel surgery processing. We used the L-shaped algorithm and a L-shaped based branch-and-cut algorithm to solve the problem. To be able to solve practical instances, we significantly improved these methods by introducing a set of new and widely applicable valid inequalities based on Jensen's inequality. Our numerical results showed that capturing uncertainty is valuable particularly when the per unit time cost of surgeon idling is high. Our results also revealed that significant cost reductions can be achieved by OR pooling, and the benefits of parallel surgery processing increases with the OR turnover time and the parallelizable portion of surgeries.

In Chapter 4, we studied the multi-OR scheduling problem with day of surgery rescheduling decisions made to better respond the resolved uncertainty in surgery durations. We formulated the problem as a three-stage SMIP that minimizes the total expected operating cost, and then considered a special case, which is a two-stage SMIP, where the uncertainty is completely resolved before the rescheduling decisions are made. Our rescheduling model considers only ORs as resources, and hence does not include any surgeon-related decisions, constraints, or performance measures. The initial scheduling decisions (i.e., the first-stage decisions) include the number of ORs to open and the allocation of surgeries to ORs, and the rescheduling decisions (i.e., the second-stage decisions) are the reallocation of surgeries to ORs. We used the integer L-shaped algorithm to solve small and moderate-sized instances of our problem, and a PH-based heuristic to find a good upper bound for the large instances. We concluded from our numerical results that the value of rescheduling during the surgical day is high when the value of capturing uncertainty is significant, which is typically observed when per unit time cost of overtime is high. Moreover, we observed from our results that a large portion of this value is due to adaptive decisions, which indicates that rescheduling would reduce the operating costs significantly even if the initial scheduling decisions are made without taking the availability of such recourse decisions into account. Our results also revealed that the benefit brought by rescheduling is higher when surgeries with longer expected durations are performed earlier in the day.

5.2 FUTURE RESEARCH DIRECTIONS

The models and the solution approaches discussed in this dissertation can be extended in several ways to investigate important research questions some of which are presented below.

- **Value of rescheduling in a multi-stage setting:** By using our two-stage stochastic multi-OR rescheduling model (2S-ORRP), we estimated the value of rescheduling under perfect information, which is an upper bound on the value of rescheduling as the uncertainty in surgery durations resolves in multiple stages in practice. As a result, evaluating the total expected cost of the 2S-ORRP solution in a multi-stage setting is of great practical value. One possible way of doing this is to solve the two stage SMIP comprised of the second and third stages of 3S-ORRP for the first-stage solution given by 2S-ORRP. Instead of evaluating the 2S-ORRP solution in a three-stage setting, solving 3S-ORRP would naturally give a better estimate for the value of rescheduling. Therefore, designing efficient solution methods for 3S-ORRP is an important research direction. Considering a multi-stage setting raises another important question that could be addressed: the determination of the rescheduling point. Rescheduling decisions should be made early enough that there would be a reasonable number of remaining surgeries to be rescheduled, and late enough that a considerable portion of uncertainty is resolved before the recourse decisions are made.
- **Impact of rescheduling on the stability of the system:** While additional overtime costs could be avoided by rescheduling surgeries during the day, excessive levels of rescheduling may be undesirable as it would cause significant deviations from the initial schedule. The trade-off between the operating costs and the stability of the system can be analyzed by considering a penalty cost associated with the rescheduled surgeries.
- **Rescheduling model in the presence of other resources and decisions:** An extended rescheduling model can be developed by introducing rescheduling decisions in the multi-OR scheduling problem described in Chapter 3. This corresponds to including sequencing decisions, and surgeon-related constraints, variables, and performance measures in the multi-OR rescheduling problem described in Chapter 4. The contribution of

such an extension would be twofold. First, it considers the surgeons' preferences on the sequence of their own surgeries. Second, the best sequence within each OR is determined by the model, which might increase the potential benefit of rescheduling as our computational results in Section 4.5.5 indicate that the surgery sequencing rule has a notable impact on the value of rescheduling.

- **Other types of uncertainties:** The models presented in this study consider only the uncertainty in surgery durations. Depending on the characteristics of the considered surgical ward and the health care institution, other types of uncertainties such as the arrival of add-on surgeries or patient no-shows can also be included in the models.
- **Investigating the underlying multi-criteria optimization problems:** We estimate the value of different resource usage schemes such as parallel surgery processing and OR pooling, and the value of rescheduling based on the total expected operating cost. Therefore, our analysis depends on the specific cost coefficients that weight the multiple criteria in the objective functions of our models. We leave an extensive explicit treatment of these multi-criteria optimization problems for future research.
- **Developing new valid inequalities based on the stage-wise structure of the stochastic programs:** The Jensen's based valid inequalities introduced in Chapter 3 are applicable to two-stage stochastic programs where the expected recourse is a convex function of the random parameters. Developing similar valid inequalities for general two-stage stochastic programs would increase the efficiency of standard solution techniques for a broader class of problems (including the 2S-ORRP, whose recourse function is not convex in ξ).

BIBLIOGRAPHY

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

[2] S. Ahmed, M. Tawarmalani, and N. V. Sahinidis. A finite branch and bound algorithm for two-stage stochastic integer programs. *Mathematical Programming*, 100(2):355–377, 2004.

[3] J.E. Bardram and T.R. Hansen. Why the plan doesn't hold: A study of situated planning, articulation and coordination work in a surgical ward. *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work*, pages 331–340, 2010.

[4] S. Batun, B.T. Denton, T.R. Huschka, and A.J. Schaefer. Operating room pooling and parallel surgery processing under uncertainty. *INFORMS Journal on Computing*, 23(2):220–237, 2011.

[5] E.M.L. Beale. On minimizing a convex function subject to linear inequalities. *Journal of the Royal Statistical Society, Series B (Methodological)*, 17(2):173–184, 1955.

[6] M.A. Begen and M. Queyranne. Appointment scheduling with discrete random durations. *Mathematics of Operations Research.*, 36(2):240–257, 2011.

[7] J. Beliën and E. Demeulemeester. Building cyclic master surgery schedules with leveled resulting bed occupancy. *European Journal of Operational Research*, 176(2):1185–1204, 2007.

[8] J.F. Benders. Partitioning procedures for solving mixed variables programming problems. *Numerische Mathematics*, 4(1):238–252, 1962.

[9] J.R. Birge. Decomposition and partitioning methods for multistage stochastic linear programs. *Operations Research*, 33(5):989–1007, 1985.

[10] J.R. Birge and F. Louveaux. *Introduction to Stochastic Programming.* Springer, New York, NY, 1997.

[11] J.T. Blake and M.W. Carter. Surgical process scheduling: A structured review. *Journal of the Society for Health Systems*, 5(3):17–30, 1997.

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

[13] C.C. Carøe and R. Schultz. Dual decomposition in stochastic integer programming. *Operations Research Letters*, 24(1-2):37–45, 1999.

[14] Centers for Medicare and Medicaid Services CMS. National Health Expenditure Data, 2009. Available from http://www.cms.hhs.gov/NationalHealthExpendData/downloads/tables.pdf, information and data accessed on January 28, 2011.

[15] T.G. Crainic, X. Fu, M. Gendreau, W. Rei, and S.W. Wallace. Progressive hedging-based meta-heuristics of stochastic network design. *Networks*, 58(2):114–124, 2011.

[16] G.B. Dantzig. Linear programming under uncertainty. *Management Science*, 1(3-4):197–206, 1955.

[17] B.T. Denton and D. Gupta. Sequential bounding approach for optimal appointment scheduling. *IIE Transactions*, 35(11):1003–1016, 2003.

[18] B.T. Denton, A.S. Rahman, H. Nelson, and A.C. Bailey. Simulation of a multiple operating room surgical suite. *Proceedings of the 2006 Winter Simulation Conference*, pages 414–424, 2006. L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, eds.

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

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

[21] F. Dexter. A strategy to decide whether to move the last case of the day in an operating room to another empty operating room to decrease overtime labor costs. *Anesthesia & Analgesia*, 91(4):925–928, 2000.

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

[23] F. Dexter, A. Macario, F. Qian, and R.D. Traub. 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, 1999.

[24] F. Dexter, A. Macario, R.D. Traub, M. Hopwood, and D.A. Lubarsky. 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, 1999.

[25] F. Dexter, R.D. Traub, and A. Macario. How to release allocated operating room time to increase efficiency: Predicting which surgical service will have the most underutilized operating room time. *Anesthesia & Analgesia*, 96(2):507–512, 2003.

[26] F. Dexter, R.H. Epstein, R.D. Traub, and Y. Xiao. Making management decisions on the day of surgery based on operating room efficiency and patient waiting times. *Anesthesiology*, 101 (6):1444–1453, 2004.

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

[28] A. Elixhauser and R.M. Andrews. Profile of inpatient operating room procedures in US hospitals in 2007. *Archives of Surgery*, 145(12):1201–1208, 2010.

[29] S.A. Erdogan and B.T. Denton. Surgery planning and scheduling. *Wiley Encyclopedia of Operations Research and Management Science*, 2011. J. J. Cochran, L. A. Cox, P. Keskinocak, J. Kharoufeh, and J. C. Smith, eds.

[30] S.A. Erdogan and B.T. Denton. Dynamic appointment scheduling with uncertain demand. *INFORMS Journal on Computing.*, 2011. In press.

[31] J.E. Everett. A decision support simulation model for the management of an elective surgery waiting system. *Health Care Management Science*, 5(2):89–95, 2002.

[32] H. Fei, C. Chu, N. Meskens, and A. Artiba. Solving surgical cases assignment problem by a branch-and-price approach. *International Journal of Production Economics*, 112(1):96–108, 2008.

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

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

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

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

[37] Y. Guan, S. Ahmed, and G.L. Nemhauser. Cutting planes for multistage stochastic integer programs. *Operations Research*, 57(2):287–298, 2009.

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

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

[40] D. Gupta. Surgical suites' operations management. *Production and Operations Management*, 16(6):689–700, 2007.

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

[42] R. Hassin and S. Mendel. Scheduling arrivals to queues: A single-server model with no-shows. *Management Science*, 54(3):565–572, 2008.

[43] R. Hemmecke and R. Schultz. Decomposition of test sets in stochastic integer programming. *Mathematical Programming*, 94(2):323–341, 2003.

[44] Healthcare Financial Management Association HFMA. Achieving operating room effciency through process integration. *Healthcare Financial Management: Journal of the Healthcare Financial Management Association*, 57(3):1–8, 2003.

[45] J. L. Higle and S. Sen. Stochastic decomposition: An algorithm for two-stage linear programs with recourse. *Mathematics of Operations Research*, 16(3):650–669, 1991.

[46] T.R. Huschka, B.T. Denton, S. Gul, and J.W. Fowler. Bi-criteria evaluation of an outpatient procedure center via simulation. *Proceedings of the 2007 Winter Simulation Conference*, pages 1510–1518, 2007. S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds.

[47] A. Jebali, A.B.H. Alouane, and P. Ladet. Operating rooms scheduling. *International Journal of Production Economics*, 99(1/2):52–62, 2006.

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

[49] P. Kall and S.W. Wallace. *Stochastic Programming*. Wiley, Chichester, 1994.

[50] K.C. Kiwiel. Proximity control in bundle methods for convex nondifferentiable minimization. *Mathematical Programming*, 46(1):105–122, 1990.

[51] N. Kong, A.J. Schaefer, and B. Hunsaker. Two-stage integer programs with stochastic right-hand sides: A superadditive dual approach. *Mathematical Programming*, 108(2):275–296, 2006.

[52] G. Laporte and F.V. Louveaux. The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, 13(3):133–142, 1993.

[53] G. Laporte, F.V. Louveaux, and L. Van Hamme. Exact solution to a location problem with stochastic demands. *Transportation Science*, 28(2):95–103, 1994.

[54] Y. Liu, C. Chu, and K. Wang. A new heuristic algorithm for the operating room scheduling problem. *Computers & Industrial Engineering.*, 61(3):865–871, 2011.

[55] F.V. Louveaux. A solution method for multistage stochastic programs with recourse with application to an energy investment problem. *Operations Research*, 28(4):889–902, 1980.

[56] G. Lulli and S. Sen. A branch-and-price algorithm for multistage stochastic integer programming with application to stochastic batch-sizing problems. *Management Science*, 50(6): 786–796, 2004.

[57] A. Madansky. Inequalities for stochastic linear programming problems. *Management Science*, 6(2):197–204, 1960.

[58] F. Margot. Pruning by isomorphism in branch-and-cut. *Mathematical Programming*, 94(1): 71–90, 2002.

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

[60] J.H. May, W.E. Spangler, D.P. Strum, and L.G. Vargas. The surgical scheduling problem: Current research and future opportunities. *Production and Operations Management*, 20(3): 392–405, 2011.

[61] National Academy of Engineering and Institute of Medicine. Building a better delivery system: A new engineering/health care partnership. Technical report, National Academy of Engineering and Institute of Medicine, 2005.

[62] J. Ostrowski, J. Linderoth, F. Rossi, and S. Smriglio. Orbital branching. *Mathematical Programming*, 126(1):147–178, 2011.

[63] O.Y. Özaltın, O.A. Prokopyev, A.J. Schaefer, and M.S. Roberts. Optimizing the societal benefits of the annual influenza vaccine: A stochastic programming approach. *Operations Research.*, 59(5):1131–1143, 2011.

[64] I. Ozkarahan. Allocation of surgeries to operating rooms by goal programming. *Journal of Medical Systems*, 24(6):339–378, 2000.

[65] D.N. Pham and A. Klinkert. Surgical case scheduling as a generalized job shop scheduling problem. *European Journal of Operational Research*, 185(3):1011–1025, 2008.

[66] R.T. Rockafellar and R.J.B. Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, 16(1):119–147, 1991.

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

[68] S.M. Sanchez and R.K. Wood. The "BEST" algorithm for solving stochastic mixed integer programs. *Proceedings of the 2006 Winter Simulation Conference*, pages 765–773, 2006. L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, eds.

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

[70] R. Schultz, L. Stougie, and M.H. van der Vlerk. Solving stochastic programs with integer recourse by enumeration: A framework using Gröbner basis reductions. *Mathematical Programming*, 83(1-3):229–252, 1998.

[71] S. Sen and J.L. Higle. The C^3 theorem and a D^2 algorithm for large scale stochastic mixed-integer programming: Set convexification. *Mathematical Programming*, 104(1):1–20, 2005.

[72] S. Sen and H.D. Sherali. Decomposition with branch-and-cut approaches for two-stage stochastic mixed-integer programming. *Mathematical Programming*, 106(2):203–223, 2006.

[73] H.D. Sherali and B.M.P. Fraticelli. A modification of Benders' decomposition algorithm for discrete subproblems: An approach for stochastic programs with integer recourse. *Journal of Global Optimization*, 22(1):319–342, 2002.

[74] H.D. Sherali and J.C. Smith. Improving discrete model representations via symmetry considerations. *Management Science*, 47(10):1396–1407, 2001.

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

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

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

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

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

[80] K. Stuart, E. Kozan, M. Sinnott, and J. Collier. An innovative robust reactive surgery assignment model. *ASOR Bulletin*, 29(3):48–58, 2010.

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

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

[83] R. Van Slyke and R.J. Wets. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17(4):638–663, 1969.

[84] P.M. Vanden Bosch and D.C. Dietz. Minimizing expected waiting in a medical appointment system. *IIE Transactions*, 32(9):841–848, 2000.

[85] R. Velásquez and M.T. Melo. A set packing approach for scheduling elective surgical procedures. volume 2005 of *Operations Research Proceedings*, pages 425–430. 2006.

[86] R.E. Wachtel and F. Dexter. Reducing tardiness from scheduled start times by making adjustments to the operating room schedule. *Anesthesia & Analgesia*, 108(6):1902–1909, 2009.

[87] P.P. Wang. Static and dynamic scheduling of customer arrivals to a single-server system. *Naval Research Logistics*, 40(3):345–360, 1993.

[88] J.P Watson and D.L. Woodruff. Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems. *Computational Management Science*, 8(4):355–370, 2011.

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

[90] J.D. Welch. Appointment systems in hospital outpatient departments. *Operational Research Quarterly*, 15(3):224–232, 1964.

[91] R.J. Wittrock. Advances in a nested decomposition algorithm for solving staircase linear programs. Technical Report SOL 83-2 Systems Optimization Laboratory, Stanford University, 1983.

[92] C.M. Yee. Scanning technology on surgery's cutting edge, 2007. Available from http://www.startribune.com/business/11244781.html, information and data accessed on May 11, 2009.

[93] B. Zeng, A. Turkcan, J. Lin, and M. Lawley. Clinic scheduling models with overbooking for patients with heterogeneous no-show probabilities. *Annals of Operations Research*, 178(1): 121–144, 2010.

[94] B. Zinn. AMC unveils new vascular operating room, 2009. Available from http://www.newsleader.com/article/20090409/NEWS01/904090344/-1/NEWS17, information and data accessed on May 11, 2009.

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