# ABSTRACT

LI, YUE. Development of an Operating Room Scheduling Support Information System. (Under the direction of Thom Hodgson and Javad Taheri.)

The dissertation focuses on development of an operating room (OR) scheduling support information system based on the study of a Veterans Affairs (VA) medical center. On one hand, there is a need for developing such a system which is smart enough to perform necessary data analysis, incorporate the analytical results with users' decisions, provide assistance and guidance for users in making daily OR schedules or preventing disorder due to personnel planned absence. On the other hand, the VA hospitals, due to their special role and administration policy, have many unique characteristics, e.g. budget control instead of profit driven, that should be addressed differently from the other medical environments that have been discussed in the literature.

We describe the development in four aspects: data analysis, simulation, scheduling and system design. A significant portion of this dissertation is devoted to analyzing the operating room turnover time, procedure duration, and length of stay(LoS) in the Post Anesthesia Care Unit (PACU) and the Intensive Care Unit (ICU). Modeling such random variables with high uncertainty is especially important in the simulation and the scheduling process. One unique aspect of our study lies in analyzing turnover time given limited information, which is rarely studied in the literature. We studied the effects from both service type and procedure complexity on the turnover time, and use practical methods for describing turnover time analytically using a data filtering technique. Our research on procedure duration includes both studying the seasonality and trend impacted by residents' learning curve, and estimating procedure duration with consideration of economic effects given the flexibility of early start of procedures. As for LoS in the PACU and ICU, we point out the necessity of considering the correlation with procedure duration and methods for handling this correlation. Using these data analysis results, we developed a simulation model for the OR centered hospital with processes including turnover, surgical procedure and recovering in the PACU, and resources including the OR, PACU, and the surgical team. We showed how to use the model for answering "what if" questions, and how to extend the model to evaluate schedules, in which surgeons' procedure duration estimates and procedure's cancelation risk are both considered. Then we developed our static scheduling model with consideration of unique aspects including the VA hospital's budget control, flexible early-start of procedures, the PACU resource's impact on the OR and procedure cancelation outside the regular OR operating time. The objective is to balance between high OR productivity and schedule quality. A practical method for solving the static problem is proposed, and tested. We also address the issue of generating real-time dynamic schedules based on the static model.

Incorporating users' need with the studies on data analysis, simulation and scheduling, we design a multi-user OR scheduling support information system that caters to different types of users based on their roles in the hospital.

Part of this dissertation also focuses on studying a new stochastic programming problem identified in this research. It is a two-stage stochastic linear programming problem with linking constraints that is generalized and relaxed from our scheduling model. We identify the special structure of such a problem and propose a new algorithm, Two-Stage L-Shaped Method, which is proved to converge to the optimal solution. Numerical tests using a modified Newsvendor problem show a significant computational advantage of this algorithm compared with solving it in CPLEX.

Development of an Operating Room Scheduling Support Information System

by
Yue Li

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Operations Research

Raleigh, North Carolina

2013

APPROVED BY:

| | |
|---|---|
| Stephen Roberts | Russell King |
| Thom Hodgson<br>Co-chair of Advisory Committee | Javad Taheri<br>Co-chair of Advisory Committee |

## DEDICATION

To my dear parents.

## BIOGRAPHY

Yue Li was born in Harbin, China, an "Ice City" famous for its world largest ice laten show. She studied in Beihang University (also known as Beijing University of Aeronautics and Astronautics) during 2004-2008, and received her Bachelor of Science degree in Information and Computational Science. She joined the Ph.D. program of Operations Research at North Carolina State University in 2008, and received a Master of Operations Research degree "en route" in 2010. She received a Graduate Industrial Traineeship at SAS Institute, Inc during 2011-2012 and upon graduation, she will join SAS as a research statistician developer.

**ACKNOWLEDGEMENTS**

First, I would like to thank my advisor, Dr. Thom Hodgson and Dr. Javad Taheri, for their profound ingenuity, patient guidance, and continuous encouragement. It is truly an honor to have them as my mentors, who care for students, and are always there when I need advice and guidance. Dr. Hodgson is a brilliant, knowledgeable, and humorous person, who is really pleasure to work with. I am grateful that he not only provided with me great academical ideas, taught me ways of learning and thinking, but also shared many philosophy of life that I will benefit throughout my whole life. Dr. Taheri is the one who brought us to this application area, coordinated with the VA medical center, extracted information we needed, and provided us with plenty of field insights. I am thankful that he made lots of efforts in this project, shared many of his technical experience, and has always been a source of encouragement and enthusiasm.

My committee members, Dr Stephen Roberts and Dr. Russell King, have generously given their time and expertise to better my work. I sincerely thank them for their great contribution and their good-natured support. I also thank my previous committee member Dr. Brian Denton, who has moved to University of Michigan, for his valuable comments.

I also want to thank my colleagues in our research group, Mingchun Zhao and Michelle Glatz. I really appreciate Michelle for her efforts in classifying the raw data, actively involving in the research discussion and proof reading my proposal. She gave me good impression of American culture and has always been very friendly and easy-going.

My sincere thanks and appreciation also go to the professors in the Operations Research program, particularly Thom Hodgson, Russell King, Yahya Fathi, Shu-Cherng Fang, Reha Uzsoy, James Wilson, Brian Denton - who taught me various courses at NC State and helped me in many ways. I had the honor to work as teaching assistant for Yahya Fathi and Shu-Cheng Fang in several courses and they all turned out to be wonderful experience for me.

I would like to gratefully acknowledge the Durham VA medical center for the support of our project. I had a wonderful visit there - it was a productive as well as fun-filled experience.

I also owe my thanks to SAS Institute, Inc for providing the industrial training opportunity and financing my last two years' graduate education. I would like to take this opportunity to thank my colleagues in SAS as well. My manager Alex Chien has always been really friendly and supportive. He provided me with a flexible and comfortable working environment, shared many insights on real-world application, and gave me plenty of encouragement in finishing my research work and graduate education. My colleague Pu Wang has gave me many advices on graduate study and helped me a lot in work. Many thanks to Ying Zhu, Lan Li, Ran Liu,

To all of my friends at NC State, it would not be such an invigorating experience without you. Special thanks goes to Zhibin Deng for our enjoyable discussion on optimization theory and programming codes, and Song Yang for all his helps since day one of my life in the US.

I also want to thank my fiance Weining Shen, who has been a tremendous source of encouragement for the last 4 years. I thank him for his guidance and support in every sphere of life. Life in the United States would not be so easy without him.

Finally, to my parents, needless to put an acknowledgement in words, but thank you for all your sacrifices, your dedication and your love. Without you, it would not even be possible to see this day, to you I dedicate this thesis.

# TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

The Veterans Health Administration is home to the United States' largest integrated health care system consisting of 152 medical centers, nearly 1,400 community-based outpatient clinics, community living centers, Vet Centers, and Domiciliaries. They provide comprehensive care to the 22 million veterans, serving more than 8.3 million veterans each year. Most of the Veterans Affairs (VA) Medical Centers operate in a similar way. Most offer medical and surgical specialty services including audiology and speech pathology, dermatology, dental, geriatrics, neurology, oncology, podiatry, prosthetics, urology, and vision care. Some also offer advanced services such as organ transplants and plastic surgery. Most VA medical centers are adjacent to medical schools whose faculty and residents perform services at the centers.

The problem to be studied is that of scheduling operating rooms (OR) at the Durham VA Medical Center. The Medical Center serves veterans in central and eastern North Carolina in its main medical center or one of its three community-based outpatient clinics. Services are available to more than 200,000 veterans living in a 26-county area. Because of the large demand for surgical services, the VA is trying to increase throughput in the OR's.

In this hospital, most of the surgical procedures done are elective. The procedures can be categorized into fourteen different service types: Cardiac, General, Gynecology, Neurology, NSU, Otolaryngology Head and Neck Surgery (OHNS), Ophthalmology, Oral, Orthopedics, Plastic, Podiatry, Thoracic, Urology, and Vascular. There are eight ORs in the hospital. The standard operating hours for OR1 and OR2 are 10 hours per day, and eight hours per day for OR3 through OR8. The service type for each schedule block (a combination of date and room, e.g. Oct. 5th morning in OR 1) is predetermined according to a "Priority Schedule Table" made by the hospital two months in advance.

They begin to let the patients know about the day of the surgery one month before the surgery but it can change up to the day before. Each service needs to post their desired surgery schedule three days before the surgery date. The surgeon fills out a form for each surgery they plan to do and submit it to the scheduler. In the desired schedule, they state which patient is first, second, and so on. The scheduler estimates the duration of each surgery based on the average historical duration of that procedure in their system, the surgical team (how well the team works together, or if resident physicians are involved which increases the duration because of the teaching process, etc.). The scheduler may also consider the surgeon's estimate of time needed. However, he/she believes that they always underestimate the time, and their estimates are mainly the time from the incision to closing. The time the scheduler posts in the schedule is for the time from patient in to patient out, so the scheduler has to add at least half an hour to the surgeon's estimate. The scheduler's time estimate is usually in 30 minute units (e.g., 1 hour 30 minutes or 2 hours), the best they can do will be in 15-minute segments. Based on the sequence and estimated duration of each surgery, the scheduler builds a schedule, and then negotiates with each surgeon until they agree with the sequence and a planned start time/end time for each surgery. If there is any change to the planned schedule, they notify the patient. By 1pm on the day before the surgery, the scheduler posts a schedule. It includes the procedure, the room, the time (e.g., 8:00-12:00), surgeon, etc. The scheduled start time of the first procedure is 9 am on Wednesdays and 8 am for all other days. The scheduled turnover time between two surgeries is 30 minutes.

There are four emergency levels for add-on cases. They indicate how soon the patient should be taken into the OR: 1 means immediately, 2 means within 6 hours, 3 means within 24 hours and 4 means any time. If a level 1 case shows up during the day, the patient will usually be taken into the next available OR. There is a rare case when all ORs are occupied and no OR is available soon when a level 1 case shows up. So they may stop a surgery that just started. There are quite a few level 4 cases (the surgeon may really want to do this surgery on this day, but there seems to be no time/OR space), they may be on the schedule with a category name "TSA" (time space available). Whether they will be done on that day or not and where depends on how everything goes on that day: e.g. a case is canceled; the surgeons are fast, etc.

On the day of the surgery, there are several preoperative activities (e.g. brief physical examination, anesthesia, etc.) done before the pre-assigned surgery time. The patient is brought into the PACU for pre-op (brought in at 6:30 am for the 8am first case), the surgical team comes and discusses the case with the patient. If everything goes as planned, the anesthetist begins to do his/her job and then bring the patient to the OR. At the same time, the nurses prepare equipment in the OR (set-up). They sometimes bring in two patients in the morning in case the first case is canceled. In this case, the set up time may be longer, so the case may start later. When the patient is in the OR, the professional people in the OR (nurses for sure, and surgeons if they are already in the OR) will start to prepare the patient for surgery, e.g. restrain the body, clean partial skins, etc. The anesthetist does his/her anesthesia job. When everything is ready, the surgeons start the surgical procedure. Before the surgery ends, the nurse calls the

Post Anesthesia Care Unit (PACU) and tells them to bring the a bed into the OR. If there is no bed in the PACU, the patient must wait in the OR after the surgery. When the closing is done, the anesthetist starts to wake the patient up. Most of the patients are moved to the PACU for at least one hour for really waking up. Then they are taken into the ward, Intensive Care Unit (ICU) or home. After the patient leaves the OR, the nurses turnover the OR and then set up for the next surgery.

If the first surgery ends early and the next patient is ready, the surgical team starts the next one earlier than planned. So the scheduled start time is really a guideline instead of a rule. If the next to last surgery planned for the day ends later than the OR close time, the surgical team decides whether they will continue to do the last one. But since most of the surgeries are elective, they usually just cancel it.

The daily OR schedules are manually created by a "scheduler". Our objective is to develop an information system to assist the staff in making daily schedules for the ORs. The system should provide performance evaluation corresponding to each schedule, so that the scheduler can foresee the effects of the schedule and adjust scheduling decisions accordingly. It should also provide suggestions for improving the schedule. When the regular "scheduler" is on leave, the system should be able to provide some help in OR scheduling to avoid disorder.

As an application problem, the challenges come from three perspectives: accuracy, efficiency and the ability to generalize. First, the model we build to describe the system, needs to be as accurate as possible. It is driven by both the cost of running the ORs and the demands for increasing throughput. Uncertain surgical durations, patients' recovery time, etc. increase the uncertainty of the whole process, so we need to perform a thorough analysis of the data to reduce the inaccuracy in describing the process. In addition, the availability of different resources like personnel and critical equipment complicate the process more. Thus a simulation model that takes into consideration the impact of different resource's availability should be built. The model will never behave exactly like reality, so we need to find a balance between a high level of detail and the ease of handling the model. Second, the system should meet the users' need and provide a practical solution. We should understand the factors users need to consider when they are making scheduling decisions and provide them with corresponding reliable information. This requires a significant amount of model validation and user involvement. In this paper, as the early stage of our study, we were not able to show the effect of the study, so the system is designed based on our current understanding of the needs. With improved understanding, the system will be adjusted to achieve this goal. Last but not least, the method we propose should be able to be generalized to other situations even though this is a scenario based study.

Figure 1.1 shows the structure of the main components of the system. We can consider the whole system as three main components: database; engine; and users' interface. In the engine, we mainly focus on data analysis, scheduling and simulation.

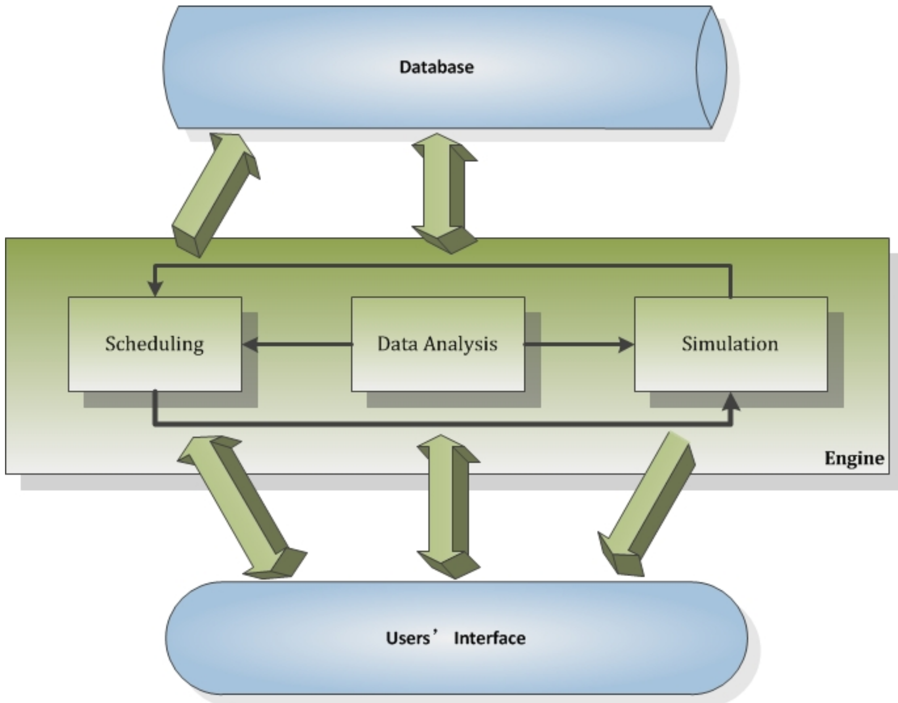



Figure 1.1: System Structure

Combining the information obtained from the users' interface with the data stored in the database, data analysis can be performed. Based on the results of data analysis, users can make schedules. The simulation will provide performance analysis for schedules. Users can adjust their schedules based on the simulation results. The system can also automatically generate schedules. Then the information and schedules will be stored in the database for future use.

In Chapter 2, we review previous work found in the literature. In Chapter 3, we show our data analysis methods and results. In Chapter 4, we describe how we simulate the hospital system based on the data analysis results. In Chapter 5, we describe how we solve the single OR scheduling problem. In Chapter 6, we describe in detail the system structure, its basic functions and interface for a multi-user decision support system, enabled by our analytical solution. In Chapter 7, we generalize the stochastic programming model used in Chapter 5 and introduce a new algorithm for solving this model optimally and efficiently. This turns out to be an unexpected opportunity to develop new methodology. Finally, in Chapter 8, we discuss conclusions and potential future work.

# CHAPTER 2

# LITERATURE REVIEW

Operating rooms, where about 42% of a hospital's revenue is generated [HFMA, 2003], have received increasing attention over the years. Based on a comprehensive literature review performed by [Cardoen et al., 2010], there are 247 references relevant to operating room planning and scheduling. The following reviews focus on three aspects: uncertainty, scheduling and implementation.

## 2.1 Uncertainty

The uncertainty inherent in surgical service times is a major issue in the development of accurate operating room schedules. Among the literature addressing uncertainty, modeling the surgical case duration is the most widely studied problem. [Wright and Bonar, 1996] showed that combining surgeons' estimates and prior case duration data together outperforms either used separately. [Hancock et al., 1988] introduced a methodology to provide procedure times based on a historical data base. The data is first subdivided by code, primary (staff) surgeon, case teaching status, patient inpatient/outpatient status, patient sex, patient age, etc. Then rules, testing and pooling are introduced to further cluster the data. [Dexter et al., 2008] systematically reviewed the articles reporting statistically significant differences in preoperative times with the specialty of general thoracic surgery, which is a typical specialty of elective surgery. They concluded that it is important to rely on the precise procedure(s), surgical team, and type of anesthetic when estimating case durations. These works provide us with suggestions on the factors that potentially have impact on procedure durations. Since OR times differ greatly for the same procedure in different hospitals [Dexter et al., 2006], we need to identify factors crucial to predict procedure durations in our case.

A constant time period (e.g. 15 minutes) is commonly used as turnover time for all services in the literature [Denton et al., 2006]. However, we found that such an assumption is not valid in practice. Turnover time should be considered as a service specific random variable. Some work has modeled the turnover time differently. [Marcon et al., 2003] considered the surgical preparation duration and clean-up duration to be dependent on the specialty. [Marcon and Dexter, 2007] adopted both a bounded two parameter Lognormal distribution and a constant.

As for Length of Stay (LoS) in the PACU, there are fewer studies using different methods. [Marcon and Dexter, 2006] assumed scheduled OR times, turnover times and PACU LoS are different for different services, and they follow Lognormal distributions but are also bounded. The PACU LoS in [Marcon et al., 2003] 's work were estimated by the anesthesiologists and the OR manager. They were dependent on range of surgical case duration and the results were mentioned to be in accordance with a statistical analysis conclusion that PACU LoS was 46% of the total length of anesthesia, which was obtained from a 2-year database in a French public hospital. Analysis about general surgery in [VanBerkel and Blake, 2007] showed that there is a statistical difference of LoS between elective patients, non-elective patients, and non-surgery patients, and LoS was statistically different for some categories. [Belin and Demeulemeester, 2007] assumed that LoS followed a Multinomial distribution with parameters which depend on the type of surgery.

## 2.2 Scheduling

Simulation proves to be a useful methodology for dealing with complex and stochastic problems because of its extensive modeling flexibility. It is used in many cases to study policies for operating room scheduling and resource allocation.

Much of the surveyed research only focused on the OR itself. [Dexter et al., 1999] studied block time allocation for surgeons using simulation and a patient survey. (*Schedule block* is an interval of time on a specified day in a specified OR that is assigned to a service for scheduling its surgical procedures (e.g. 8 am to 12 pm on Oct. 5th in OR 1)). Various block scheduling methods were evaluated and it was shown that operating room utilization can be maximized by allocating block time for the elective cases based on expected total hours of elective cases, scheduling patients into the first available date provided open block time is available within 4 weeks, and otherwise scheduling patients in "overflow" time outside of the block time. [Dexter and Traub, 2002] tested two strategies for scheduling a new case into an OR, the Earliest Start Time and the Latest Start Time, on the efficiency of the use of an OR. It was shown in several scenarios that Earliest Start Time performed substantially better when the objective was to minimize overutilization, while Latest Start Time would perform better at balancing workload among different services' OR time. Simulation using historical data was used to show that these two heuristics are robust.

A few researchers dealt with an integrated operating room or simulated the entire hospital providing a systematic view. [Marcon et al., 2003] devised a simulation model to determine the minimum number of PACU beds, and investigated how factors like LoS and number of porters influence the hourly occupancy of the PACU. Their model included ORs, PACU and OR staff. The preoperative process was composed of transportation of the patient from a ward bed to the OR, anesthesia induction, surgical preparation, surgical procedure, patient's PACU stay and transportation of the patient back to the ward. [Marcon and Dexter, 2006] used discrete event simulation to study the impact of several different surgery sequencing rules on the phase I PACU staffing and over-utilized OR time resulting from delays in PACU admission. The best rules were shown to be those that smooth the flow of patients entering the PACU. In their model, they assumed the actual OR time of each case to be equal to the scheduled OR time multiplied by a normally distributed random number with a mean of 1 and standard deviation of 0.25. [Denton et al., 2006] applied a Monte-Carlo simulation model and simulated annealing to schedule a multi-OR surgical suite. The process in the simulation included intake, surgery and recovery. [Baumgart et al., 2007] proposed a conceptual framework of using computer simulations in different stages of the business process management lifecycle for operating room management.

As for optimization methods in OR scheduling problems, many previous research works adopt deterministic models. [Aida Jebalia, 2006] developed a two-step Mixed Integer Programming (MIP) model. The first step consists of assigning surgical operations to operating rooms. Resources including ORs, surgeon, equipment, ICU, etc. are all taken into consideration. The second step focus on sequencing the assigned operations with the objective of minimizing the total overtime in all ORs.

There are also a number of stochastic programming optimization models for OR scheduling. [Denton and Gupta, 2003] studied a single OR scheduling problem with the objective of minimizing the total expected cost of customer waiting time, server idle time, and tardiness with respect to the session length. They developed a modified L-shaped algorithm based on derived upper and lower bounds that are independent of procedure duration type. [Denton B, 2007] described a stochastic optimization model and some practical heuristics for computing surgery sequencing and start-time decisions. [Batun et al., 2011] present a two-stage stochastic mixed integer programming model to minimize total expected operating cost for a generalized parallel operating room environment.

## 2.3 Implementation

As mentioned in [Brailsford, 2005] and [Cardoen et al., 2010], there are very few examples of implementation. [Ernst et al., 1977] described the implementation of a software program system. Requests are sorted and assigned using the system according to their relative priority, which is based on the service priority, suggested time interval, surgeon priority, and room preference. [Hanson, 1982] reported a system providing procedure-specific information, special equipment reservation and availability of equipment. [Ozkarahan, 1995] introduced an expert hospital decision support system which combined mathematical programming, knowledge base, and database technologies. An enhanced version using a goal programming model was described in [Ozkarahan, 2000]. [Harper, 2002] proposed a generic framework. It features a system creating statistically and clinically meaningful patient groups using Classification and Regression Tree analysis, and estimating the parameters of statistical distributions using a simplex optimization algorithm. A simulation tool for hospital resources was also introduced. The framework was illustrated by cases drawn from a set of local hospitals. [Harper and Gamlin, 2003] applied a simulation modeling approach to examine various appointment schedules in order to reduce patient waiting times in an outpatient department. [Belin et al., 2006] developed a software system that visualizes the impact of the master surgery schedule on demand for various resources throughout the hospital. And later in [Belin et al., 2009], the author presented a corresponding decision support system relying on both a mixed integer programming technique solving multi-objective linear and quadratic optimization problems and a simulated annealing meta-heuristic.

From the literature review it seems that most scheduling work stays in the position of academic research and consultant analysis, instead of actually putting results to use. One reason is that the academics' need to demonstrate theoretical or methodological advances tends to lead to complex, sophisticated models, in contrast with the objective of the end-user: a simple, easy-to-use model [Brailsford, 2005]. There is a clear need for generic systems acceptable to users, balancing user-friendliness with scientific rigor and validity.

# CHAPTER 3

# DATA ANALYSIS

We consider the hospital as a system centered on the ORs. There are three important processes of the OR-centered system: OR turnover, performing a surgical procedure, and patient stay in the PACU and/or ICU. The distribution of turnover time, procedure duration and LoS in the PACU and ICU are critical to simulating the whole system and generating OR schedules. We also need to give point estimates of these times so that the users can use the information for making their decisions.

Our process of understanding the system started with analyzing historical data. The records of case ID, Current Procedural Terminology (CPT) code, OR, date, patient in OR time, patient out of OR time, start time and end time of each surgical procedure performed are available. Patient movement information includes patient id, ward location, transaction (could be check in, transfer, check out, etc.), transaction times are also availabe.

The CPT code is used to describe medical, surgical and diagnostic services. Since there is no record about which service type each procedure should be categorized into, we used the CPT code to identify the service type. Six months of data (Aug. 15th 2009 - Feb. 10th 2010) was collected. The procedure names and descriptions of CPT codes were manually checked to categorize the corresponding procedures into different services. The CPT code ranges corresponding to each service type were recorded. In the following study, the service for each procedure is determined by these identified CPT code ranges. Most of the later data analysis results are based on 4.5 years of data (from Oct. 3rd 2006 to Apr. 11th 2011) from the orthopedics service, which is a representative elective surgical service.

First, it is necessary to define some common terms:

*Procedure duration* is defined as the time from when a patient enters an OR until his/her surgical procedure is finished and the patient is ready to leave the OR. It includes three parts: the duration for preoperative activities, duration for the surgery and the duration for postoperative activities. It does not include a delay in OR because of the availability of PACU.

*Turnover time* is the time used to prepare for the next procedure. It includes cleanup times and set up times, but not the delays between contiguous procedures in the OR. The delays may be caused by the unavailability of surgeons or critical equipment.

*Length of Stay in PACU* is defined as the time from when a patient enters a PACU after his/her surgical procedure until the patient leave the PACU.

In this chapter, we describe in detail how to perform data analysis to reduce the uncertainty in our description of three time related variables and show our data analysis results. The three types of time related variables are crucial to OR scheduling: turnover time, surgical procedure duration, and LoS in the PACU and the ICU.

## 3.1 Turnover Time

The difficulty of estimating turnover time lies in the fact that there is no direct data relative to actual turnover times. The most relevant data we have is the time gap between two consecutive procedures. The time gaps may be caused by not only the turnovers, but also the availability of surgical teams and critical equipment, patient readiness, etc.

In order to get a better estimate of turnover time, we only use the type of time gap illustrated in Figure 3.1, in which the first procedure ends after the scheduled start time of the following procedure. The advantage of only using this type of time gap is that it reduces the randomness caused by the schedule. In other words, if the first procedure ends earlier than the scheduled end time, then the patient and surgical team for the following surgery may not be ready, so the time gap would be longer. The data summary of the 4.5 years' worth of sample data for both the original time gap and the filtered time gap are listed in Table 3.1. The workflow for getting the time gap and the filtered time gap is described in Figure 3.2.

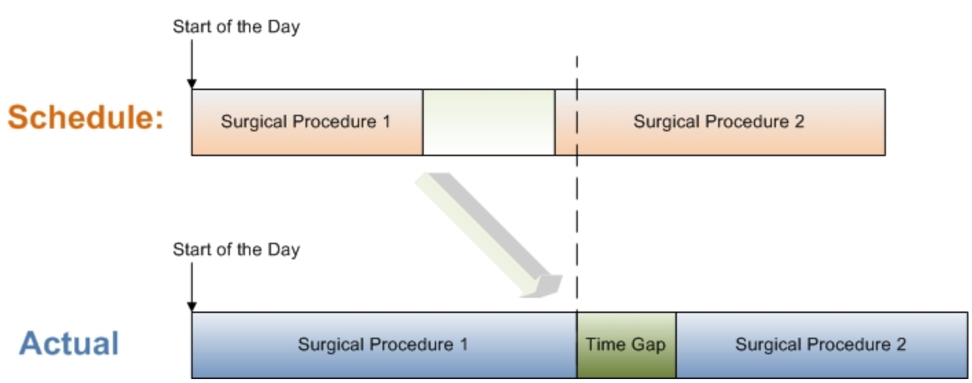



Figure 3.1: Illustration of Filtered Time Gap

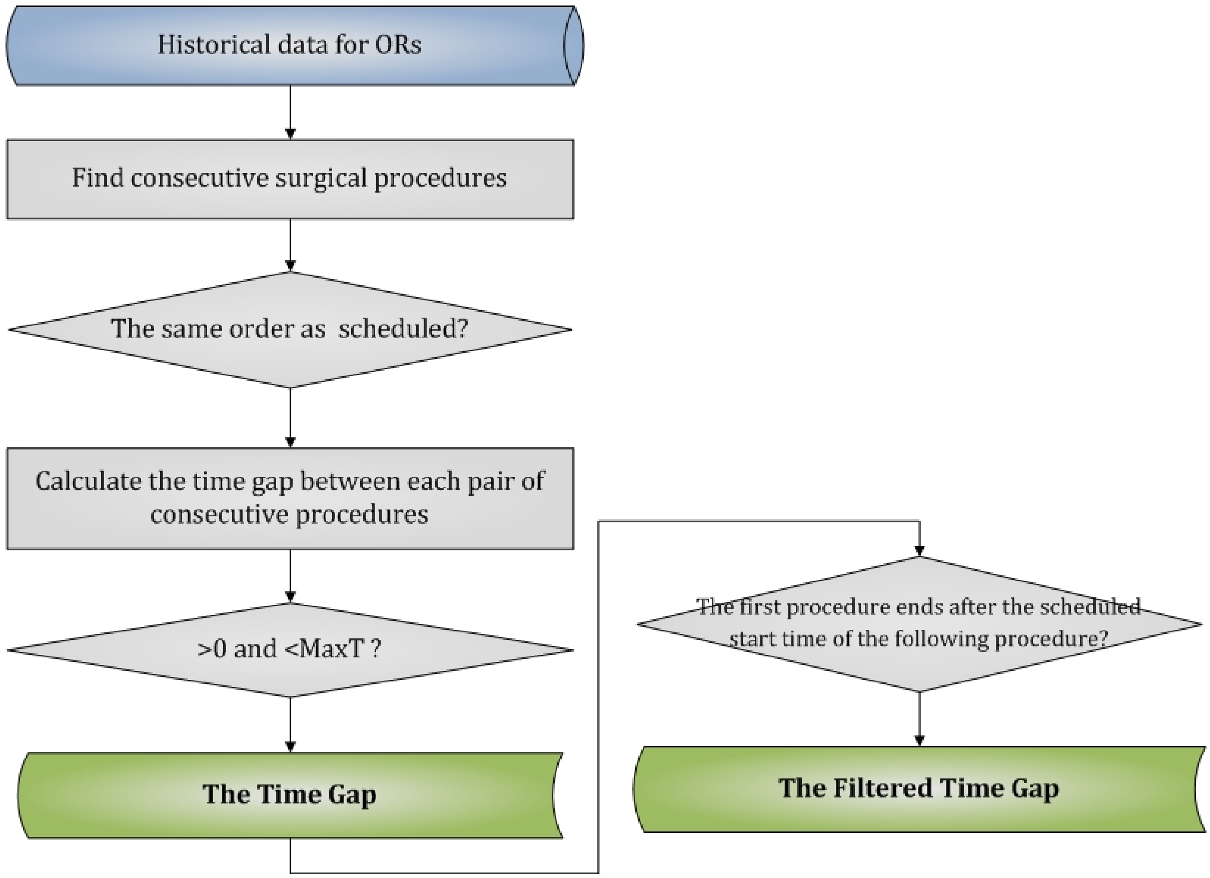



Figure 3.2: Workflow for Getting the Time Gap and the Filtered Time Gap

Table 3.1: Sample Summary for Time Gap and Filtered Time Gap (in the unit of minute)

| | Time Gap | Filtered Time Gap |
|---|---|---|
| Number of Observations | 1065 | 375 |
| Max | 497 | 194 |
| Mean | 48.1 | 45.7 |
| Standard Deviation | 30.4 | 23.2 |
| Coefficient of Variation | 0.63 | 0.51 |

In the following sections, we first study the effects of service type and procedure complexity respectively in estimating the turnover time, then we describe how to find the distribution of

Table 3.2: Estimates for Expected Turnover Time

| Service Type | Cardiac | Oral | General | Gyn | Neurology |
|---|---|---|---|---|---|
| Turnover Time | 60 | 51 | 15 | 41 | 35 |
| Service Type | OHNS | Ophthalmology | Orthopedic | Plastic | Podiatry |
| Turnover Time | 33 | 15 | 30 | 21 | 45 |
| Service Type | Thoracic | Urology | Vascular | | |
| Turnover Time | 38 | 25 | 29 | | |

turnover time based on the filtered time gap.

### 3.1.1 Estimates of Turnover Time for Different Services

Since the time gap is an upper bound for turnover time, the expected turnover time should be a certain percentile of the time gaps. Instinctively, the turnover time for different services (e.g. Cardiac and Ophthalmology) may be different. To estimate turnover times, we collected one month of historical data (Oct. 2009) pertaining to time gaps between two surgical procedures of the same service performed in the same OR on the same day. We use the 20th percentile of the collected data as our estimate of the expected turnover time for each service in our study (Table 3.2). Using this percentile, we are able to remove some outliers due to data entry errors and also help to remove the idle time in the time gap.

The results are quite reasonable based on verification by people working in the hospital. The flaw here is that the analysis is based on one month's data and the 20th percentile selection is arbitrary based on the results. We also applied the same method on 5 years of historical data from orthopedics procedures, and we got the same results, 30 minutes, as shown in the table, which supports the robustness of the 20th percentile method.

We also used these results in estimating how many additional cases the hospital could schedule when they open an OR room only for minor surgeries. Since minor surgeries usually take less time, the accuracy of turnover time estimates would be more critical compared with scheduling major surgeries. Our estimates based on these results were considered reasonable to the hospital personnel, which also shows the reasonability of the turnover time estimate results in Table 3.2.

### 3.1.2 Impact of Major/Minor Surgeries on Turnover Time

It was suggested by hospital staff that surgical complexity, major or minor, may affect the turnover time between two procedures. For example, if two consecutive procedures are both major surgeries, the turnover time would be longer than if they are both minor surgeries.

To test this hypothesis, records of Oct. 2009 were used. The surgeries were categorized into two surgery types: major and minor. According to the surgery complexity of every pair of consecutive surgeries, each turnover time was classified as one of the four turnover types: major-to-major, major-to-minor, minor-to-major and minor-to-minor. For each service, Analysis of Variance (ANOVA) was used to test the differences between expected time gaps of these four turnover types. Statistically, there was no significant difference between the expected time gaps of these four types, which implies no significant difference between the expected turnover times.

This study showed that based on a limited data set, whether two surgeries are major or minor surgeries does not significantly affect the turnover time between them. However, the study was based on historical data from only one month, so further analysis using more data is needed.

Since there is no direct record about a surgery being classified as either major or minor, lots of expert opinions from hospital staff are needed for this study. Under an assumption that major surgeries take more time and minor surgeries take less time, we can address this issue by checking the relation between the durations of two consecutive procedures and the time gap length in between. Figure 3.3 shows the bubble plot for the filtered time gap. The

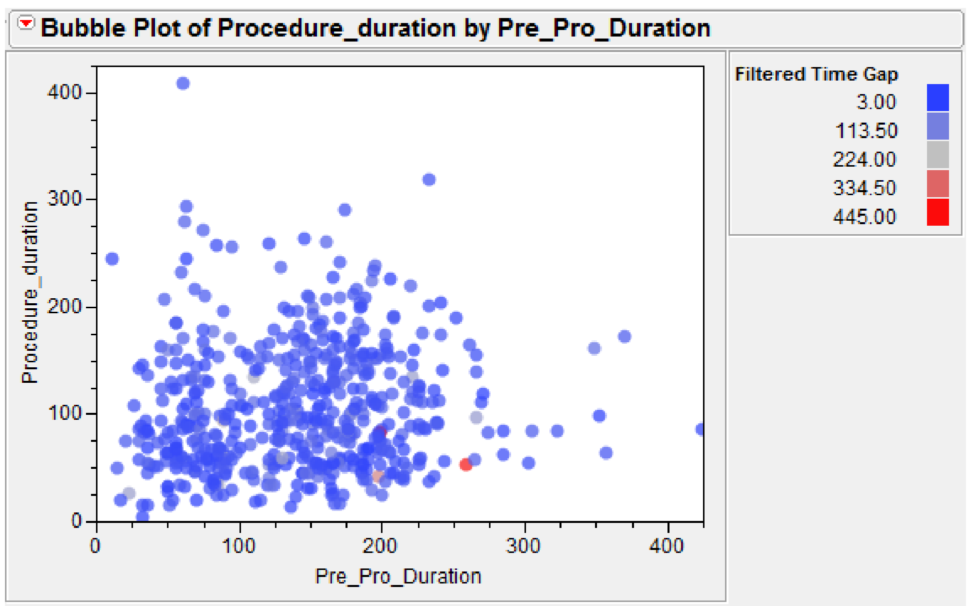



Figure 3.3: Bubble Plot for Filtered Time Gap

horizontal axis and vertical axis show the first procedure duration and the following procedure

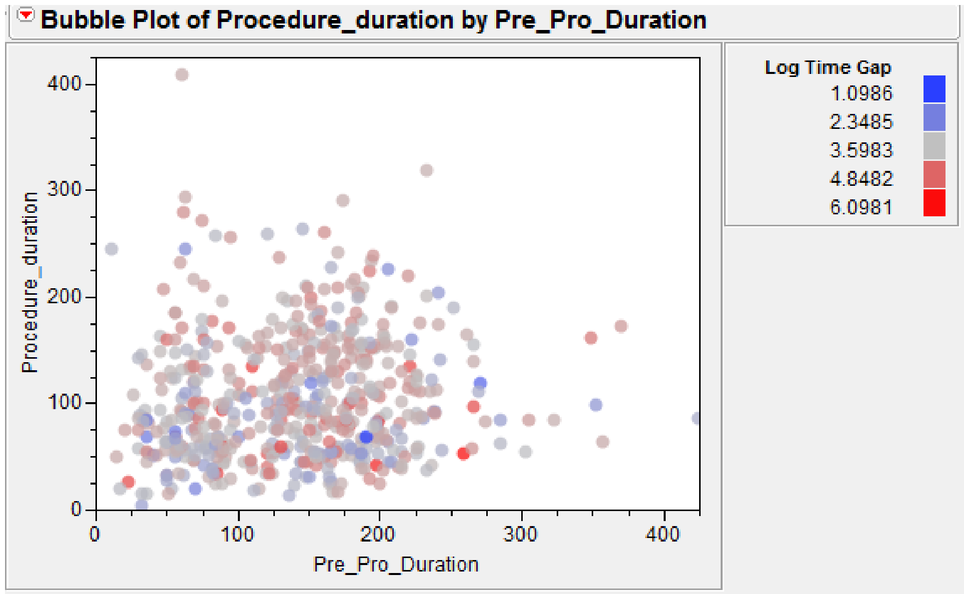



Figure 3.4: Bubble Plot for Log Transform of Filtered Time Gap

duration separately, and the color of the bubble shows the filtered time gap as explained in the legend. Since the filtered time gap has a long end tail, the choices of the color make the results not quite obvious. Thus, we take the natural logarithm transform of the filtered time gap and the corresponding results are shown in Figure 3.4. We can see that there is no obvious color pattern, which means there is no significant result showing that the procedure lengths would impact the turnover time. A linear model is also tried with the filtered time gap as the dependent variable and the two procedure durations as the explanatory variables. The p-values for the two explanatory variables are 0.9315 and 0.1188, respectively, which also supports our conclusion that procedure durations have no significant impact over the turnover time.

### 3.1.3 The Distribution of Turnover Time

Turnover time is a random variable (i.e. one can't guarantee that it takes exactly 15 minutes every time, maybe it will take 13 minutes this time, and 18 minutes next time). So there is a need to model the distribution of turnover time.

The focus of our research is to address turnover time analytically. Although we do not have historical data on actual turnover times, we can use the time gaps between two consecutive procedures. It is reasonable to assume that the gap, which is a random variable, is composed of two random variables, the actual turnover time and the idle time (Figure 3.5). One may assume that the turnover time follows some distribution, and the idle time may follow another distribution. Since we know what the distribution of the time gaps between consecutive orthopedic procedures looks like the shape shown in Figure 3.6, we can compute different convolution results of the two random variables under different assumptions of distributions, and compare the results with the distribution of the time gaps. Thus, we attempt to find a reasonable distribution of the turnover time in this way.

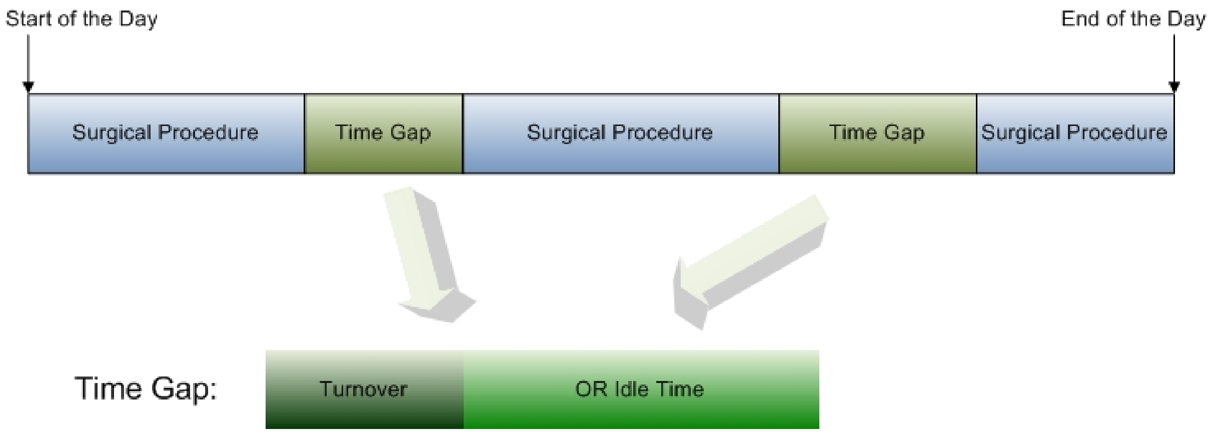



Figure 3.5: Components of Time Gaps between Consecutive Procedures

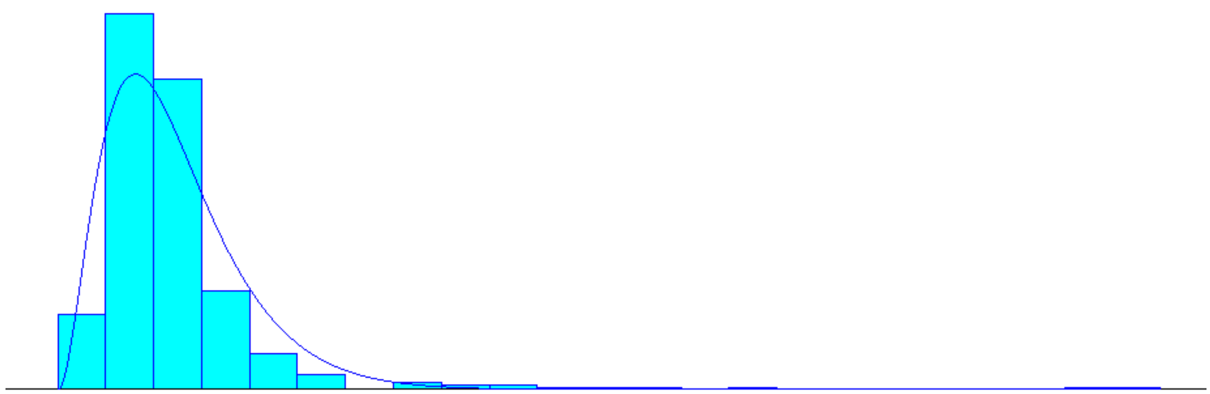


Figure 3.6: Histogram of Time Gaps between Consecutive Orthopedics Procedures

Since the turnover should be done within a limited time and it should have less variance compared with the idle time, one reasonable assumption would be that the turnover time follows a bounded distribution, such as a Uniform distribution or a Triangular distribution. In our study, we consider 4 distributions, 1-parameter Uniform, 2-parameter Uniform, 1-parameter symmetric Triangular and 2-parameter Triangular distributions. Except for the 2-parameter Uniform distribution, the remaining three distributions all use 10 minutes, which is the shortest turnover time in the historical record, as the left-side boundary. This way, we reduce the number of parameters that need to be estimated by 1.

As for the idle time, we tried well-known distributions including Exponential, 2-stage Erlang, 2-parameter Hypo-exponential and Lognormal distributions. The first two distributions contain 1 unknown parameter and the other two contain 2 unknown parameters.

We only consider the combined models with at most 3 unknown parameters. This is because in order to solve 4 parameters, we would need, at the least, information about kurtosis, which is the fourth moment, however, as we know, higher moments tend to be less accurate and robust than lower moments if we estimate them from a sample. So in our study, we do not assume the accuracy of moments higher than the third moment.

Using the moments' information from the samples, we can estimate the parameters by solving an optimization problem. For example, for models with 3 unknown parameters, we minimize the absolute difference between the model skewness and the sample skewness, with the constraints that the mean and variance of the model equal the mean and variance of the sample. Since the feasible region is not convex, we first solve the two equations in the constraints to represent the remaining 2 parameters using the first parameter in the turnover time distribution. This way we can covert the optimization problem with 3 variables into one with only 1 variable. Since this variable is bounded, we can choose a sufficient number of initial solutions within the boundaries to develop a deepest gradient algorithm to solve the non-convex nonlinear programming problem. The details of the models are listed in Appendix A.1.

The Kolmogorov-Smirnov and Chi-Square tests are performed to test how good our models are using the filtered time gap data based on the 4.5-year of historical data. Table 3.3 shows the size of the samples, and Table 3.4 shows the p-value in the Kolmogorov-Smirnov test for each model and each sample year and Table 3.5 shows that for the Chi-Square test.

Table 3.3: Number of Observations in the Samples

| Sample | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 |
|---|---|---|---|---|---|---|
| Number of Observations | 20 | 80 | 79 | 107 | 67 | 20 |

By only using the moments' information, we can solve the problem quickly. However, the disadvantage is that the higher the moments, the less reliable they become. This is one of the most important reasons that we only selected the above models instead of other models with

Table 3.4: P-value in Kolmogorov-Smirnov Test for 12 Different Models by Matching Moments

| Model | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 |
|---|---|---|---|---|---|---|
| 1-Unif. +Expon. | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 |
| 1-Unif. +Erlang2 | <0.05 | <0.15 | >=0.20 | >=0.20 | >=0.20 | >=0.20 |
| 1-Unif. + Hypoexpon.2 | >=0.20 | <0.20 | <0.01 | >=0.20 | >=0.20 | >=0.20 |
| 1-Unif. +Lognorm.l | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 |
| Sym-Triang. +Expon. | >=0.20 | <0.10 | >=0.20 | >=0.20 | >=0.20 | >=0.20 |
| Sym-Triang. +Erlang2 | >=0.20 | <0.15 | >=0.20 | >=0.20 | >=0.20 | >=0.20 |
| Sym-Triang. +Hypoexpon.2 | >=0.20 | <0.10 | <0.01 | >=0.20 | >=0.20 | >=0.20 |
| Sym-Triang. +Lognorm. | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 |
| 2-Unif. +Expon. | >=0.20 | <0.01 | >=0.20 | <0.05 | <0.05 | >=0.20 |
| 2-Unif. +Erlang2 | >=0.20 | <0.15 | >=0.20 | >=0.20 | >=0.20 | >=0.20 |
| 2-Triang. +Expon. | >=0.20 | <0.10 | <0.01 | >=0.20 | >=0.20 | <0.01 |
| 2-Triang.+Erlang2 | <0.01 | <0.01 | <0.01 | <0.01 | >=0.20 | <0.01 |

Table 3.5: P-value in Chi-Square Test for 12 Different Models by Matching Moments

| Model | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 |
|---|---|---|---|---|---|---|
| 1-Uniform +Exponential | 0.5375 | 0.3155 | 0.2114 | 0.6749 | 0.3303 | 0.6487 |
| 1-Uniform +Erlang2 | 0.8392 | 0.0732 | 0.0379 | 0.7415 | 0.2857 | 0.6487 |
| 1-Uniform + Hypoexponential2 | 0.5375 | 0.2790 | 0.0000 | 0.7053 | 0.2763 | 0.7836 |
| 1-Uniform +Lognormal | 0.6692 | 0.6363 | 0.0320 | 0.7556 | 0.2029 | 0.6868 |
| Sym-Triangular +Exponential | 0.5494 | 0.1526 | 0.1719 | 0.7240 | 0.1745 | 0.8931 |
| Sym-Triangular +Erlang2 | 0.5859 | 0.0618 | 0.0034 | 0.7143 | 0.2949 | 0.9847 |
| Sym-Triangular +Hypoexponential2 | 0.5375 | 0.1526 | 0.0000 | 0.7240 | 0.1745 | 0.9847 |
| Sym-Triangular +Lognormal | 0.6692 | 0.6363 | 0.0690 | 0.7556 | 0.2029 | 0.8931 |
| 2-Uniform +Exponential | 0.5375 | 0.0000 | 0.4100 | 0.0001 | 0.0229 | 0.7836 |
| 2-Uniform +Erlang2 | 0.5094 | 0.0732 | 0.4156 | 0.7415 | 0.3670 | 0.9847 |
| 2-Triangular +Exponential | 0.5494 | 0.1526 | 0.0000 | 0.6387 | 0.1551 | 0.0000 |
| 2-Triangular +Erlang2 | 0.0000 | 0.0004 | 0.0000 | 0.0006 | 0.2857 | 0.0000 |

more parameters.

To reduce the impact caused by the unreliability of higher moments, we can borrow information from the cumulative distribution function (CDF), which is more reliable than moments. We can change the objectives of the optimization problem to that of minimizing the difference between the sample CDF values and the model CDF values. The disadvantage is that we will increase the computation time (from several minutes to a day for each sample) because there are no closed form CDFs for most models and we have to generate a set of numerical CDFs for each feasible solution for each model. The advantage is that we can get more accurate results (Table 3.6 and Table 3.7). So we can use the method if we have enough time and use the faster method that matches the higher moment when a quick result is desired. The details of the models using the closed CDF form are listed in Appendix A.2.

Table 3.6: P-value for Kolmogorov-Smirnov Test for 12 Different Models by Matching CDF

| Model | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 |
|---|---|---|---|---|---|---|
| 1-Unif. +Expon. | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 |
| 1-Unif. +Erlang2 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 |
| 1-Unif. + Hypoexpon.2 | >=0.20 | >=0.20 | <0.01 | >=0.20 | >=0.20 | >=0.20 |
| 1-Unif. +Lognorm. | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 |
| Sym-Triang. +Expon. | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 |
| Sym-Triang. +Erlang2 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 |
| Sym-Triang. +Hypoexpon.2 | >=0.20 | <0.10 | <0.01 | >=0.20 | >=0.20 | >=0.20 |
| Sym-Triang. +Lognorm. | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 |
| 2-Unif. +Expon. | >=0.20 | <0.01 | >=0.20 | <0.05 | <0.05 | >=0.20 |
| 2-Unif. +Erlang2 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 |
| 2-Triang. +Expon. | >=0.20 | >=0.20 | <0.01 | >=0.20 | >=0.20 | <0.01 |
| 2-Triang. +Erlang2 | <0.01 | <0.01 | <0.01 | <0.01 | <0.15 | <0.01 |

Aside from the turnover and regular idle times, sometimes the filtered time gap may include some noise. For example, the time gap could be very large due to some typographical errors (typos), since the data is manually entered into the database. So we need a method to detect and remove the noisy data point. We extend the ideas of the Kalman filter [Kalman, 1960] to apply it here for modeling the distribution of turnover time. The basic steps are listed below.

**Algorithm 1** Kalman Filtering for Modeling the Distribution of Turnover Time

*Step 0.* Let the data set $D$ contain the corresponding time gaps data, and Set a parameter $\alpha$.
*Step 1.* Given data set $D$, Find the best-fit distribution with CDF $F$.
*Step 2.* Find the data set $d = \{x|F(x) > 1 - \alpha, x \in D\}$.
If $d = \emptyset$ : Found the best-fit distribution, Stop.
Else : Set $D = D \backslash d$, and Go back to Step 1.

Table 3.7: P-value for Chi-Square Test for 12 Different Models by Matching CDF

| Model | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 |
|---|---|---|---|---|---|---|
| 1-Uniform +Exponential | 0.8087 | 0.7172 | 0.6414 | 0.5991 | 0.2881 | 0.9875 |
| 1-Uniform +Erlang2 | 0.5859 | 0.8527 | 0.4395 | 0.8528 | 0.2881 | 0.9998 |
| 1-Uniform + Hypoexponential2 | 0.5094 | 0.2680 | 0.0000 | 0.7851 | 0.3097 | 0.9847 |
| 1-Uniform +Lognormal | 0.8208 | 0.9392 | 0.2088 | 0.6903 | 0.5900 | 0.8931 |
| Sym-Triangular +Exponential | 0.8087 | 0.9246 | 0.1501 | 0.5356 | 0.1745 | 0.9998 |
| Sym-Triangular +Erlang2 | 0.7385 | 0.8132 | 0.0014 | 0.8528 | 0.2229 | 0.9998 |
| Sym-Triangular +Hypoexponential2 | 0.5375 | 0.1526 | 0.0000 | 0.7240 | 0.3303 | 0.9847 |
| Sym-Triangular +Lognormal | 0.8208 | 0.8932 | 0.0226 | 0.6785 | 0.2684 | 0.8931 |
| 2-Uniform +Exponential | 0.5094 | 0.0000 | 0.0047 | 0.0001 | 0.0229 | 0.9847 |
| 2-Uniform +Erlang2 | 0.5094 | 0.1261 | 0.2455 | 0.7415 | 0.2857 | 0.9847 |
| 2-Triangular +Exponential | 0.5375 | 0.2790 | 0.0000 | 0.7053 | 0.2551 | 0.0000 |
| 2-Triangular +Erlang2 | 0.0000 | 0.0013 | 0.0000 | 0.0043 | 0.1594 | 0.0000 |

For each year's time gap data set, we set $\alpha = 0.01$ and use the matching moments method in the filtering algorithm to find the best-fit distribution in each iteration. Table 3.8 shows the size of the samples after the filtering algorithm. As we can see, the number of data points filtered in each data set is between 1 and 4. Table 3.9 shows the p-value in the Kolmogorov-Smirnov test for each model and sample year and Table 3.10 shows this for the Chi-Square test. We can see that the best fitted test results are better than previous test results for the samples without the filtering.

Table 3.8: Number of Observations in the Samples After filtering

| Sample | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 |
|---|---|---|---|---|---|---|
| Number of Observations | 19 | 77 | 79 | 106 | 63 | 18 |

Based on the filtered sample data, we use the matching CDF method to model the distributions again. The p-values for the Kolmogorov-Smirnov test and Chi-Square test are listed in Table 3.11 and Table 3.12. We can see that the p-values for the best fitted distributions are larger than the previous test results from using the original time gaps data. Based on these results, we can see that the 1-parameter Uniform distribution is the best fitted distribution for modeling the turnover time.

Table 3.9: P-value for Kolmogorov-Smirnov Test for 12 Different Models by Matching Moments after Filtering

| Model | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 |
|---|---|---|---|---|---|---|
| 1-Unif. +Expon. | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 |
| 1-Unif. +Erlang2 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 |
| 1-Unif. + Hypoexpon.2 | >=0.20 | <0.01 | <0.01 | >=0.20 | <0.01 | >=0.20 |
| 1-Unif. +Lognorm. | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 |
| Sym-Triang. +Expon. | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 |
| Sym-Triang. +Erlang2 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 |
| Sym-Triang. +Hypoexpon.2 | >=0.20 | <0.01 | <0.01 | >=0.20 | <0.01 | >=0.20 |
| Sym-Triang. +Lognorm. | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 |
| 2-Unif. +Expon. | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 |
| 2-Unif. +Erlang2 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 |
| 2-Triang. +Expon. | >=0.20 | <0.01 | <0.01 | >=0.20 | <0.01 | <0.01 |
| 2-Triang. +Erlang2 | >=0.20 | <0.01 | <0.01 | <0.01 | <0.01 | <0.01 |

Table 3.10: P-value for Chi-Square Test for 12 Different Models by Matching Moments after Filtering

| Model | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 |
|---|---|---|---|---|---|---|
| 1-Uniform+Exponential | 0.9801 | 0.6337 | 0.2114 | 0.7813 | 0.9974 | 0.9921 |
| 1-Uniform +Erlang2 | 0.9991 | 0.9724 | 0.0379 | 0.8752 | 0.9742 | 0.9185 |
| 1-Uniform + Hypoexponential2 | 0.0000 | 0.0000 | 0.0000 | 0.9089 | 0.0000 | 0.0000 |
| 1-Uniform +Lognormal | 0.9991 | 0.8133 | 0.0320 | 0.6889 | 0.9969 | 0.9847 |
| Sym-Triangular +Exponential | 0.9999 | 0.9533 | 0.1719 | 0.4767 | 0.3760 | 0.9047 |
| Sym-Triangular +Erlang2 | 0.9981 | 0.9440 | 0.0034 | 0.8839 | 0.1879 | 0.9480 |
| Sym-Triangular +Hypoexponential2 | 0.0000 | 0.0000 | 0.0000 | 0.8839 | 0.0000 | 0.0000 |
| Sym-Triangular +Lognormal | 0.9999 | 0.9521 | 0.0690 | 0.6889 | 0.0385 | 0.9480 |
| 2-Uniform +Exponential | 0.9991 | 0.9236 | 0.4100 | 0.9359 | 0.9969 | 0.9566 |
| 2-Uniform +Erlang2 | 0.9991 | 0.9759 | 0.4156 | 0.8595 | 0.9969 | 0.8685 |
| 2-Triangular +Exponential | 0.0000 | 0.0000 | 0.0000 | 0.9227 | 0.0000 | 0.0000 |
| 2-Triangular +Erlang2 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

## 3.2 Procedure Duration

The importance of accurately estimating/describing the procedure duration in OR scheduling is self-evident. In this section, we discuss the distribution, the effects of learning curve and

Table 3.11: P-value for Kolmogorov-Smirnov Test for 12 Different Models by Matching CDF Using the Filtered Samples

| Model | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 |
|---|---|---|---|---|---|---|
| 1-Unif. +Expon. | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 |
| 1-Unif. +Erlang2 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 |
| 1-Unif. + Hypoexpon.2 | <0.01 | <0.01 | <0.01 | >=0.20 | <0.01 | <0.01 |
| 1-Unif. +Lognorm. | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 |
| Sym-Triang. +Expon. | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 |
| Sym-Triang. +Erlang2 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 |
| Sym-Triang. +Hypoexpon.2 | <0.01 | <0.01 | <0.01 | >=0.20 | <0.01 | <0.01 |
| Sym-Triang. +Lognorm. | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 |
| 2-Unif. +Expon. | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 |
| 2-Unif. +Erlang2 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 | >=0.20 |
| 2-Triang. +Expon. | <0.01 | <0.01 | <0.01 | >=0.20 | <0.01 | <0.01 |
| 2-Triang. +Erlang2 | <0.01 | <0.01 | <0.01 | <0.01 | <0.01 | <0.01 |

Table 3.12: P-value for Chi-Square Test for 12 Different Models by Matching CDF Using the Filtered Samples

| Model | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 |
|---|---|---|---|---|---|---|
| 1-Uniform +Exponential | 0.9991 | 0.8787 | 0.6414 | 0.9551 | 0.9914 | 0.9875 |
| 1-Uniform +Erlang2 | 0.9991 | 0.9908 | 0.4395 | 0.9081 | 0.9867 | 0.9875 |
| 1-Uniform + Hypoexponential2 | 0.0000 | 0.0000 | 0.0000 | 0.9012 | 0.0000 | 0.0000 |
| 1-Uniform +Lognormal | 0.9991 | 0.9948 | 0.0180 | 0.6120 | 0.9969 | 0.9047 |
| Sym-Triangular +Exponential | 0.9981 | 0.9489 | 0.1719 | 0.7862 | 0.8413 | 0.9875 |
| Sym-Triangular +Erlang2 | 0.9981 | 0.9797 | 0.0024 | 0.9427 | 0.8116 | 0.9875 |
| Sym-Triangular +Hypoexponential2 | 0.0000 | 0.0000 | 0.0000 | 0.8752 | 0.0000 | 0.0000 |
| Sym-Triangular +Lognormal | 0.9991 | 0.9799 | 0.0252 | 0.6120 | 0.6761 | 0.9480 |
| 2-Uniform +Exponential | 0.9981 | 0.9769 | 0.0047 | 0.7813 | 0.9969 | 0.9047 |
| 2-Uniform +Erlang2 | 0.9991 | 0.9872 | 0.2455 | 0.8839 | 0.9969 | 0.9480 |
| 2-Triangular +Exponential | 0.0000 | 0.0000 | 0.0000 | 0.9556 | 0.0000 | 0.0000 |
| 2-Triangular +Erlang2 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

economic effect of procedure duration.

### 3.2.1 Distribution of Procedure Duration

Many previous researchers have adopted well-known distributions ([Dexter et al., 1999]; [Marcon et al., 2003]; [May et al., 2000]; [Paoletti and Marty, 2007]; [Persson and Persson, 2010]) with the lognormal distribution being of particular interest. Since this effort is scenario based, the suitability for our case needs to be tested.

Six months of historical data (Aug. 15th 2009 - Feb. 10th 2010) from which procedure durations was collected. The data was first classified according to service type. For each service type, the Chi-Square test and Kolmogorov-Smirnov test were used to test the goodness of fit of some well-known distributions. We found little commonality between the best-fit distributions for each service. Table 3.13 shows the test results, which includes the best-fit distribution, p-values for Chi-Square test, p-value for Kolmogorov-Smirnov test, and the number of observations for each service.

Table 3.13: Best-fit Distribution for the Procedure Duration of Several Services

| Service | Distribution | Expression | Chi-Square test p-value | K-S test p-value | No. of OBS |
|---|---|---|---|---|---|
| Plastic | Hypo-exp. | 14+HYPO(18.02, 43.13) | 0.321 | >0.20 | 114 |
| Orthopedics | Beta | 59+266BETA(1.45,1.69) | 0.005 | 0.1 | 202 |
| General | Weibull | 28+WEIB(103, 1.25) | 0.242 | >0.15 | 164 |
| Vascular | Hypo-exp. | 42+HYPO(26, 77, 26) | 0.598 | >0.20 | 88 |
| Urology | Gamma | 25+GAMM(79.9, 1.26) | 0.009 | 0.126 | 163 |
| Neurology | Erlang | 25+ERLA(57.6, 2) | 0.455 | >0.15 | 143 |
| Ophthalmology | Weibull | 19+WEIB(61,1.48) | >0.75 | >0.15 | 303 |
| OHNS | Gamma | 25+GAMM(48.7, 1.48) | 0.023 | >0.15 | 108 |

Then we look at the 4.5 years of orthopedics data. The data was first categorized by their CPT code, because the categorization did reduce the coefficients of variation (CV), making the description and estimation of procedure durations more accurate. Table 3.14 lists the number of observations, mean, standard deviation, and CV for the procedure duration of all orthopedic procedures, and that of three orthopedic procedures with most observations.

[1] CPT 27447: Arthroplasty, knee, condyle and plateau. medial AND lateral compartments with or without patella resurfacing (total knee arthroplasty)

[2] CPT 27130: Arthroplasty, acetabular and proximal femoral prosthetic replacement (total hip arthroplasty), with or without autograft or allograft

[3] CPT 29881: Arthroscopy, knee, surgical; with meniscectomy (medial OR lateral, including any meniscal

Table 3.14: Some Statistics about Procedure Durations in Orthopedics

| Procedure Type | No. of Observations | Mean | Standard Deviation | CV |
|---|---|---|---|---|
| Orthopedics | 2853 | 122.97 | 70.22 | 0.57 |
| 27447[1] | 369 | 153.51 | 36.61 | 0.24 |
| 27130[2] | 259 | 162.07 | 42.16 | 0.26 |
| 29881[3] | 117 | 64.49 | 28.04 | 0.43 |

For some types of procedures (identified by the CPT code) with enough data, the tests were run and similar results were obtained (Table 3.15 and Figure 3.7). The results show that some well-known distributions do not work well in our case with the best-fit distributions vary. More importantly, only 6 out of the 590 different procedures have more than 50 observations, which means that if we use some well-known distribution to describe the rest procedures, the results could be quite biased. So the empirical distribution appears to be a better choice.

Table 3.15: Best-fit Distribution for the Procedure Duration of Several Orthopedics Procedures

| Procedure CPT | Distribution | Expression | Chi-Square test p-value | K-S test p-value | No. of OBS |
|---|---|---|---|---|---|
| 27447 | Normal | NORM(154, 36.6) | <0.005 | 0.020 | 369 |
| 27130 | Beta | 85+325 BETA(2.31, 7.44) | 0.052 | >0.15 | 259 |
| 29881 | LogNormal | LOGN(4.08, 0.41) | 0.001 | 0.097 | 117 |

### 3.2.2 The Learning Curve of Residents

As for estimating the procedure duration, we start our study with the intuition that the experience level of the surgeons has an obvious impact on procedure durations. According to the opinion of the schedule nurse in the hospital, there is a cycle starting with the arrival of a new class of residents, which usually starts in July. The procedure takes less time during the first several months since the attending physicians do all the work while the residents observe. Later, the procedure takes more time as the resident starts to do part of the procedures. As the residents become more experienced, it takes less and less time to complete a procedure.

shaving)

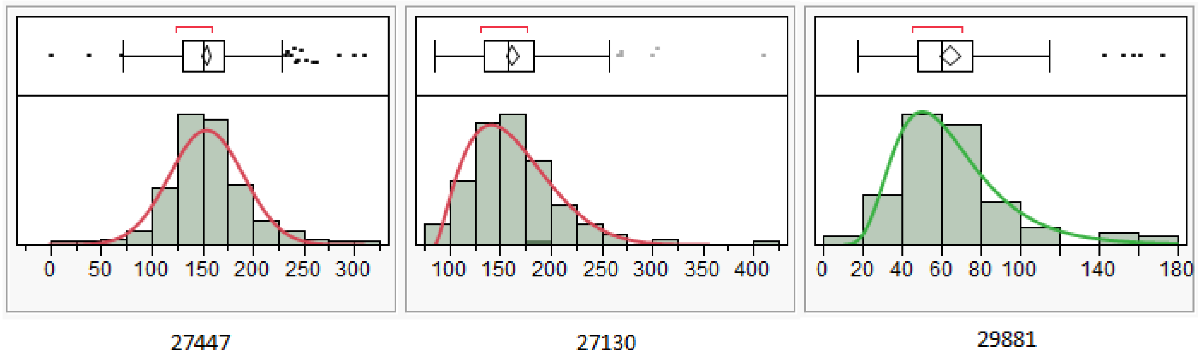



Figure 3.7: Histogram and Fitted Curve for Procedure Duration of Procedures with CPT Code 27447, 27130 and 29881

So, there should be a seasonality of procedure durations based on the physician's experience. However, our analysis results show that such seasonality is not that significant and consistent. Figure 3.8 and 3.9 show the average procedure duration each year and the average procedure duration throughout the whole 4.5 years for two of the most frequently performed procedures in the VA hospital. We can see that for procedure 66984[4], there are trend and seasonality patterns in some years, but not all years, and such patterns did not show up in the case for procedure 27447. The standard deviations of the procedure durations for both 66984 and 27447 are over 30, so one of the reasons for not observing the learning curve could potentially comes from the high variation of the procedure duration.

### 3.2.3 Economic Effects

The economic effects are another interesting issue we should consider in estimating the procedure duration. If we over-estimate the procedure durations in making OR schedules, we may expect more OR idle time because a procedure would finish earlier than expected and the resources needed for the following procedure may not be ready based on the schedule. If we under-estimate the procedure duration, we may expect more surgical team's waiting time because a procedure would finish later than expected so that the surgical team for the following procedure has to wait. So we should take the OR idle time cost and surgical team's waiting time cost into consideration for estimating procedure duration. If the OR idle time costs less, we would want any error to favor an over estimate, and if the surgical team's waiting time costs less, we would want any error to favor an under estimate.

Assume the OR idle time unit cost is $c_i$,and the procedure team's waiting time unit cost is

[4]CPT 66984: Extracapsular cataract removal with insertion of intraocular lens prosthesis (1 stage procedure), manual or mechanical technique (eg, irrigation and aspiration or phacoemulsification)

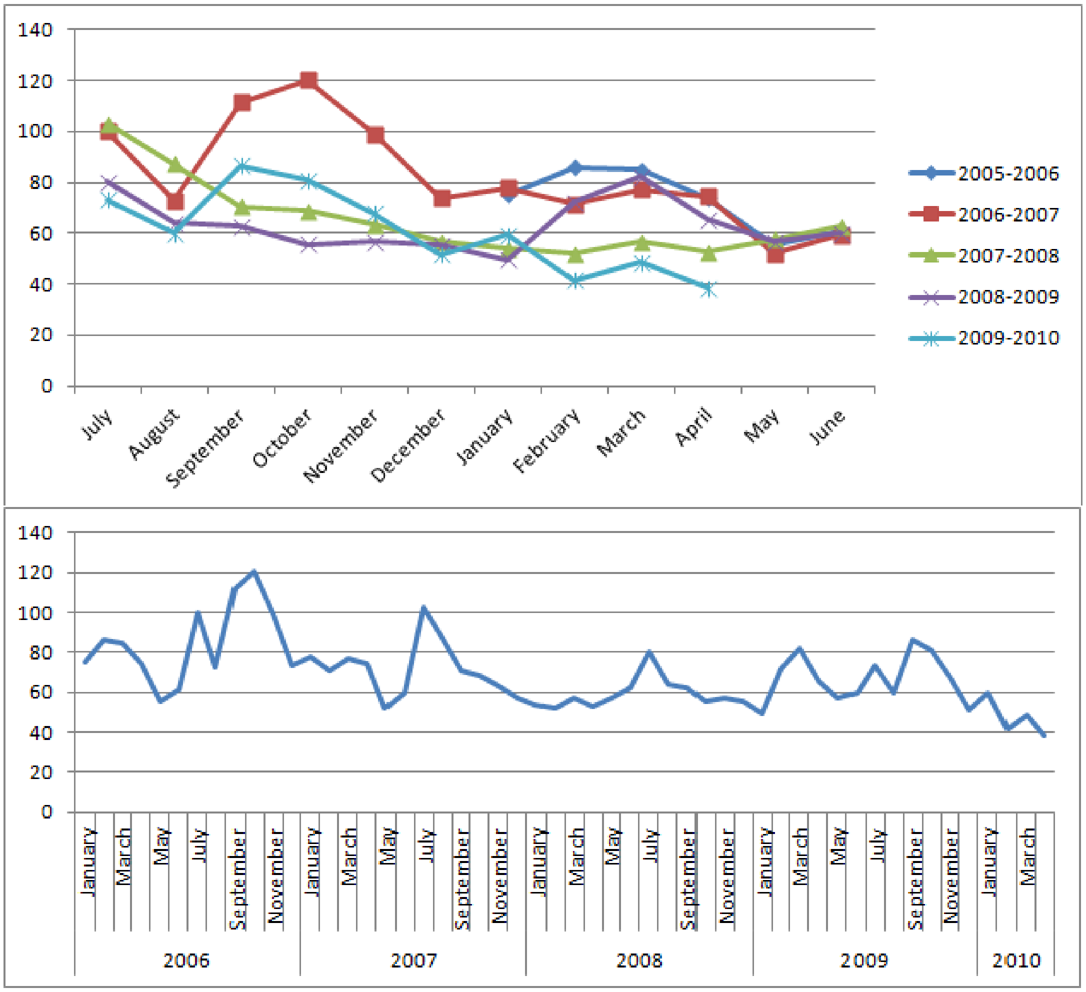



Figure 3.8: Average Procedure Duration Each Year for Procedures with CPT Code 66984 Based on 2238 Observations

$c_w$. We use $D$ to denote the random variable procedure duration with density function $f(D)$, $x^*$ to denote the optimal estimated procedure duration, and $e$ to denote the time length earlier than the scheduled procedure start time that all resources are ready, for example, if we only consider patients' arrival, then it would be the amount of time the patient is required to arrive and be ready prior to their scheduled procedure start time. So the expected total idle time cost and waiting time cost would be:

$$\begin{aligned} total\ cost &= c_i E[(x^* - e - D)^+] + c_w E[(D - x^*)^+] \\ &= c_i \int_0^{x^*-e} (x^* - e - D) f(D) dD + C_w \int_{x^*}^{\infty} (D - x^*) f(D) dD. \end{aligned} \tag{3.1}$$

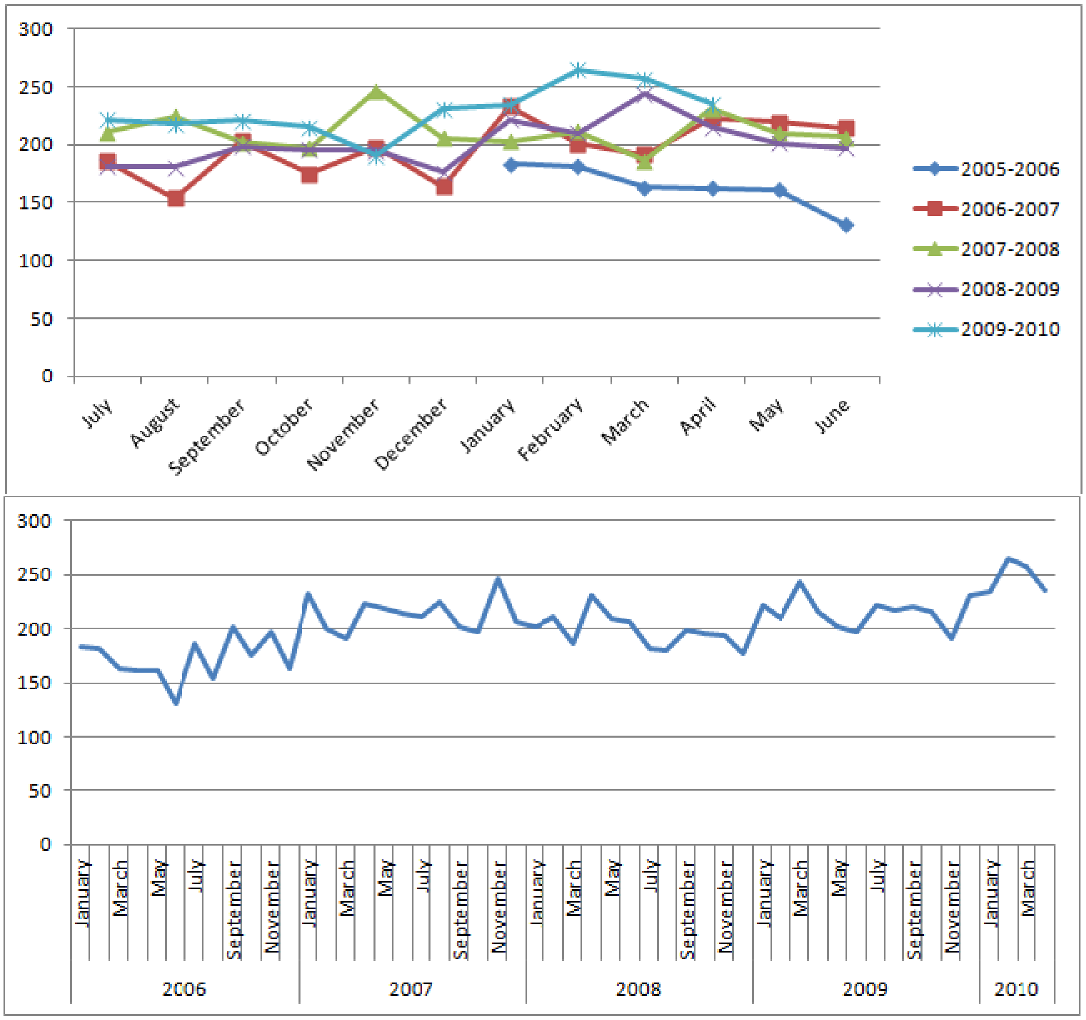



Figure 3.9: Average Procedure Duration Each Year for Procedures with CPT Code 27447 Based on 331 Observations

To find $x^*$ that minimizes the total cost, we take the derivative of the total cost function with respect to $x^*$ and set it to be zero, thus $x^*$ should satisfy the following equation:

$$c_i P\{D \leq x^* - e\} = c_w P\{D \geq x^*\} \tag{3.2}$$

An extreme case would be assuming that all procedures start no earlier than the scheduled start time, i.e., $e = 0$, then we can simplify equation 3.2 to obtain the following equation:

$$P\{D \le x^*\} = \frac{c_w}{c_i + c_w} \tag{3.3}$$

## 3.3 Length of Stay in PACU and ICU

The PACU and the ICU are important and expensive resources, and should also be considered in the OR scheduling process. The availability of the PACU and the ICU impact the effectiveness of the OR schedule. So modeling the distributions of LoS in the PACU and ICU is of interest. Based on our data analysis results, the LoS in PACU and ICU vary greatly, even for the same procedure, because it may be also very sensitive to the patient's physical condition and other factors. Table 3.16 shows some statistics about the LoS in PACU for all orthopedics procedures and two most commonly performed procedures. You can see that there is not much difference between statistics of these three data sources. Given this insignificant difference and the limited observations, we think that it is more robust to pool the LoS of each service type together, i.e., assume that the LoS in PACU/ICU for the same service follows one distribution, and can be estimated together.

Table 3.16: Some Statistics about LoS in PACU for Orthopedics

| Procedure Type | No. of Observations | Mean | Standard Deviation | CV |
|---|---|---|---|---|
| Orthopedics | 142 | 144.84 | 135.31 | 0.93 |
| 27447 | 29 | 145.90 | 133.19 | 0.92 |
| 27130 | 22 | 147.95 | 139.02 | 0.94 |

One would expect that there should be a correlation between the LoS in PACU, LoS in ICU and the procedure duration. For example, the patients who took longer in the OR (e.g. caused by complications during the surgery) may tend to take longer to recover. However, this issue has not been well addressed in previous literature. Figure 3.10 shows a weak correlation between procedure duration and LoS in the PACU based on our historical data. We also picked two most commonly recorded procedures in orthopedics, procedures with CPT code 27447 and 27130, and the correlation between the procedure duration and LoS in PACU for these procedures are 0.06 and 0.24, respectively. As for the correlation between procedure duration and LoS in the ICU, our historical data shows a correlation of 0.3, so it is hard to determine whether or not it is noise. If we assume that such correlations exist, it is not suitable to use independent distributions to simulate the three processes. The Norta method [Cario and Nelson, 1997] could be adopted to handle the problem.

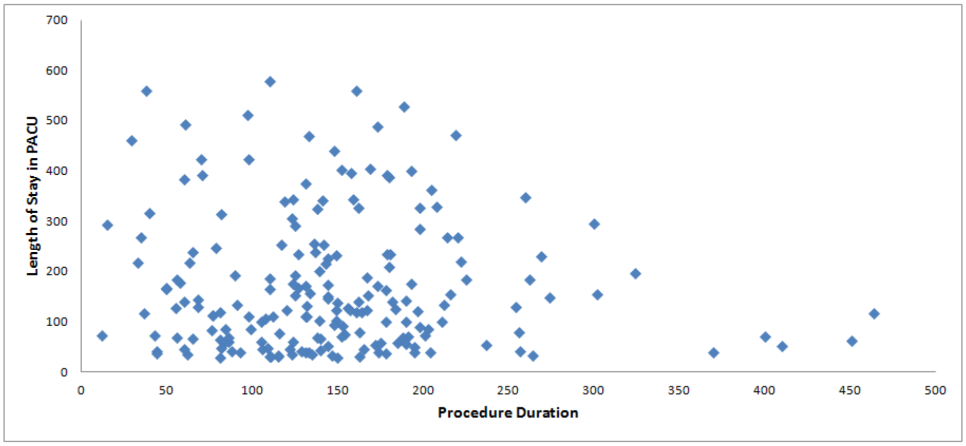



Figure 3.10: Correlation between Procedure Duration and LoS in the PACU

# CHAPTER 4

# SIMULATION

With a better analysis of the data, a more accurate simulation model can be built to study the system as a whole. It allows us to study how good a schedule is. It also allows us to test some "what if" scenarios to determine their impact before implementation. For example, how many extra procedures will the hospital be able to perform if the standard operating hours of all ORs are extended to 10 hours? What if additional operating hours are provided during the weekends instead?

In Section 4.1, we describe the conceptual simulation model. In Section 4.2, we first use some deterministic examples taken from the historical data to verify the model; and then we show the related numerical experiments on the simulation model to study the number of replications required for desired measurement accuracy, and last we use some historical data to validate the model. Some other numerical tests are also shown in Section 5.3 with the numerical tests for generating schedules. The simulation model is used in either evaluating a schedule, or as a method for answering "what if" questions. We describe how to use the model for answering "what if" questions in Section 4.3, and describe how to evaluate schedules in Section 4.4. How to evoke the simulation in the OR Scheduling Support Information system is described in Chapter 6 with the description of the system design. Some parameters in the simulation process, e.g. total number of replications in the simulation, can be changed by the user interface as well.

First, it is necessary to define some common terms we use in the following sections:

*Overtime associated with a procedure* is the OR overtime associated with a procedure, which is computed as the positive difference between the finish time of the procedure and the end time of the standard operating hours of the OR/block.

*Waiting time associated with a procedure* is the surgical team's waiting time associated with a procedure, which is computed as the positive difference between the actual start time and the scheduled start time of the procedure.

*Idle time associated with a procedure* is the OR idle time associated with a procedure, which is computed as the positive difference between the actual start time of the next procedure and the summation of the finish time of the procedure and the turnover time after the procedure,.

*Workload of an OR/block* is the total time that the OR/block is actually in use. It is the summation of procedure durations and turnover times.

*Utilization of an OR/block* is the workload of the OR/block divided by the standard operating hours of the OR/block.

*Total overtime of an OR/block* is the summation of the overtime associated procedures performed in the OR/block.

*Total waiting time of an OR/block* is the summation of the waiting time associated procedures performed in the OR/block. It is also considered as the waiting time of all surgical teams that performed procedures in the OR/block.

*Total idle time of an OR/block* is the summation of the idle time associated procedures performed in the OR/block.

## 4.1 Conceptual Model

A Monte-Carlo simulation model is adopted here. The conceptual simulation model is shown in Figure 4.1. The objective is to replicate the hospital system described in Chapter 1 by considering three resources: OR, PACU, and surgical team, and three processes: turnover, performing surgical procedure and recovering in PACU. The details are described as follows.

- **Workflow.** A patient is sent to OR for a surgical procedure after arrival if both the OR and surgical team are available. Depending on the case requirement, he/she may be sent to a PACU bed after the procedure is done, or depart (be sent to ICU or ward, or go home directly). We do not consider the process of patients going to ICU or ward in our model because both ICU and Ward beds are not the bottleneck resources in the case we study, and they have little impact in the scheduling decision making process. After the patient leaves the OR, the turnover process begins immediately. The OR becomes available to the next patient after the turnover is finished and if it is within the regular OR operating time range. The standard operating time range (e.g. from 8:00AM to 4:00PM) defines the time range that a new patient can be brought into the OR. If the turnover is finished after the time range, the next patient cannot be brought into the OR for the next procedure, which means that the next procedure will be canceled.

- **Resources.** The resources we consider include OR, surgical team, and PACU beds. We assume that there is only one PACU bed available for each OR, which means that if the PACU is occupied, the patient has to wait in the OR after surgery. We limit the capacity of each resource to be one.

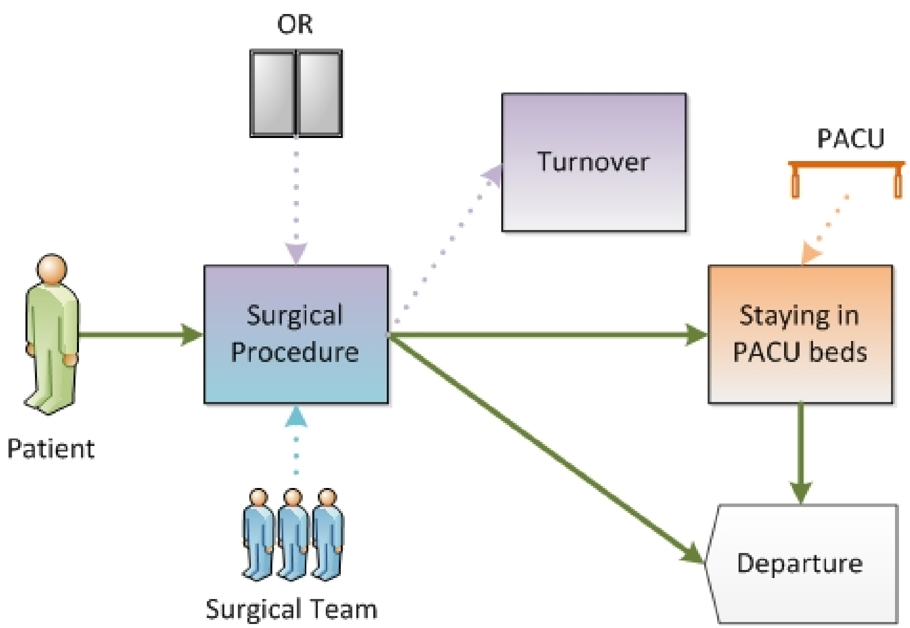



Figure 4.1: Simulation Process Illustration

- **Input.** The input to the model contains both the availability of the resources, which includes the standard operating time of the OR and the earliest available time for each surgical team of interest, and a schedule for the schedule block, which includes a list of procedures scheduled to be performed on that particular day, the CPT code for each procedure, the performing order of these procedures, the scheduled start time for each procedure, a flag indicating whether the patient needs a PACU bed after the surgical procedure, and surgical team that performs each procedure. For example, the regular operating hour of the OR is from 8:00AM to 4:00PM; all surgeons' earliest available times are 9:00AM; a schedule with two cases (CPT code 27447 and 27130) are also given with scheduled start time 8:00AM and 1:00PM, respectively; the 27447 case starts first, it requires a PACU bed, and surgical team A performs it; the 27130 case is the second, it does not require a PACU bed, and surgical team B performs it.

- **Arrival Time.** Patients' arrival time is assumed to be dependent on the scheduled start time for each procedure (e.g. the patient is available for surgical procedure 60 minutes before the scheduled start time.). This is because hospitals usually have some policy of requiring patients arrive some time (60 minutes for example) before the surgery, and we assume that all patients follow this requirement. How long the patients are assumed to be arrived before the scheduled start time are considered to be the same across all the procedures. For example, if the scheduled start time for one procedure is 12:00PM, and arriving 60 minutes before schedule is required, the patient arrival time would be 11:00AM.

- **Process Time.** We adopt the data analysis results we obtain in the previous chapter to describe the time for each process: the surgical procedure duration is assumed to follow an Empirical distribution based on all the historical procedure duration data for ones with the same CPT code, and if there is no historical data for some procedure, the Empirical distribution based on all historical data for the corresponding service is used. The turnover time follows the best-fit distribution found based on the method shown in Section 3.1.3, in the case of orthopedics service in our study, the best-fit distribution for turnover time is a Uniform distribution. The LoS in PACU are generated based on an Empirical distribution dependent on the service type.

- **Output.** Statistics including expected OR utilizations, expected total overtime, expected total waiting time, expected total idle time of the OR are the key performance measurements we are interested in.

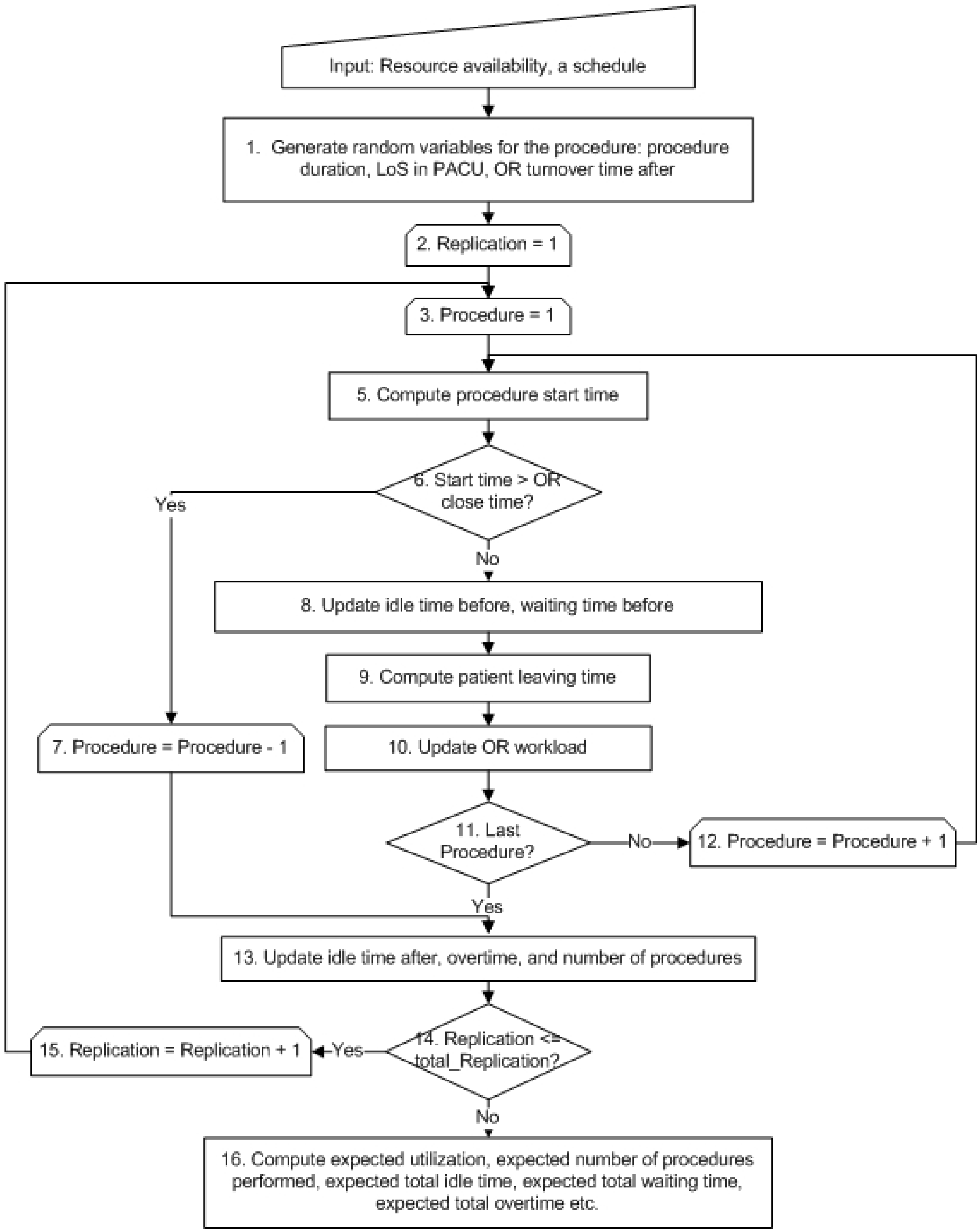



Figure 4.2: Main Simulation Process Flowchart

The flow chart for this main simulation model is shown in Figure 4.2.

- **Input.** Resource availability information and a schedule are given as the input of the simulation process.
- **Step 1.** We generate procedure durations, the LoS in PACU (if PACU bed is required) for each case, and turnover time after each procedure for all the replications.
- **Step 2.** Initialize Replication = 1.
- **Step 3.** Initialize Procedure = 1.
- **Step 5.** Compute procedure start time.
  - Procedure start time = $max\{$ OR available time, surgical team available time, patient's arrival time $\}$;
  - OR available time = $max\{$ Previous Patient Leaving OR Time + Turnover Time after Previous Procedure, OR open time $\}$.
- **Step 6.** Check whether the procedure start time computed in Step 5 exceeds the OR standard operating time range. If it does not exceed the time range, which means that the patient can be brought into the OR, continue to Step 8; Otherwise, the corresponding procedure has to be canceled, go to Step 7.
- **Step 7.** Moved the procedure index back to the previous one (i.e. the last procedure performs) for computing measurements in Step 13.
- **Step 8.** Update Total Idle Time and Total Waiting Time of the block.
  - Idle Time before = Current Procedure Start Time$-$Finish Time of Previous Procedure $-$ Turnover Time after Previous Procedure;
  - Waiting Time before = $max\{$ 0, Current Procedure Scheduled Start Time $-$ Current Procedure Start Time $\}$;
  - Total Idle Time = Total Idle Time + Idle Time before;
  - Total Waiting Time = Total Waiting Time + Waiting Time before.
- **Step 9.** Compute current patient leaving OR time.
  - If PACU bed is required:
    * PACU Available Time = Last PACU Required Patient Leaving OR Time + Last PACU Required Patient LoS in PACU;

    * Patient Leaving OR Time = $max\{$ Procedure Start Time + Procedure Duration, PACU Available Time $\}$.
  - Else:
    * Patient Leaving OR Time = Procedure Start Time + Procedure Duration.

- **Step 10.** Update Total Workload.
  - Total Workload = Total Workload + Current Procedure Duration + Turnover Time after Current Procedure.

- **Step 11.** Check whether the current procedure is the last procedure in the schedule list. If it is the last procedure, which means that we have gone through all procedures in the current replication, go to Step 13; Otherwise, go to Step 12 to continue to the next procedure.

- **Step 12.** Increment procedure index. Go to Step 5.

- **Step 13.** Update the total idle time or the total overtime of the block.
  - Idle Time after = $max\{$ 0, Turnover Finish Time - OR Standard Operating End Time $\}$;
  - Overtime = $max\{$ 0, Turnover Finish Time - OR Standard Operating End Time $\}$;
  - Total Idle Time = Total Idle Time + Idle Time after;
  - Total Overtime = Total Overtime + Overtime;
  - Total Number of Procedures = Total Number of Procedures + Current Procedure Index.

- **Step 14.** The current replication is complete, and check whether the current replication counter is less than the total number of replications. If the desired number of replications have been finished, go to Step 16 to compute the statistics; otherwise, go to Step 15.

- **Step 15.** Update the replication counter. Go back to Step 3 for the next replication.

- **Step 16.** Compute the desired statistics.
  - Expected Workload = Total Workload / Number of Replications;
  - Expected Utilization = Expected Workload / Length of the OR Standard Operating Time;
  - Expected Number of Procedures = Total Number of Procedures / Number of Replications;

– Expected Total Idle Time = Total Idle Time / Number of Replications;

– Expected Total Waiting Time = Total Waiting Time / Number of Replications;

– Expected Total Overtime = Total Overtime / Number of Replications;

## 4.2 Model Verification and Validation

In this section, we show some numerical experiments we conducted using the historical data for orthopedics service to verify our simulation model. The experiments are based on 6 most representative real cases occurred in March 2011, which not only contains both 2-procedure cases and 3-procedure cases, but also contain different combinations of frequently-performed procedures and rarely-performed procedures. The historical schedule for the 6 cases, and the procedure duration, turnover time and LoS in PACU for procedures occurred in the history are shown in Table 4.1.

Table 4.1: Some Examples Taken from Historical Data

| Test Case | OR Operating Time | Order | CPT | Scheduled Start Time | PACU required | Surgical Team's Earliest Available Time | Procedure Duration | Turnover Time | LoS in PACU |
|---|---|---|---|---|---|---|---|---|---|
| E1 | 8:00AM-6:00PM | 1 | 27447 | 8:00AM | No | 8:00AM | 162 | 46 | |
| | | 2 | 27447 | 12:00PM | No | 8:00AM | 201 | 30 | |
| E2 | 8:00AM-4:00PM | 1 | 27447 | 8:00AM | No | 8:00AM | 239 | 45 | |
| | | 2 | 27130 | 12:00PM | No | 8:00AM | 194 | 30 | |
| E3 | 8:00AM-6:00PM | 1 | 27447 | 8:00AM | Yes | 8:00AM | 214 | 38 | 235 |
| | | 2 | 29822[1] | 8:00AM | No | 8:00AM | 220 | 30 | |
| E4 | 8:00AM-6:00PM | 1 | 29888[2] | 8:00AM | No | 8:00AM | 320 | 21 | |
| | | 2 | 27301[3] | 11:00AM | No | 8:00AM | 123 | 30 | |
| E5 | 8:00AM-6:00PM | 1 | 27447 | 8:00AM | Yes | 8:00AM | 200 | 42 | 218 |
| | | 2 | 29881 | 11:30AM | No | 8:00AM | 103 | 25 | |
| | | 3 | 27090[4] | 12:30PM | No | 8:00AM | 143 | 30 | |
| E6 | 8:00AM-4:00PM | 1 | 27447 | 8:00AM | No | 8:00AM | 172 | 60 | |
| | | 2 | 27486[5] | 12:00PM | No | 8:00AM | 168 | 0 | |
| | | 3 | 20670[6] | 4:30PM | No | 8:00AM | 46 | 30 | |

Table 4.2: Computed Statistics for the Examples

| Test Case | Utilization | Overtime | Waiting Time | Idle Time |
|---|---|---|---|---|
| E1 | 0.732 | 0 | 0 | 161 |
| E2 | 1.058 | 28 | 44 | 0 |
| E3 | 0.837 | 0 | 252 | 98 |
| E4 | 0.823 | 0 | 161 | 106 |
| E5 | 0905 | 16 | 205 | 73 |
| E6 | 0.992 | 46 | 0 | 50 |

First, we use this historical record as a deterministic case to verify the workflow. The resource availability input of the simulation model, which includes the OR operating time and surgical team's earliest available time, and the schedule input, which includes the order and the scheduled start time of the procedures, PACU required flag, are listed in the table. We only simulate one replication with procedure duration, turnover time and LoS in PACU assumed to be constant numbers as listed in the table. Given these input, the corresponding statistics, with replication of 1, can be computed as shown in Table 4.2. Now take test case E5 as an example. Based on the time line illustrated in Figure 4.3, we can see that the start time of the second procedure (29881) is 12:02PM, and the waiting time before this procedure is the time between the scheduled start time and this actual start time, which is 32 minutes. The turnover after this procedure ends at 2:10PM, however, since the PACU bed is occupied, the patient's leaving time becomes 2:58PM. The start time for the last procedure is 3:23PM, which make the waiting time before be 173 minutes and the idle time before be 73 minutes. The last procedure ends at 5:46PM and the turnover after that finishes at 6:16PM, so the overtime is 16 minutes.

Then we include the randomness in the test and simulate the 6 test cases with 1000 replications for each. The input is the same as the one-replication test above. The procedure duration, turnover time and LoS in PACU are generated as described in Section 4.1. Table 4.3 shows the confidence interval for expected utilization, expected overtime, expected waiting time and expected idle time in each test case with significant level of 97.5%. Table 4.4 shows the corresponding relative error. These results provide us a guideline for choosing the total number of replications in the simulation in order to obtain the desired confidence for statistics of interest.

[1]CPT 29822: Arthroscopy, shoulder, surgical; debridement, limited
[2]CPT 29888: Arthroscopically aided anterior cruciate ligament repair/augmentation or reconstruction
[3]CPT 27301: Incision and drainage, deep abscess, bursa, or hematoma, thigh or knee region
[4]CPT 27090: Removal of hip prosthesis; (separate procedure)
[5]CPT 27486: Revision of total knee arthroplasty, with or without allograft; 1 component
[6]CPT 20670: Removal of implant; superficial (eg, buried wire, pin or rod) (separate procedure)

Table 4.3: 97.5 % Confidence Interval for Some Statistics in Simulation with 1000 Replications

| Test Case | Exp. Utilization | Exp. Overtime | Exp. Waiting Time | Exp. Idle Time |
|---|---|---|---|---|
| E1 | 0.786±0.007 | 0.56±0.40 | 12.97±1.73 | 129.08±3.89 |
| E2 | 1.032±0.009 | 31.23±3.20 | 14.79±1.93 | 15.91±1.88 |
| E3 | 0.719±0.007 | 0.07±0.12 | 194.59±3.21 | 168.76±4.31 |
| E4 | 0.705±0.009 | 0.41±0.29 | 74.10±2.55 | 177.18±5.17 |
| E5 | 0.934±0.012 | 49.54±4.85 | 144.59±6.76 | 89.04±7.12 |
| E6 | 1.142±0.009 | 75.69±4.55 | 13.73±1.72 | 7.56±1.31 |

Table 4.4: Relative Error for Some Statistics in Simulation with 1000 Replications

| Test Case | Exp. Utilization | Exp. Overtime | Exp. Waiting Time | Exp. Idle Time |
|---|---|---|---|---|
| E1 | 0.85% | 70.71% | 13.32% | 3.01% |
| E2 | 0.87% | 10.24% | 13.06% | 11.81% |
| E3 | 1.00% | 166.33% | 1.65% | 2.55% |
| E4 | 1.24% | 69.57% | 3.45% | 2.92% |
| E5 | 1.32% | 9.80% | 4.67% | 8.00% |
| E6 | 0.81% | 6.01% | 12.50% | 17.30% |

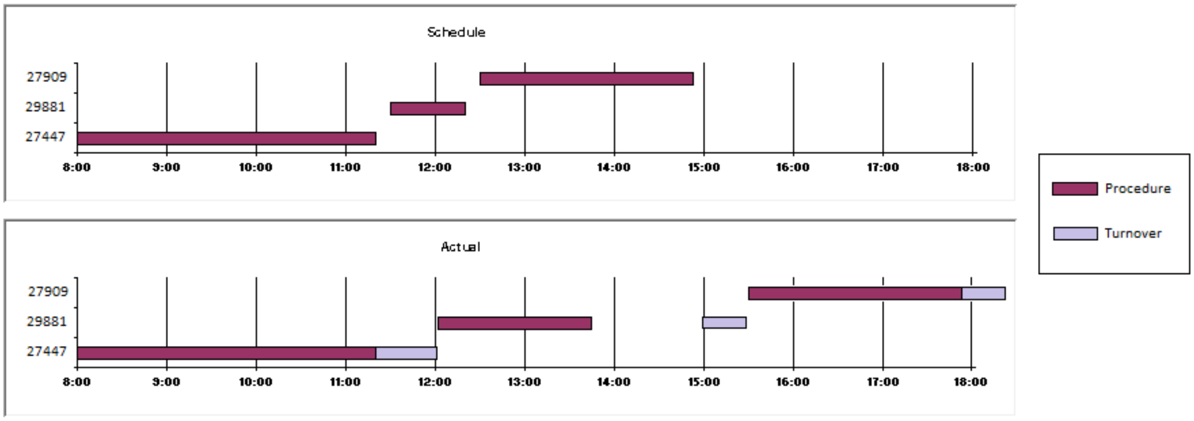



Figure 4.3: Simulation Example E5 Illustration

The next test we conduct is to use 2 months of historical data to validate the simulation model. We use the orthopedics schedule blocks from 02/01/2011 to 04/01/2011, which contains 27 10-hour blocks, 20 8-hour blocks and 1 7-hour block. The historical data from 10/01/2006 to 01/31/2011 is used to determine the distribution of the procedure durations, turnover time, and LoS in PACU. We use the schedules in the historical data as the schedule inputs. The number of replications we conduct in the simulation process for each block is 10000. We compare the expected performance measurements for each block in the simulation with the actual performance measurements. The results are listed in Table 4.5. It is noticed that the expected number of procedures performed in each block is different between the simulation run and the actual data. This is because the last procedures in 2 of 3-procedure blocks may be canceled in some simulation replications because the previous procedures are delayed and the last procedures cannot start within the OR operating time. Table 4.5 also contains the performance measurement comparison for the rest 46 blocks. The closeness of the expected performance in the simulation and actual realization shows the validation of the simulation model.

Table 4.5: Statistics Comparison between Simulation and Actual Realization

| Measurements | 48 blocks | | 46 blocks | |
|---|---|---|---|---|
| | Simulation | Actual | Simulation | Actual |
| Exp. Num Procedures | 1.46 | 1.50 | 1.37 | 1.37 |
| Exp. Utilization | 0.57 | 0.59 | 0.54 | 0.56 |
| Exp. Overtime (min) | 16.94 | 19.75 | 13.46 | 15.60 |
| Exp. Waiting Time (min) | 14.69 | 20.29 | 11.92 | 17.23 |
| Exp. Idle Time (min) | 165.28 | 156.73 | 168.99 | 160.05 |

## 4.3 "What-if" Simulation

The simulation model can be used to answer some "what-if " questions, i.e. what is the impact if some operating strategy is changed. In this section, we first describe how to conduct the "what-if" simulation based on the conceptual model introduced in Section 4.1, then we show the simulation results for two "what-if " questions.

The key to answering "what-if" questions is about how to generate the schedule inputs for the simulation model. In the case of the VA hospital, there are a large number of patients waiting for surgeries, so it is reasonable to assume that the probability of each procedure's occurrence can be approximated by the the procedure's frequency in the historical data. Based on this assumption, we can generate the list of procedures in the schedule by randomly picking procedures based on their frequency, and the sequence of the procedures are thus determined as well. The scheduled start time for each procedure is automatically generated using the method in Section 5.2. After the schedule is generated, the technique and desired results are similar to those mentioned above.

The first "what-if" scenario we test is about extending the standard operating hour of the ORs. As introduced in Chapter 1, there are eight ORs in the VA hospital. Currently, the standard operating hours for OR1 and OR2 are 10 hours per day, and eight hours per day for OR3 through OR8. So an interesting question would be *"what is the advantage of extending the standard operating hours for OR3 through OR8 to be 10 hours?"*.

The test is conducted based on the historical data for orthopedics service. The historical data from 10/01/2006 to 04/11/2011 is used to determine the distribution of the procedure durations, turnover time, etc. Based on these analysis results, we generate schedules for the 48 blocks from 02/01/2011 to 04/01/2011 that are assigned to orthopedics service, in which 27 blocks are assigned to OR1 or OR2. Surgeons are randomly assigned to procedures and are assumed to be available from 9:00AM on Wednesdays due to a weekly staff meeting, and from 8:00AM on the rest weekdays. Among the 48 blocks, there is only 1 block assigned on Wednesday, and the OR is OR3. We conduct two simulations, one uses the current OR operating hours, and another one uses the extended OR operating hours. Then we compare the expected total number of procedures performed in both simulations. The number of replications we conduct in the simulation process for each block is 10000. For details about other parameters used in the scheduling process, please refer to the first test described in Section 5.3. We repeat the test 10 times and the expected results are shown in Table 4.6, which includes the expected of procedures performed, expected utilization, expected overtime, expected waiting time and expected idle time for each block under both current operating hours and extended hours, and their differences in percentage. We can see that the expected number of procedures performed in each block has been increased by 27% with a little increase of overtime (from 4.41 minutes to 7.87 minutes). So the hospital administrators can decide whether the OR operating hours should be extended by considering both the increase in the hospital output as we showed and the cost associated with this decision.

Table 4.6: Statistics Comparison for Different OR Operating Hours

| | Current Hours | Extended Hours | % Improvement |
|---|---|---|---|
| Exp. Num Procedures | 1.51 | 1.92 | 26.72% |
| Exp. Utilization | 0.57 | 0.68 | 18.10% |
| Exp. Overtime (min) | 4.41 | 7.87 | 78.39% |
| Exp. Waiting Time (min) | 6.42 | 11.23 | 74.85% |
| Exp. Idle Time (min) | 22.96 | 41.32 | 79.94% |

The second "what-if" scenario we test is about the patient arrival requirement policy. In the other tests we show in this paper, we use a 60-minute-policy, i.e. patients are required and assumed to be arrived and be ready for surgery 60 minutes before the scheduled start time. A question of interest would be *"what is the impact if we change the 60-minute-policy to be longer or shorter?"*.

The test is conducted using the same settings and methods as the first one, except the OR operating hours are set back to the current operating hours, and the patient arrival times are adjusted to the corresponding policy. We compare a 30-minute-policy, 60-minute-policy and 90-minute-policy. The results are shown in Table 4.7, which provides a guideline on the impact of changing the policy. If we change the policy to a 30-minute-policy, the expected number of procedures performed in each block and expected utilization would be reduced by 10%; while, the advantage we gain from changing to a 90-minute-policy is 5.3%. So hospital administrators should balance between the efforts paid for changing the policy and the advantage gain from the change, to make decisions about this policy.

Table 4.7: Statistics Comparison for Different Patient Arrival Policies

| | Statistics for Policy | | | % Improvement for Policy | |
|---|---|---|---|---|---|
| | 30-min | 60-min | 90-min | 30-min | 90-min |
| Exp. Num Pro. | 1.35 | 1.51 | 1.59 | -10.88% | 5.10% |
| Exp. Utilization | 0.52 | 0.57 | 0.61 | -10.21% | 5.30% |
| Exp. Overtime | 2.90 | 4.41 | 5.98 | -34.19% | 35.53% |
| Exp. Wait Time | 5.94 | 6.42 | 4.87 | -7.47% | -24.13% |
| Exp. Idle Time | 17.50 | 22.96 | 24.29 | -23.78% | 5.81% |

## 4.4 Schedule Evaluation

The simulation model we use for schedule evaluation is an extended model from the main model shown in Section 4.1, in which both surgeons' procedure duration estimates and procedures' cancelation risk are considered. Figure 4.4 shows the flowchart, in which the extended pieces are indicated in red color.

In addition to the input we mentioned before, we also need some input about the cases from surgeons: surgeons' estimate of procedure duration in each case, their confidence level about the estimates, and surgeons' estimate about the cancelation risk for each case.

In Step 1, we combine our estimates with surgeons' to generate procedure duration for each case. This is because when we estimate the procedure duration, we look at the all historical durations for the same procedure, which are independent of patients' status; while, when surgeons estimate the procedure duration, they may also consider the details of the case, for example, the illness of this patient is more severe than usual, so it may take longer time. So combining the surgeons' estimates with our estimates would potentially improve how we describe the process. In the extended model, the input includes surgeons' estimates of the procedure duration and how confidence they are about the estimates, so when we generate procedure durations, we should also combine this estimates based on the confidence level. For example, surgeons' estimates for one case is 200 minutes and they are 20% confident of this estimate, when we generate procedure duration for this case in 10000 replications, 2000 of them are generated to be 200 minutes, and the rest 8000 are randomly generated from the historical data as we described before.

In Step 1 we also randomly generate a cancelation flag for each case in each replication based on its cancelation risk to indicate whether it is canceled. In Step 4, we check the cancelation flag: if yes, go to the next procedure; otherwise, continue to the following steps as in the main simulation model. We consider surgeons' cancelation risk estimates because surgeons are more familiar with the case, so they may have an idea of the probability that the case will have to be cancelled due to patients' status. By putting this probability as the cancelation risk input to our model, we can have a better description of what is expected to happen in the block.

The rest of the steps are the same as what we described before for the main model. Using this extended model, we can evaluate a given schedule. The output performance measures provide the users with a quantitative guideline on how good the schedule of interest is, so that they can adjust their schedule as needed.

In the OR Scheduling Support Information system, the main simulation model is used in evaluating schedules that are generated by the system (Section 5.2) using historical data, to test the schedule settings (Section 6.1.2); "what if" simulation model is used in answering some "what if" scenario questions (Section 6.1.11); and the schedule evaluation model described in this section is used to evaluate either user provided schedules, or system generated schedules for the blocks in the future or in the current day (Section 6.1.6).

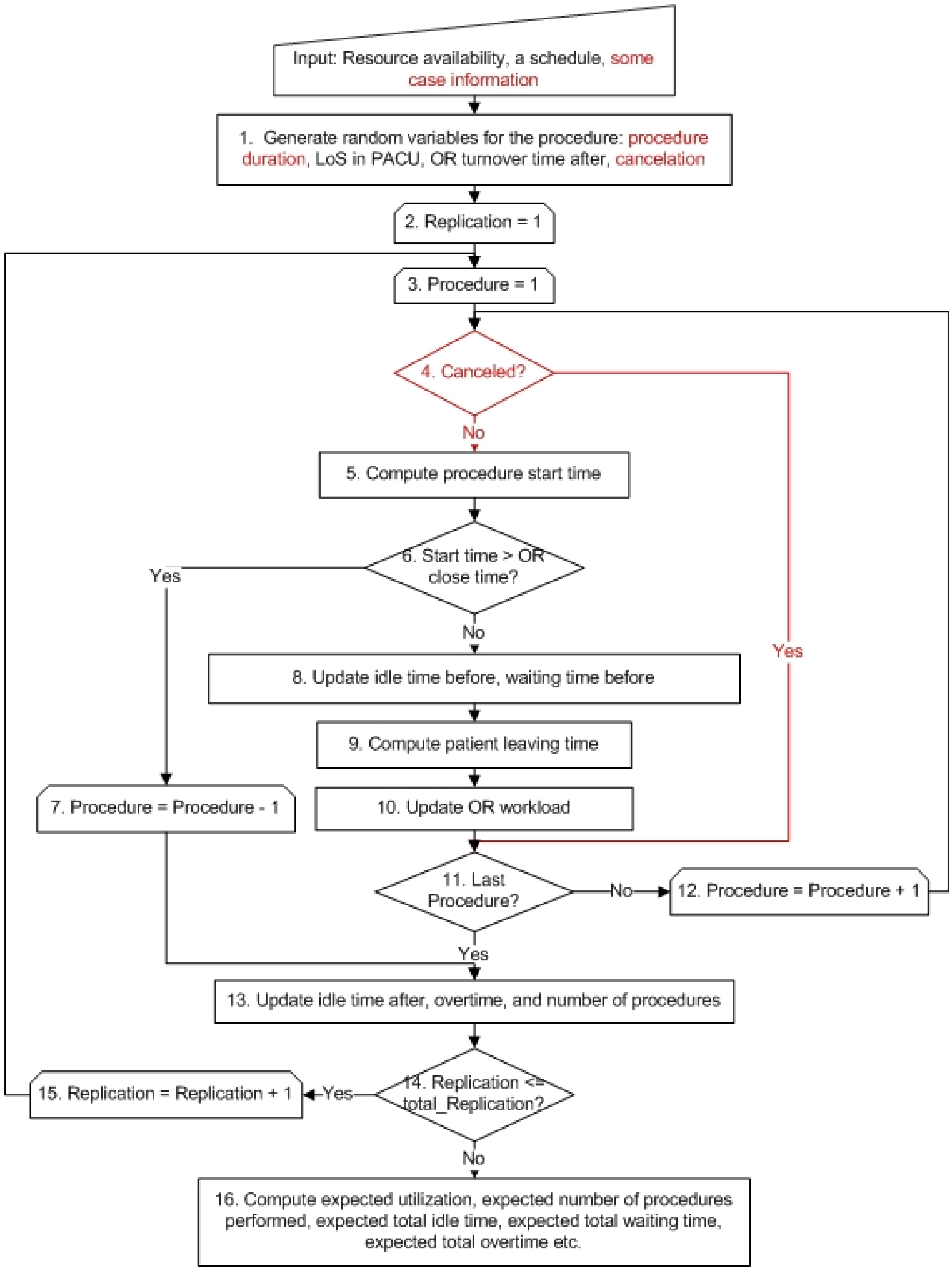



Figure 4.4: Simulation Process Flowchart for Schedule Evaluation

# CHAPTER 5

# SINGLE OR SCHEDULING PROBLEM

Generally, scheduling problems include three stages, assignment, sequencing and scheduling. In our problem, assignment refers to assigning each surgical procedure to a proper OR. Sequencing refers to arranging the order of the surgical procedures performed in an OR in one day. Scheduling refers to determining the start time for each surgical procedure. Since the service is pre-assigned to each schedule block by the hospital (Priority Schedule Table), we will focus on sequencing and scheduling in the Single-OR environment. Our objective is to provide satisfying schedules that include more surgeries by tuning the sequence and start time of the surgeries. In the following sections, we develop a new type of two-stage stochastic integer programming model, which considers many unique constants and has a unique structure. Then we discuss about how to solve the problem practically, with numerical tests results presented. At last, we introduce how to use the static model to solve dynamic scheduling problems.

## 5.1 Static Stochastic Programming Model

The surgeons provide a primary list of procedures to be performed three days in advance. These procedures together with some backup procedures form the set of procedures we are going to choose and schedule. We can give the procedures on the list higher priority if we want to try to maintain the list as much as we can.

In Figure 5.1, we illustrate the idle time, waiting time and overtime for a single OR scheduling problem. Idle time and overtime usually suggest a loose schedule, while waiting time suggests a tight schedule. There is no actual cost for these events. We can just put relative cost coefficients on the events to tune the schedule to make the planned start times more satisfying. When there is overtime, the hospital will pay extra money to the staff based on their hourly rate. The total amount of such overtime cost has to be controlled by the budget.

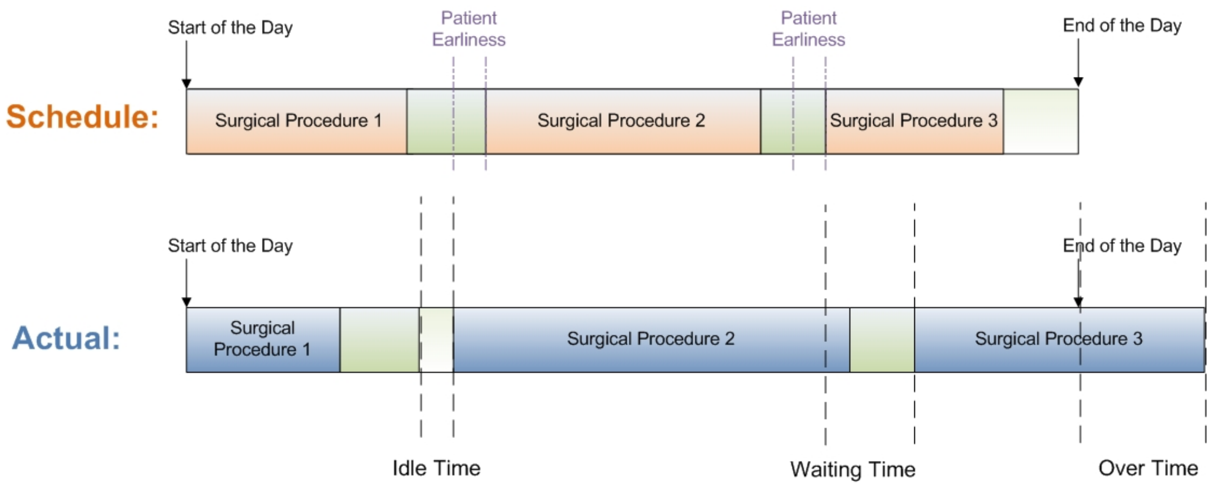



Figure 5.1: Illustration of Idle Time, Waiting Time and Overtime

[Weiss, 1990] solves small stochastic problem instances that include two or three surgeries. Their objective is to minimize the sum of the expected costs of waiting and idle time and their decision variables are the sequence of surgeries and their start time. They show that for certain selective choices of distributions the optimal solution is in order of increasing variance. However, we note that this conclusion contradicts the usual hospital procedure of scheduling longer, more variable cases first. This is because from the hospital administrators' point of view, the procedures with longer time and more variable usually represent more complicated cases, which may need more surgeon and staff in the team. So if such procedures are performed beyond the regular working hours, the overtime will affect more people than when the overtime is caused by some minor surgery. Based on this fact, we should consider a general model, in which the overtime "cost" for each surgical procedure may be different.

Unlike the other types of hospitals, VA hospitals do not profit from performing surgeries. Although they do want to do as many surgeries as they can, they have a limited budget to pay for the staff working over time. So instead of putting the overtime cost in the objective function, as seen in some previous work, we should constrain the overtime cost by the budget. The budget constraint should be a "soft" constraint instead of "hard" constraint. By a "hard" constraint, it is meant the ones that work for all scenarios. Since the tight overall budget is for a period much longer than one schedule block, there is no need to guarantee that the absolute overtime cost in each scenario be controlled by the budget; instead, we only need to constrain the expected overtime cost. This is one of the major differences between our work and that of previous literature. This difference makes our optimization model have a different structure than the commonly-seen two-stage stochastic programming model structure. We will explore the characteristic later.

Another interesting fact about overtime is that generally the case illustrated in Figure 5.2 will not happen in reality. Assume 5 procedures were scheduled, if the 4th procedure ends later than the end of regular working hour, the 5th procedure will usually be postponed to another day. We should consider this kind of cancelation when we model the scheduling problem. Another interesting fact we want to consider is the expected cancelation risk comes with this overtime case. If you schedule too many procedures in the schedule, you would expect that some of them will be canceled due to this overtime reason. Frequent occurrence of such cancelation indicates bad schedule quality because it will highly disappoint patients and affect surgeons' plans. So we can use the expected cancelation risk to control the number of procedures scheduled.

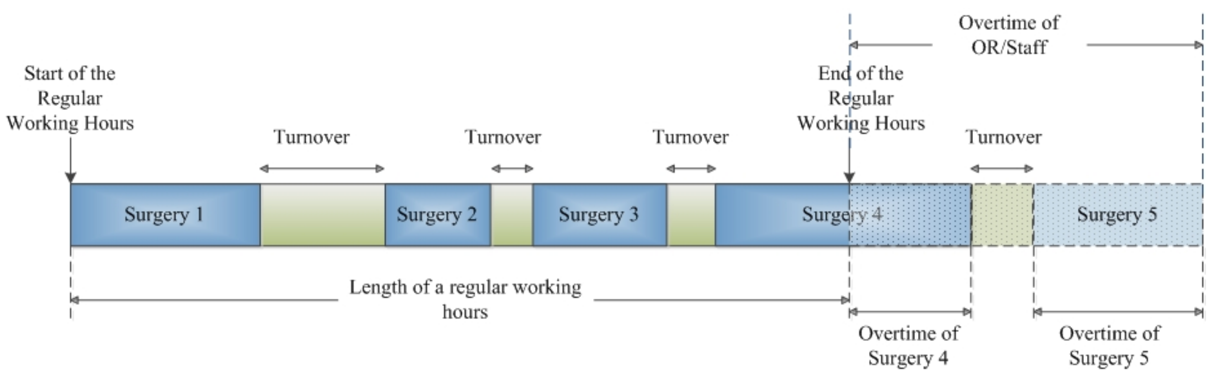



Figure 5.2: Illustration of a Case that Should Not Happen

When the procedure is done, most patients will be sent to PACU. If there is no available room in the PACU, the patient has to wait in the OR, which means that the turnover and the next procedure cannot be performed until the PACU is ready and the patient leaves. The capacity of the PACU is about the same as the capacity of the OR. To simplify the problem, we assume that there is only one PACU bed available for each OR.

In this section, we develop a two-stage stochastic integer programming model, and the model can handle all the interesting facts we describe. The first stage is about making sequencing and scheduling decisions before the surgery date, and the second stage is about what happens on the surgery date.

### 5.1.1 Notation

Table 5.1: Indices

| | | |
|---|---|---|
| $i$,$k$ | : | indices for a specific procedure, e.g. procedure $i$ |
| $j$ | : | index for the order of a procedure, e.g. the $j$th procedure |
| $\omega$ | : | index for a scenario (realization) |

Table 5.2: Parameters

| | | |
|---|---|---|
| $\alpha$ | : | parameter used to balance the objectives in the second stage problem |
| $M$ | : | a large number |
| $d$ | : | total regular open time for the block/OR |
| $cthresh$ | : | a cancelation threshold, procedures with expected cancelation risk higher than this threshold will not be scheduled |
| $early$ | : | the minimal time length required between patient getting ready and the scheduled procedure start time |
| $priority_i$ | : | priority of procedure $i$ |
| $CO_i$ | : | overtime cost coefficient for procedure $i$ |
| $CW_i$ | : | waiting time cost coefficient for procedure $i$ |
| $CI_i$ | : | idle time cost coefficient for procedure $i$ |
| $r_i$ | : | release time (earliest start time) for procedure $i$ |
| $BUDGET$ | : | expected budget for staff overtime cost |
| $staffcost_i$ | : | staff overtime cost per time unit for procedure $i$ |
| $prob_\omega$ | : | probability of scenario $\omega$ |
| $t_j(\omega)$ | : | turnover time after the $j$th procedure in scenario $\omega$ |
| $p_i(\omega)$ | : | procedure duration time for procedure $i$ in scenario $\omega$ |
| $pa_i(\omega)$ | : | length of stay in PACU for procedure i in scenario $\omega$ |
| $A_i(\omega)$ | : | $= 1$, if procedure $i$ is not canceled in scenario $\omega$; $= 0$, otherwise. |

In the parameters shown in Table 5.2, $\alpha$ is used to balance the objective of scheduling more procedures with high priority and the objective of having a high quality schedule. Within a certain range, the larger $\alpha$ is, the more important the second objective is. Some sensitivity analysis results are shown in Section 5.3.

The value of *early* defines the arrival policy for patients. For example, a hospital may ask all patients to get ready at least 30 minutes before the scheduled procedure start time. Then by setting $early = 30$, we can model the real start time of the procedure in our model to be no earlier than patients ready time, which is the scheduled start time minus 30 minutes. By changing values of this parameter and studying their effects on the solution, we may be able to come up with a better policy.

The value of $r_i$ can be used to control surgical team readiness. It is of particular interest in the VA hospital because most surgeons working in the VA hospital also work in other hospitals. In our case, the surgeons also work at the Duke University hospital across the street. So the availability of the surgical team should be considered in scheduling.

You may also notice that $t_j(\omega)$ is independent of some specific procedure $i$. This is based on our previous data analysis results for turnover time. For all $j$, the turnover time should follow the same distribution.

Table 5.3: First Stage Decision Variables

| | | |
|---|---|---|
| $x_{ij}$ | : | $= 1$, if procedure $i$ is scheduled to be the $j$th procedure to be done; |
| | : | $= 0$, otherwise. |
| $s_j$ | : | scheduled start time of the $j$th procedure |

The first stage decision variable $x_{ij}$ is the solution for the sequencing problem, and $s_j$ is the solution for the scheduling problem.

The second stage decision variable $y_{ij}(\omega)$ is introduced to avoid the case illustrated in Figure 5.2. It is used to show what happens in scenario $\omega$, instead of what is scheduled ($x_{ij}$). $AuST_{1..3,j}(\omega)$ and $AuL_{1..2,j}(\omega)$ are auxiliary variables that are used to bound the constraints for decision variables $ST_j(\omega)$ and $L_j(\omega)$.

Table 5.4: Second Stage Decision Variables

| | | |
|---|---|---|
| $O_{ij}(\omega)$ | : | = overtime for the $j$th procedure in scenario $\omega$, if procedure $i$ is the $j$th procedure performed in scenario $\omega$; |
| | : | = 0, otherwise. |
| $W_{ij}(\omega)$ | : | = waiting time for the $j$th procedure in scenario $\omega$, if procedure $i$ is the $j$th procedure performed in scenario $\omega$; |
| | : | = 0, otherwise. |
| $I_{ij}(\omega)$ | : | = Idle time for the $j$th procedure in scenario $\omega$, if procedure $i$ is the $j$th procedure performed in scenario $\omega$; |
| | : | = 0, otherwise. |
| $ST_j(\omega)$ | : | start time for the $j$th procedure in scenario $\omega$ |
| $L_j(\omega)$ | : | patient leaving OR time for the $j$th procedure in scenario $\omega$ |
| $y_{ij}(\omega)$ | : | = 1, if procedure $i$ is the $j$th procedure performed in scenario $\omega$; |
| | : | = 0, otherwise. |
| $AuST_{1..3,j}(\omega)$ | : | = 1, if the corresponding constraint for defining $ST_j(\omega)$ is tight; |
| | : | = 0, otherwise. |
| $AuL_{1..2,j}(\omega)$ | : | = 1, if the corresponding constraint for defining $L_j(\omega)$ is tight; |
| | : | = 0, otherwise. |

### 5.1.2 Two-Stage Stochastic Programming Model

The first stage problem is as follows.

$$Max \quad E_\omega[Q(x, s, \xi(\omega))] \tag{5.1}$$

$$s.t. \quad \sum_i x_{ij} \le 1, \quad \forall j, \tag{5.2}$$

$$\sum_i x_{i,j+1} \le \sum_i x_{ij}, \quad \forall j, \tag{5.3}$$

$$\sum_j x_{ij} \le 1, \quad \forall i, \tag{5.4}$$

$$\sum_i r_i x_{ij} \le s_j, \quad \forall j, \tag{5.5}$$

$$s_j \le s_{j+1}, \quad \forall j, \tag{5.6}$$

$$s_j = 30k_j, \quad \forall j, \tag{5.7}$$

$$k_j \in \{1, \ldots, 20\}, x_{i,j} \in \{0, 1\} \quad \forall i, j. \tag{5.8}$$

Constraints (5.2)-(5.4) guarantee that $x_{ij}$ shows the order of the procedure. Constraint (5.5) defines that the scheduled start time should be bounded by the release time of each procedure. Constraint (5.6) makes sure that the scheduled start time for the procedures follow the order. Constraint (5.7) says that the scheduled start times are on a half-hour time unit. This is because it makes more sense to have a procedure scheduled at 2pm or 2:30pm instead of 2:12pm. We can also adopt a 15-minute time unit by constraining the scheduled start times to be 15 times some integers.

The second stage problem: $E_\omega[Q(x,s,\xi(\omega))] =$

$$
\begin{aligned}
Max \quad & \textstyle\sum_\omega prob_\omega[\sum_i priority_i \sum_j y_{ij}(\omega) - && \\
& \alpha \textstyle\sum_i (CO_i \sum_j O_{ij}(\omega) + CW_i \sum_j W_{ij}(\omega) + CI_i \sum_j I_{ij}(\omega))] && (5.9)\\
s.t. \quad & \textstyle\sum_\omega prob_\omega(\sum_i staffcost_i \sum_j O_{ij}(\omega)) \le BUDGET, && (5.10)\\
& x_{ij} \le \frac{1}{1-cthresh} \textstyle\sum_\omega y_{ij}(\omega), & \forall i,j,\omega, \quad & (5.11)\\
& s_j - early \le ST_j(\omega), & \forall j,\omega, \quad & (5.12)\\
& \textstyle\sum_i r_i y_{ij}(\omega) \le ST_j(\omega), & \forall j,\omega, \quad & (5.13)\\
& L_j(\omega) + t_j(\omega) \textstyle\sum_i y_{i,j+1}(\omega) \le ST_{j+1}(\omega), & \forall j,\omega, \quad & (5.14)\\
& ST_j(\omega) \le s_j - early + M(1 - AuST_{1,j}(\omega)), & \forall j,\omega, \quad & (5.15)\\
& ST_j(\omega) \le \textstyle\sum_i r_i y_{ij}(\omega) + M(1 - AuST_{2,j}(\omega)), & \forall j,\omega, \quad & (5.16)\\
& ST_{j+1}(\omega) \le L_j(\omega) + t_j(\omega) \textstyle\sum_i y_{i,j+1}(\omega) + M(1 - AuST_{3,j}(\omega)), & \forall j,\omega, \quad & (5.17)\\
& AuST_{1,j}(\omega) + AuST_{2,j}(\omega) + AuST_{3,j}(\omega) = 1, & \forall j,\omega, \quad & (5.18)\\
& ST_j(\omega) + \textstyle\sum_i p_i(\omega) y_{ij}(\omega) \le L_j(\omega), & \forall j,\omega, \quad & (5.19)\\
& L_j(\omega) + \textstyle\sum_i pa_i(\omega) y_{ij}(\omega) - M(\sum_i y_{i,j}(\omega) - \sum_i y_{i,j+1}(\omega)) && \\
& \quad \le L_{j+1}(\omega), & \forall j,\omega, \quad & (5.20)\\
& L_j(\omega) \le ST_j(\omega) + \textstyle\sum_i p_i(\omega) y_{ij}(\omega) + M(1 - AuL_{1,j}(\omega)), & \forall j,\omega, \quad & (5.21)\\
& L_{j+1}(\omega) \le L_j(\omega) + \textstyle\sum_i pa_i(\omega) y_{ij}(\omega) + M(1 - AuL_{2,j}(\omega)), & \forall j,\omega, \quad & (5.22)\\
& AuL_{1,j}(\omega) + AuL_{2,j}(\omega) = 1, & \forall j,\omega, \quad & (5.23)\\
& -s_j + ST_j(\omega) \le W_{ij}(\omega) + M(1 - y_{ij}(\omega)), & \forall i,j,\omega, \quad & (5.24)\\
& s_{j+1} - early - t_j(\omega) \textstyle\sum_i y_{i,j+1}(\omega) - ST_j(\omega) - \sum_i p_i(\omega) y_{ij}(\omega) && \\
& \quad \le I_{ij}(\omega) + M(1 - y_{ij}(\omega)), & \forall i,j,\omega, \quad & (5.25)\\
& ST_j(\omega) + \textstyle\sum_i p_i(\omega) y_{ij}(\omega) - d - M(1 - \sum_i y_{ij}(\omega)) && \\
& \quad \le O_{ij}(\omega) + M(1 - y_{ij}(\omega)), & \forall i,j,\omega, \quad & (5.26)\\
& y_{ij}(\omega) \le A_i(\omega) x_{ij}, & \forall i,j,\omega, \quad & (5.27)\\
& \textstyle\sum_i y_{i,j+1}(\omega) \le \sum_i y_{ij}(\omega), & \forall j,\omega, \quad & (5.28)
\end{aligned}
$$

$$
\begin{aligned}
ST_j(\omega), E_j(\omega), L_j(\omega), W_{ij}(\omega), I_{ij}(\omega), O_{ij}(\omega) \geq 0,& \\
y_{ij}(\omega), AuST_{1..3,j}(\omega), AuL_{1..2,j}(\omega) \in \{0, 1\}, \qquad \forall j, \omega.& \qquad (5.29)
\end{aligned}
$$

In the objective function, we are trying to balance between having more procedures and having a satisfying process. Constraint (5.10) is the "soft" constraint we mentioned before to control the expected budget. Constraint (5.11) is the "soft" constraint about using the expected cancelation risk to control the number of procedures scheduled. The seven constraints, (5.12)-(5.18), define the procedure start time in each scenario. It should be

$$
ST_j(\omega) = max\{s_j - early, \sum_i r_i y_{ij}(\omega), L_{j-1}(\omega) + t_j(\omega) \sum_i y_{i,j+1}(\omega).\} \qquad (5.30)
$$

Then the five constraints, (5.19)-(5.23), describe the time the patient leaves the OR, which considers the availability of the PACU. It is equal to

$$
L_j(\omega) = max\{ST_j(\omega) + \sum_i p_i(\omega) y_{ij}(\omega), L_{j-1}(\omega) + \sum_i pa_i(\omega) y_{i,j-1}(\omega).\} \qquad (5.31)
$$

Constraints (5.24)-(5.26) define the waiting time, idle time and overtime for each procedure in each scenario. The last two constraints, (5.27)-(5.28), bound the decision variable $y_{ij}(\omega)$, which defines the order of the procedures that are performed in each scenario. It is used to rule out the case illustrated in Figure 5.2. Procedure cancelation is also considered in constraint (5.27).

Assume that there are $m$ procedures and $n$ scenarios. There are $m^2 + m$ first-stage variables, $4m^2n + 7mn$ second-stage variables, $6m$ first-stage constraints, and $5m^2n + 13mn$ second-stage constraints.

In the two-stage stochastic programming model we usually see in the literature, the subproblems generated by each scenario are independent. However, by adding the budget constraint (5.10), in which the expected cost instead of the absolute cost in each scenario is constrained, and the cancelation constraint (5.11), in which the first-stage order decision is constrained by expected cancelation risk, the the decisions for each scenario are no longer independent. This means that the L-shaped method, which is usually used to solve traditional two-stage stochastic programming problems, cannot be applied here. We introduce a new algorithm to deal with the two-stage stochastic programming problem with linking constraints in Chapter 6.

## 5.2 Practical Method for Solving the Static Scheduling Problem

The decisions we are trying to make include: whether we should include each procedure in the schedule or not, the sequence of the included procedures, and the scheduled start time based on the order for each included procedure. They form all of the first-stage decision variables. Given the values of such decision variables which satisfy the first-stage constraints, we can easily compute whether the procedure is performed or not, the procedure start time, the time patient leaves OR, the waiting time, the idle time and the over time for each procedure in each scenario. These computed results form the second-stage variables. Given the values of these second-stage variables, we can then check the linking constraints to see whether the solution is feasible or not, update the first-stage decision about whether we include each procedure in the schedule or not by comparing the expected cancelation risk with the cancelation threshold, and also compute the objective function value.

Since the number of procedures performed in each OR is limited, the size of the problem is small, especially for services like Cardiac and orthopedics. Based on the historical data for the orthopedics service in Durham VA hospital, the maximal number of procedures performed in an OR each day is 3. So one practical method for solving this problem could be doing a random sampling in the first-stage decision space, evaluate each solution by deriving the second-stage decisions and then computing the objective function value, and then pick the best among all solutions evaluated. Assume there are $n$ slots in the schedule block (e.g. for a block with operating time of 10 hours, if we adopt a 15-minute interval, the number of slots would be 40) and $m$ procedures, the size of the decision space is $\sum_{i=1}^{m} \binom{m}{i} \sum_{j=1}^{i} n!\binom{i+j-1}{j-1}\binom{n}{j}$ (Lemma 1) with a lower bound of $\frac{n^{m+1}-1}{n-1}$ (Lemma 2). If no overlapped scheduled start time is allowed, the size of the decision space would be $\sum_{i=1}^{m} \binom{m}{i} \frac{n!}{(n-i)!}$ (Lemma 3). If we add more conditions on the decision space by considering a fixed number ($\bar{m}$) of procedures included in the schedule and the sequence of these procedures are pre-determined, which is the best scenario case that may occur in the scheduling process in the hosptial, the size of the decision space would be reduced to $n!\binom{2\bar{m}-1}{\bar{m}-1}$ (Lemma 4).

**Lemma 1.** *Assume there are $n$ slots in the schedule block and $m$ procedures, then the size of the decision space is* $\sum_{i=1}^{m} \binom{m}{i} \sum_{j=1}^{i} n!\binom{i+j-1}{j-1}\binom{n}{j}$.

*Proof.* Assume we include $i$ procedures in the schedule ($1 \leq i \leq m$), since overlapped scheduled start time is allowed, we may end up choosing $j$ slots out of the total $n$ slots ($1 \leq j \leq i$). Denote the number of procedures scheduled in each chosen slot as $a_k, k = 1, \ldots, j$, then the number of possible cases for allocating the $i$ procedures into the $j$ out of $n$ slots would be

$$\sum_{\substack{a_1+\ldots+a_j=i \\ a_1,\ldots,a_j\geq 0}} a_1!a_2!\ldots a_j!\binom{n}{a_1}\binom{n-a_1}{a_2}\ldots\binom{n-\sum_{k=1}^{j-1}a_k}{a_j} = \frac{n!}{(n-i)!}\binom{i+j-1}{j-1}. \tag{5.32}$$

By considering the possible combinations in choosing $j$ out of $n$ slots, and the range of $j$, we have the number of possible cases for allocating the $i$ procedures into the $n$ slots as

$$\sum_{j=1}^{i}\frac{n!}{(n-i)!}\binom{i+j-1}{j-1}\binom{n}{j}. \tag{5.33}$$

Since $i$ could be any number chosen from 1 to $m$, the size of the decision space would be

$$\sum_{i=1}^{m}\binom{m}{i}\sum_{j=1}^{i}\frac{n!}{(n-i)!}\binom{i+j-1}{j-1}\binom{n}{j}. \tag{5.34}$$

□

**Lemma 2.** *The size of the decision space has a lower bound of* $\frac{n^{m+1}-1}{n-1}$.

*Proof.* Assume we include $i$ procedures in the schedule ($1 \leq i \leq m$). If we do not consider the sequence of the $i$ procedures, the number of possible cases for allocating the $i$ procedures into the $n$ slots with overlapping allowed would be $n^i$. Since the sequence is considered in (5.33), this $n^i$ is a lower bound for (5.33), thus $\sum_{i=1}^{m} n^i = \frac{n^{m+1}-1}{n-1}$ is a lower bound for (5.34).

□

**Lemma 3.** *If overlapped scheduled start time is not allowed, the size of the decision space is* $\sum_{i=1}^{m}\binom{m}{i}\frac{n!}{(n-i)!}$.

*Proof.* Assume we include $i$ procedures in the schedule ($1 \leq i \leq m$), since overlapped scheduled start time is not allowed, the number of possible cases for allocating the $i$ procedures into the $i$ out of $n$ distinct slots would be $\binom{n}{i}i!$. Since $i$ could be any number chosen from 1 to $m$, the size of the decision space given the no overlapping assumption would be $\sum_{i=1}^{m}\binom{m}{i}\binom{n}{i}i! = \sum_{i=1}^{m}\binom{m}{i}\frac{n!}{(n-i)!}$. □

**Lemma 4.** *Given a fixed number of procedures included in the schedule $\bar{m}$ and the pre-determined sequence of the $|\bar{m}|$ procedures, the size of the decision space is* $\frac{n!}{(n-m)!}\binom{2\bar{m}-1}{\bar{m}-1}$

*Proof.* This scenario is a special case for (5.32) by setting $i = \bar{m}$ and $j = \bar{m}$. □

We can solve the problem by randomly generating the sequence, randomly picking the scheduled start times from the slots, match the sequence and the start time for each procedure, and iteratively update the decision of whether each procedure is included in the schedule based on the corresponding expected cancelation risk. Given $M$ the maximum number of solutions you want to compare, the summary of the practical method is listed below.

**Algorithm 2** Algorithm for Solving the Static Scheduling Problem

*Step 0.* Set $z* = -\infty$, $c = 0$.

*Step 1.* Put all $m$ procedures into the set $P$.

*Step 2.* Randomly generate the sequence of the procedures in set $P$, randomly pick $|P|$ scheduled start times, and match the $|P|$ start times to the $|P|$ procedures based on the sequence.

*Step 3.* Check all first-stage constraints, if any one is violated, go to Step 8.

*Step 4.* Given the set $P$, the corresponding sequence and scheduled start time, compute all second-stage variables.

*Step 5.* Given the second stage variable, check the budget constraint. If the constraint is violated, go to Step 8.

*Step 6.* Set $p = \emptyset$. Check the expected cancelation risk for procedure in $P$, put procedures with the risk higher than threshold into set $p$. . Then,

If $p = \emptyset$ : go to Step 7.

Else : Set $P = P \backslash p$, and go back to Step 2.

*Step 7.* Compute the objective function value $z$, if $z > z*$ set $z* = z$ and record the current solution as the best solution.

*Step 8.* $c = c + 1$. If $c < M$, go back to Step 1; else Stop.

## 5.3 Numerical Test

We first tested the practical method on the historical data for orthopedics service. The data from 10/01/2006 to 01/31/2011 is used to determine the distribution of the procedure durations, turnover time, etc. Based on these analysis results, we apply the practical method to generate schedules for the real cases that occurred from 02/01/2011 to 04/01/2011, then compare the schedules with the optimal schedules and the schedules that the hospital came up before.

During the test period, there are 48 blocks (Date/OR combinations) for orthopedics service. The number of blocks for 1-procedure, 2-procedure and 3-procedure to schedule are 28, 16 and 4, respectively. Since the optimal solution for a 1-procedure scheduling problem is self-evident, we only test the 2-procedure and 3-procedure cases, i.e. 20 blocks. Because of the randomness in the practical method and in the simulation process, we duplicate 10 cases for each block, so there are 200 blocks to schedule.

We use an empirical distribution to describe the procedure duration for each procedure with a unique CPT code. The turnover time follows a Uniform distribution with the lower bound as 10 and the upper bound as 45.49. The LoS in PACU is described by an empirical distribution based on all procedures in the orthopedics service (independent of the CPT code).

We gave the procedures descending priorities based on their performing orders occurred in the test data. The relative costs are set to be the same, the cancelation rate is set to be 0, the cancelation threshold is set to be 0.8, each patient is assumed to have arrived 60 minutes before the scheduled start time. The schedule interval unit is 15 minutes. The number of realizations in the static scheduling problem is set to be 1000. The value of $\alpha$ is set to be 0.115, and the reason we choose this value is explained in the next sensitivity test. The budget is set to be 100.

We consider two types of test scenarios: the order of the procedures is pre-determined (as the original order occurred in the data), or the order of the procedures can be changed. For each test scenario, we first compare the practical solution with the optimal solution, then we compare the practical solution, optimal solution and historical solution (the schedule the hospital actually used) in a simulation run (10000 realizations), and last we compare the practical solution and the historical solution. Since the quality of the practical solution depends on how many random solutions we generated, we will do the above comparison for different numbers of solutions generated. The results are shown in the tables (5.5-5.10). The confidence intervals are all at 97.5% significance level, the computation times are in the units of second.

All cases were tested using Microsoft Visual Basic for Applications 2010 on a Windows 7 environment computer with Intel Core i7 2.93 GHz CPU and 8GB RAM.

Table 5.5: Comparison between the Practical Solution and the Optimal Solution for the Original Order Case

| No. Solutions | 500 | 1000 | 1500 | 2000 |
|---|---|---|---|---|
| Optimal % | 88.5% | 96% | 98% | 100% |
| Obj Gap CI | (-12.8%, 0%) | (-2.18%,0%) | (0%,0%) | (0%, 0%) |
| 2-p Optimal% | 100% | 100% | 100% | 100% |
| 3-p Optimal% | 42.5% | 80% | 90% | 100% |
| Computation Time CI | (5.2, 5.6) | (10.1, 11.0) | (15.0, 16.2) | (20.0, 21.6) |

Table 5.6: Simulation Results Comparison between Threes Solutions for the Original Order Case

| No. Solutions | 500 | 1000 | 1500 | 2000 |
|---|---|---|---|---|
| (P $\geq$ O)% | 89.5% | 96% | 99% | 100% |
| P-O Gap CI | (-22.0%, 0%) | (-3.6%,0%) | (0%,0%) | (0%, 0%) |
| (P $\geq$H)% | 88% | 88% | 87% | 88% |
| P-H Gap CI | (-141%, 341%) | (-111%, 325%) | (-104%,344%) | (-101&, 306%) |

Table 5.7: Comparison between the Practical Solution and the Historical Solution for the Original Order Case

| No. Solutions | 500 | 1000 | 1500 | 2000 |
|---|---|---|---|---|
| (P $\geq$ H) % | 89% | 86.5% | 88.5% | 88% |
| P-H Gap CI | (-34%, 1028%) | (-34%, 1028%) | (-34%, 1028%) | (-34%, 1028%) |
| 2-p (P $\geq$ H) % | 92.5% | 89.4% | 91.9% | 91.3% |
| 3-p (P $\geq$ H) % | 75% | 75% | 75% | 75% |

Table 5.8: Comparison between the Practical Solution and the Optimal Solution for the Re-Order Case

| No. Solutions | 1000 | 3000 | 5000 | 9000 | 13000 |
|---|---|---|---|---|---|
| Optimal% | 87% | 89% | 96% | 98% | 100% |
| Obj Gap CI | (-2.0%, 0%) | (-0.9%, 0%) | (-0.4%, 0%) | (0%, 0%) | (0%, 0%) |
| 2-p Optimal% | 100% | 100% | 100% | 100% | 100% |
| 3-p Optimal% | 35% | 45% | 80% | 88% | 100% |
| Comput. Time CI | (9.1, 9.6) | (27.3, 28.8) | (45.4, 47.9) | (81.7,86.2) | (118, 125) |

We can see that the solution quality and the computation time is highly dependent on the number of random solutions generated. Based on the test results, we have reasonable estimates of such dependency. These results provide a guideline to users about how they could adjust the

Table 5.9: Simulation Results Comparison between Threes Solutions for the Re-Order Case

| No. Solutions | 1000 | 3000 | 5000 | 9000 | 13000 |
|---|---|---|---|---|---|
| (P ≥ O)% | 87% | 91% | 96% | 97% | 99% |
| P-O Gap CI | (-2.3%, 0%) | (-1.0%, 0%) | (-0.3%, 0%) | (-0.0%, 0%) | (0%, 0%) |
| (P ≥ H) % | 99.5% | 100% | 100% | 100% | 100% |
| P-H Gap CI | (0%, 218%) | (0%, 230%) | (0%, 225%) | (0%, 236%) | (0%, 225%) |

Table 5.10: Comparison between the Practical Solution and the Historical Solution for the Re-Order Case

| No. Solutions | 1000 | 3000 | 5000 | 9000 | 13000 |
|---|---|---|---|---|---|
| (P ≥ H) % | 95% | 95% | 95% | 95% | 95% |
| P-H Gap CI | (-18%,298%) | (-18%,298%) | (-18%,298%) | (-18%,298%) | (-18%,298%) |
| 2-p (P ≥ H) % | 93.75% | 93.75% | 93.75% | 93.75% | 93.75% |
| 3-p (P ≥ H) % | 100% % | 100% | 100% | 100% | 100% |

settings to meet their requirement.

Next we study how sensitive the schedules are to the parameter $\alpha$. We still use the same schedule blocks, orthopedics schedule blocks from 02/01/2011 to 04/01/2011, which contains 27 10-hour blocks, 20 8-hour blocks and 1 7-hour block. The historical data from 10/01/2006 to 04/11/2011 is used to determine the distribution of the procedure durations, turnover time, LoS in PACU. The data is also used to compute the occurrence frequency of each procedure. Instead of using the real cases occurred in each schedule block, we randomly generated 3 ordered cases based on the occurrence frequency of each procedure. The reason we generate 3 cases is because it is the maximal number of procedures we scheduled for orthopedics service in the historical data. The priority of each procedure is 10, 9 and 8, respectively. Given the cases for each block, we use the practical method (Section 5.2) to generate a schedule assuming that the order of the procedures is pre-determined. All parameter settings are the same as previous tests except the number of solutions generated in each block scheduling process is fixed to be 2000, but the value of $\alpha$ varies.

We repeat the tests 10 times, and record the number of procedures scheduled for each block in each replication. Figure 5.3 shows how the expected number of procedures scheduled for each block changes over the value of $\alpha$. The larger $\alpha$ is, the smaller the expected number is, i.e. less procedures scheduled with high schedule quality is preferred. When $\alpha$ is 0.115, the expected number of procedures scheduled for each block is about 1.5, which is the same as what we observed in the historical data, so we use this value of $\alpha$ in the other tests to make sure that the tightness of the schedules are similar to what the hospital prefers.

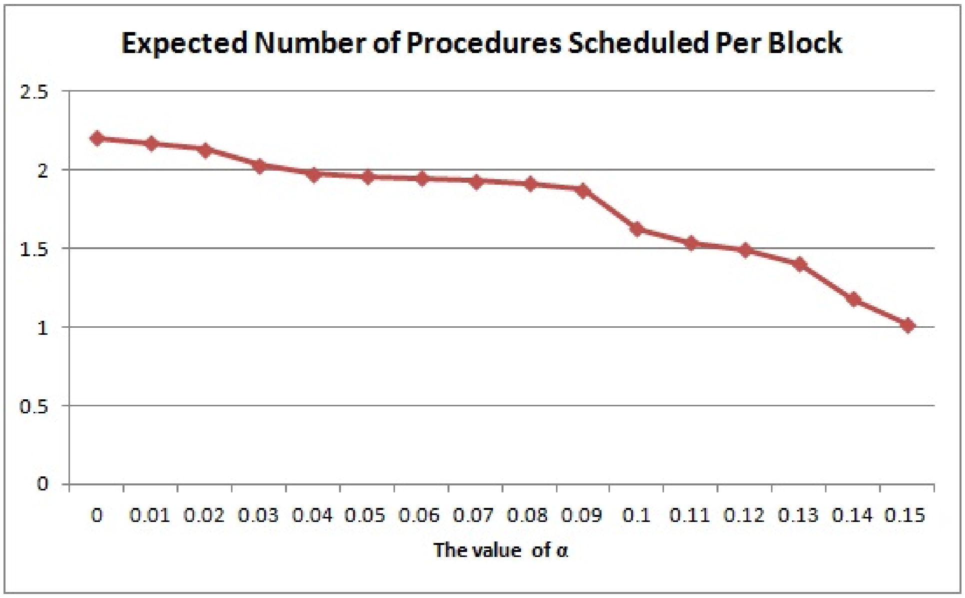



Figure 5.3: Expected Number of Procedures Scheduled Per Schedule Block over Different Values of $\alpha$

## 5.4 Dynamic Scheduling Problem

The schedule we provide in the last section is a static schedule. In this section, we describe how to adjust the schedule each day based on the real time situation. The different scenarios in the dynamic schedule environment were already considered when we modeled the static model, so we mainly focus on how to set up the static schedule so that the dynamic scheduling problem can be solved. The procedure finish time mentioned in this section refers to the time the patient leaves the OR.

### 5.4.1 Procedure is Canceled or Will Finish Early

There are two scenarios for this case. One is that we know for sure that some procedure scheduled to be performed later today is canceled, so the time scheduled for this canceled procedure is open. Another scenario is that the current procedure is about to finish and the finish time will be so much earlier than the expected finish time that the patient for the next procedure will not arrive when the turnover is done, so it leaves some open time as well. In both scenarios we want to know if we should add in back-up procedures to fill up the time; and if we want to add more procedures, if we should adjust the existing schedules.

The simplest way to handle the problem would be to run the static model again by adjusting some parameters. This time, the total regular open time of the OR should be tuned to match the time left. The canceled procedure should have a cancelation rate of 1. The priority of the other procedures scheduled should be set high enough to make sure they will be included in the new schedule. If a surgeon's calendar is affected by the original schedule, the release time should be set no earlier than the original scheduled start time. If a patient's arrival time is also affected by the original schedule (e.g. 1 hour before the original scheduled start time), the release time should also be set no earlier than the patient arrival time.

### 5.4.2 Procedure Will Be Finished Late

There are two scenarios in this case one is that we know for sure that the procedure currently being performed will be finished late due to some complication; and another is that the procedure is done but the patient cannot leave the OR because there is no room in PACU. In this case, we need to determine if we should cancel a procedure scheduled later, and schedule a small back-up procedure as a replacement.

We can utilize the static model to solve the problem by adjusting the release time of the following procedure. The release time should be no earlier than the expected finish time of the current procedure. The priority of each procedure should be adjusted correspondingly depending on if we want to introduce new procedures to replace existing ones. Finally, the total regular open time of the OR should be tuned to match the time left.

### 5.4.3 Procedure Will Be Delayed because of Resource Availability

The resource could be human resources, special equipment, etc. One scenario is that only one procedure will be delayed for certain due to patient's status, or a surgeon's sudden change of schedule, or waiting for important equipment. We need to determine if we should cancel this delayed procedure, and/or adjust the schedules for the following procedures.

The release time of this procedure should be adjusted to the expected ready time. If the procedure is really important and we do not want to cancel it, we can adjust the priority of the procedure to make sure that it will appear in the new schedule. The total regular open time of the OR should be tuned to match the time left.

The three cases outlined above are common cases that are used to demonstrate how to utilize the static model to solve the dynamic schedule problem. In reality, there could be more complicated scenarios. The users could solve the corresponding problems by adjusting the parameters in the static model. In general, the users can achieve different goals by adjusting the priority of the schedules for procedure selection, and/or by adjusting the release time for each procedure based on the scenario. The total regular open time of the OR should also be tuned to match the time left.

# CHAPTER 6

# SYSTEM DESIGN

In this chapter, we describe the system structure and its basic functionalities for a multi-user decision support system, enabled by our analytical solution. The aim is to develop a decision support information system to assist the scheduler in making daily schedules for the ORs. The system is built using Microsoft's Visual Basic for Application (VBA) language, with Microsoft Access used for managing data and displaying information. The process is explained in detail with the corresponding functions and users' role in Section 6.1. In Section 6.2, we talk about what information users with different priorities expect to get from this system.

## 6.1 Functionality

What does the scheduler need to make efficient schedules? First, all useful information about the procedures to be scheduled is needed. This includes what the procedures are, who will perform the procedures, what their schedules are, whether any special equipment is needed, etc. Thus, the scheduler can understand what is going on and what decisions need to be made. Then, the scheduler will need a platform to conveniently schedule these procedures. Estimation of procedure duration and turnover time would be helpful in this process. After a schedule is made, its efficiency should be evaluated and the evaluation information used as a guide to help the scheduler modify the schedule. Finally, when the schedule is completed, the relevant personnel should be able to view it with an option to provide feedback.

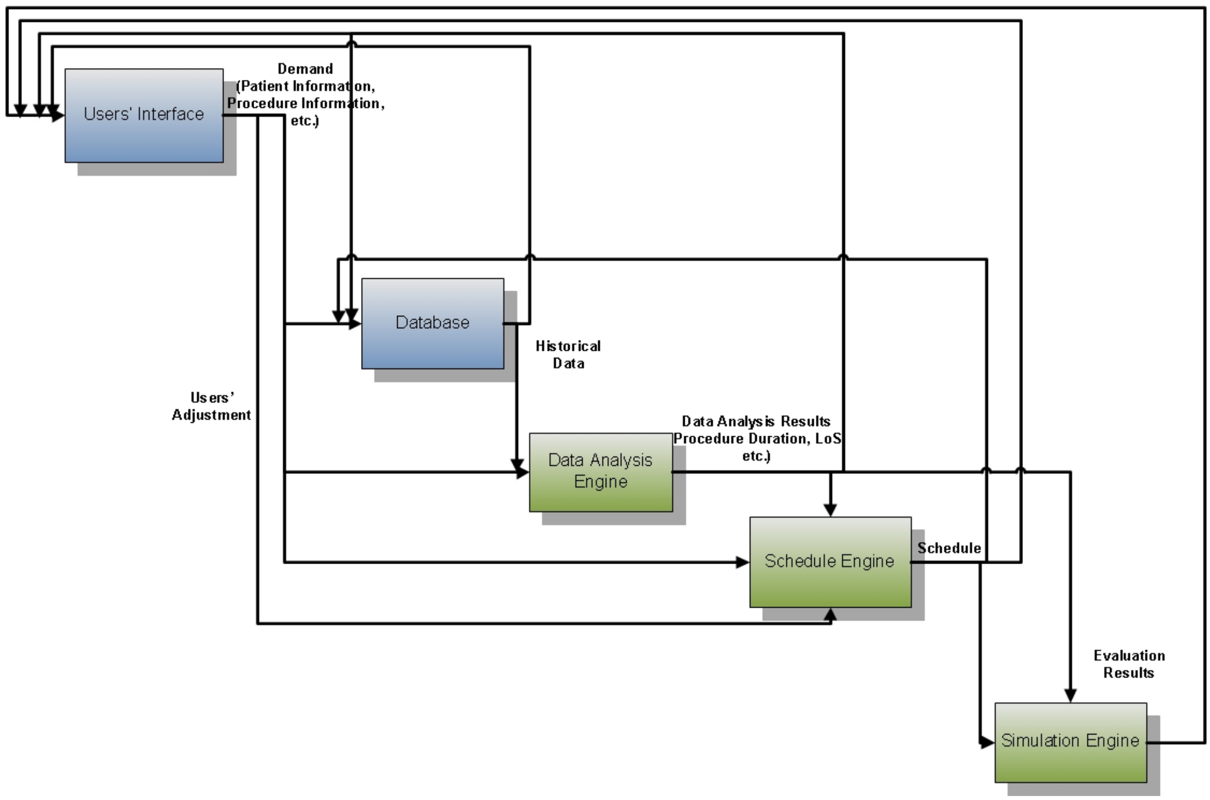



Figure 6.1: System Work Flow

The system is required to provide users with the following main functions: view and edit patient information, build schedules, evaluate schedules, view schedules, update case status, check case status, make calendars, check historical performance, try "what if" scenario test, and set up basic information about the hospital. These functions are addressed in the following subsections with the corresponding work flow in Figure 6.1. The data analysis engine is triggered in the form for setting up basic information about the hospital (Figure 6.4) for preparing and analyzing historical data, and all the results are stored in the database and surfaced to the users' interface based on the requirement. The scheduling engine is triggered by buttons in several forms for generating schedule dynamically (Figure 6.11, Figure 6.22, and Figure 6.31). In the process of generating schedules, the data analysis engine is also called to generate the distributions of procedure duration, turnover time and LoS in PACU. The simulation engine can be triggered in several forms for evaluating schedules (Figure 6.24, Figure 6.26, and Figure 6.31), the data analysis engine is also needed in this simulation process for generating the distributions.

### 6.1.1 Hospital Basic Information Setting

There are two tables that should be imported into the system before the users start to use it, History (Figure 6.2), which records history for surgical procedures, and PatientMovement (Figure 6.3), which records patients' movement.

History

| Field Name | Data Type |
|---|---|
| CaseNum | Text |
| ID | Number |
| PatientName | Text |
| CPT | Text |
| diagnosis | Text |
| OR | Text |
| Scheduled start | Text |
| Scheduled end | Text |
| Patient in OR | Date/Time |
| Patient out of OR | Date/Time |
| Operation Begins | Date/Time |
| Operation Ends | Date/Time |
| Physicians | Text |
| Nurses | Text |
| Sequence | Text |
| Priority | Number |
| Estimated_Duration | Number |
| Service | Text |
| Earliest_Start_Time | Date/Time |

Figure 6.2: Contents for Input Table "History"

PatientMovement

| Field Name | Data Type |
|---|---|
| CaseNum | Text |
| ID | Number |
| PatientName | Text |
| TRANSACTION | Text |
| WARD LOCATION | Text |
| DATE/TIME | Date/Time |

Figure 6.3: Contents for Input Table "PatientMovement"

When the user first begins to use the system, the form shown in Figure 6.4 can be used to initialize the system. There are two modes for the users to use: one is for demo and testing purposes; and the other is for regular daily scheduling purposes. All functionalities work for both modes, the main difference is that in the demo mode the system generates all settings in one click using some default settings, while in the regular mode, the users need to select setting individually and enter some resource and personnel information.

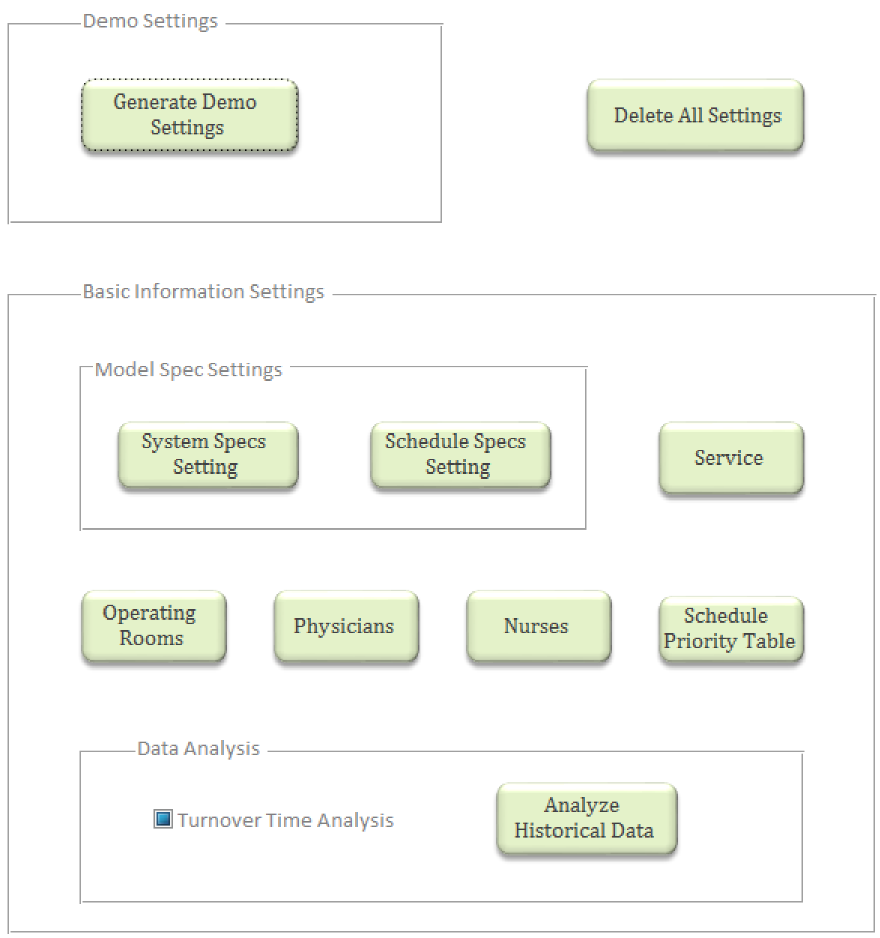
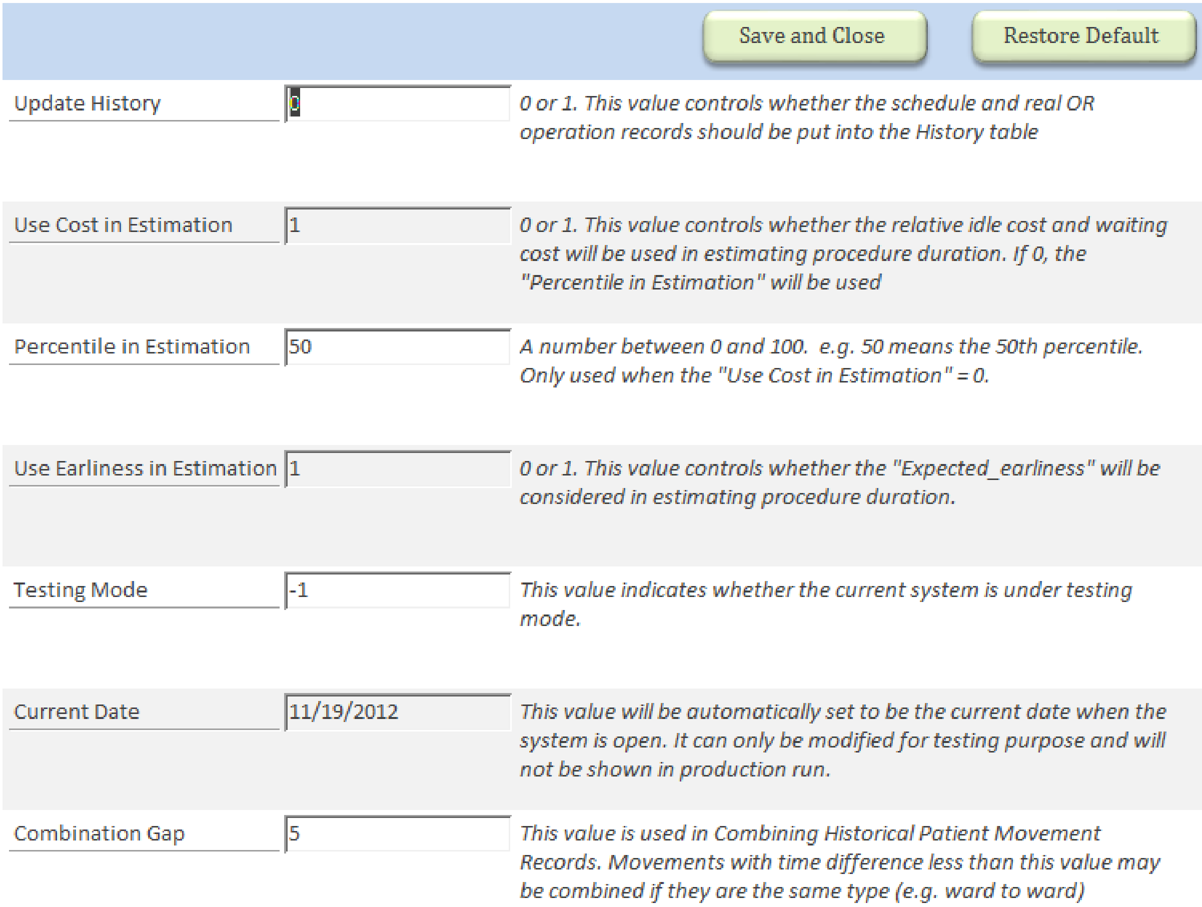



Figure 6.4: Hospital Basic Information Setting From

If "Generate Demo Settings" is clicked, the system analyzes historical data, generate some default specs and personnel and resource information for the user to familiarize themselves with the system. When the users want to switch to the regular mode, they can either click on "Delete All Settings", which sets the system back to the original status, or edit the Basic Information Settings below one by one to feed the system with correct information for regular operations.

The button "System Specs Setting" will bring up a form (Figure 6.5) which allows the user to control some options related to how the system handles the data, and "Schedule Specs Setting" will bring up a form (Figure 6.6) which allows the user to control options related to how the system generates schedules. "Service" allows the user to edit the service types the hospital provides and some default procedure duration, turnover time and LoS in PACU for each service on a pop-up form shown in Figure 6.7. "Operating Rooms" allows the user to edit the OR list, closing time for each OR, and OR open time for each weekday on a pop-up form illustrated in Figure 6.8. "Physicians" is for editing the physicians' list, their experience levels and the service they mainly concentrated on a pop-up form (Figure 6.9). "Nurses" is used for editing the names of the procedural nurses, which is on a pop-up form(Figure 6.10). The form brought up by "Schedule Priority Table" is for the user to edit this table, which controls the block-service arrangement. In the Data Analysis part, when "Turnover Time Analysis" is selected, clicking on "Analyze Historical Data" button triggers the system to analyze procedure durations, LoS in PACU and the turnover time, while when it is not selected, the turnover time analysis will not be analyzed again. Since the turnover time analysis takes quite a bit of time, it is recommended to be done on weekends. The entire data analysis part is recommended to be done each time after new input data is fed into the system.



Save and Close | Restore Default

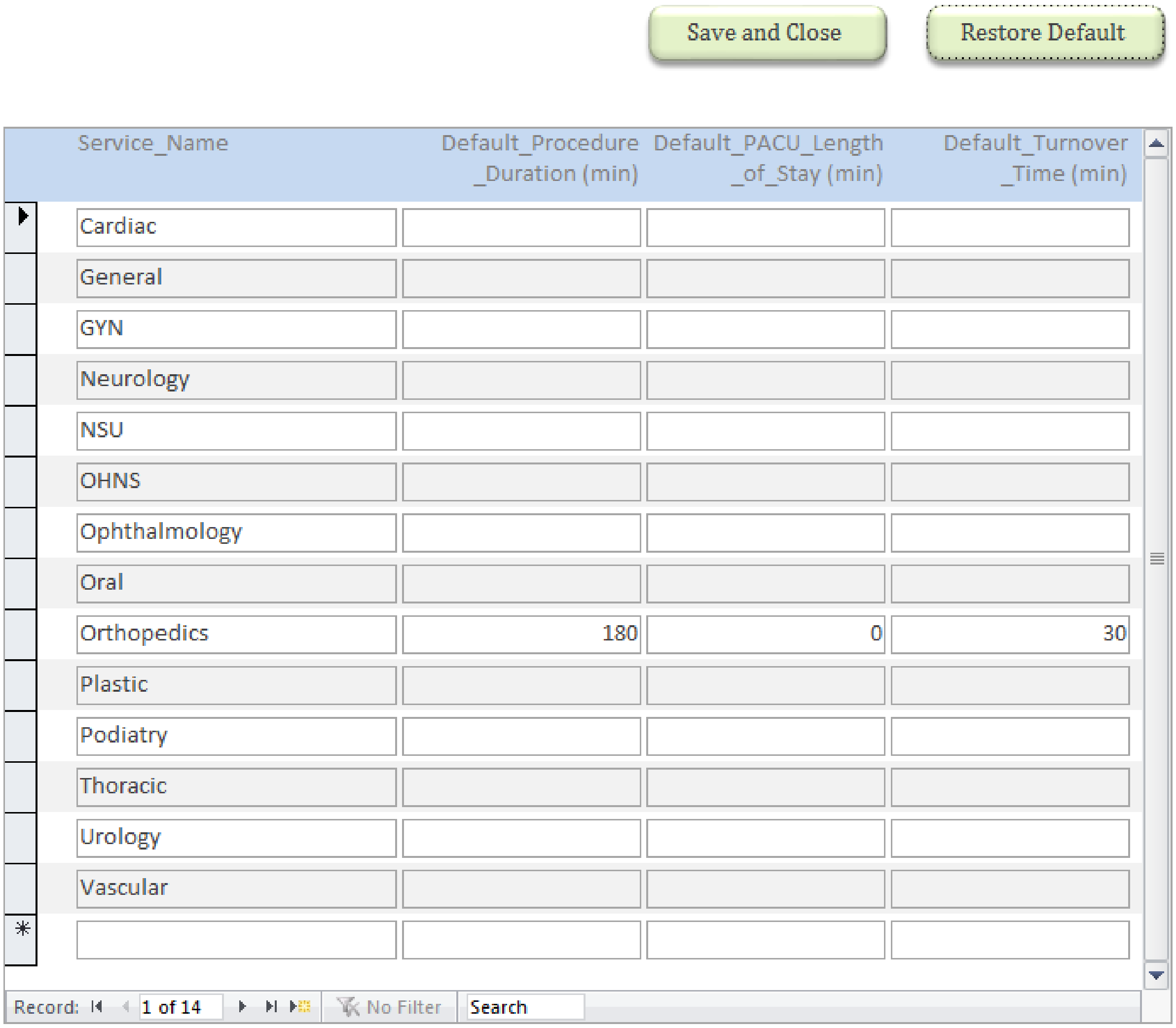
| | | |
|---|---|---|
| Update History | 0 | *0 or 1. This value controls whether the schedule and real OR operation records should be put into the History table* |
| Use Cost in Estimation | 1 | *0 or 1. This value controls whether the relative idle cost and waiting cost will be used in estimating procedure duration. If 0, the "Percentile in Estimation" will be used* |
| Percentile in Estimation | 50 | *A number between 0 and 100. e.g. 50 means the 50th percentile. Only used when the "Use Cost in Estimation" = 0.* |
| Use Earliness in Estimation | 1 | *0 or 1. This value controls whether the "Expected_earliness" will be considered in estimating procedure duration.* |
| Testing Mode | -1 | *This value indicates whether the current system is under testing mode.* |
| Current Date | 11/19/2012 | *This value will be automatically set to be the current date when the system is open. It can only be modified for testing purpose and will not be shown in production run.* |
| Combination Gap | 5 | *This value is used in Combining Historical Patient Movement Records. Movements with time difference less than this value may be combined if they are the same type (e.g. ward to ward)* |

Figure 6.5: System Settings Form

Save | Restore Default

| | | |
|---|---|---|
| Max_Num_Solution | 1000 | *Maximum number of solutions the automatic scheduling algorithm will evaluate* |
| Max_Num_Scenarios_in_Scheduling | 1000 | *Maximum number of scenarios used for evaluating each solution in automatic scheduling process* |
| Max_Num_Scenarios_in_Simulation | 10000 | *Maximum number of scenarios used for evaluating each schedule in simulation process, should be less than 10000* |
| Alpha | 1 | *Coefficient in the objective function to balance between number of procedures scheduled and schedule quality* |
| Same_cost_coefficient_for_all | 0 | *=0 when cost coefficient for each procedure is proportional to number of physicians; =1 when cost coefficient for each procedure is the same* |
| Expected_overtime_cost_unit | 1 | *Expected overtime cost per physician per minute* |
| Expected_waiting_cost_unit | 0.5 | *Expected waiting cost per physician per minute* |
| Expected_idle_cost_unit | 1 | *Expected idle cost per physician per minute* |
| Expected_staff_overtime_cost | 1 | *Expected staff overtime cost per staff per minute* |
| Expected_daily_overtime_budget | 2000 | *Expected overtime budget per OR per day* |
| Expected_earliness | 60 | *Minutes patients are supposed to arrive before scheduled start time* |
| Big_number_in_model | 100000 | *A large number used in the model* |
| Schedule_min_interval | 15 | *e.g. If it is 30, then the scheduled start time the system automatically generate will be like 8:00am, 8:30am, 9:00am, etc* |
| Duration_Estimate_Confidence | 0.1 | *A number between 0 and 1 to indicate how confident you are about the estimated procedure duration; =1 means only the estimation will be used in scheduling and simulation.* |
| Schedule_Cancellation_Tolerance | 0.8 | *A number between 0 and 1 to indicate when the procedure will not be put into schedule. e.g. default 0.8 means if the expected cancellation rate of a procedure in the schedule evaluation is 0.8* |

Figure 6.6: Schedule Settings Form

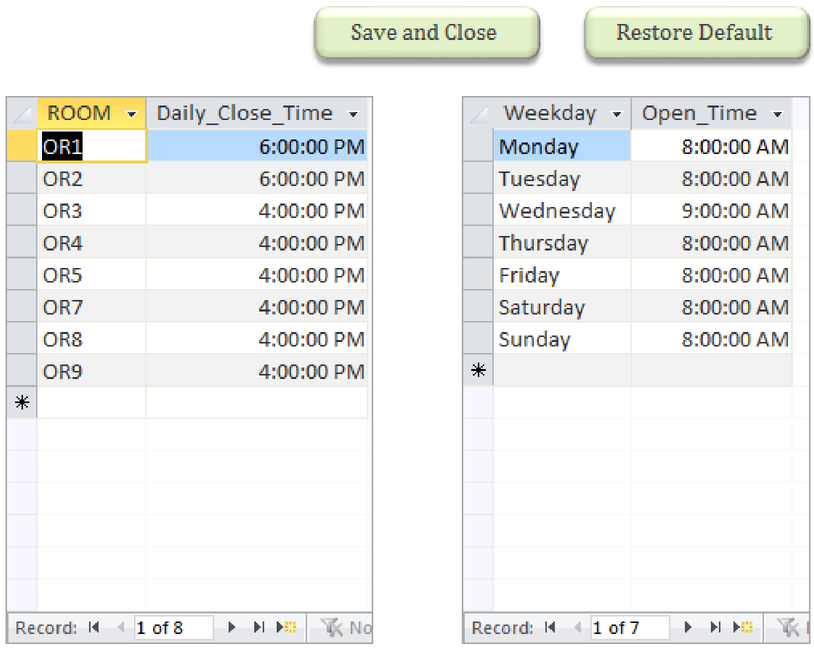


Save and Close | Restore Default

| Service_Name | Default_Procedure_Duration (min) | Default_PACU_Length_of_Stay (min) | Default_Turnover_Time (min) |
|---|---|---|---|
| Cardiac | | | |
| General | | | |
| GYN | | | |
| Neurology | | | |
| NSU | | | |
| OHNS | | | |
| Ophthalmology | | | |
| Oral | | | |
| Orthopedics | 180 | 0 | 30 |
| Plastic | | | |
| Podiatry | | | |
| Thoracic | | | |
| Urology | | | |
| Vascular | | | |
| | | | |

Record: 1 of 14 No Filter Search

Figure 6.7: Service Setup Form



Save and Close | Restore Default

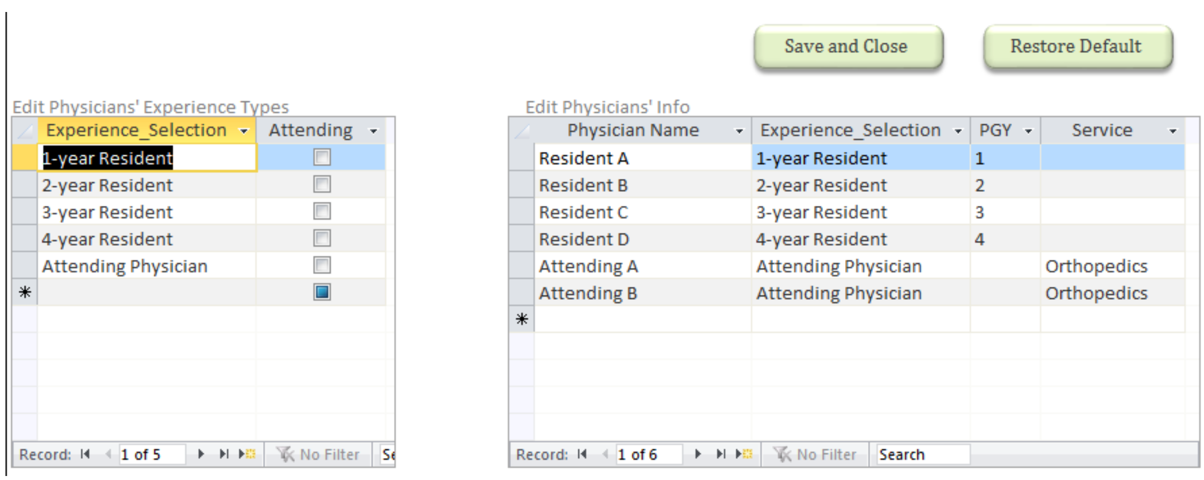
| ROOM | Daily_Close_Time |
|---|---|
| OR1 | 6:00:00 PM |
| OR2 | 6:00:00 PM |
| OR3 | 4:00:00 PM |
| OR4 | 4:00:00 PM |
| OR5 | 4:00:00 PM |
| OR7 | 4:00:00 PM |
| OR8 | 4:00:00 PM |
| OR9 | 4:00:00 PM |

Record: 1 of 8

| Weekday | Open_Time |
|---|---|
| Monday | 8:00:00 AM |
| Tuesday | 8:00:00 AM |
| Wednesday | 9:00:00 AM |
| Thursday | 8:00:00 AM |
| Friday | 8:00:00 AM |
| Saturday | 8:00:00 AM |
| Sunday | 8:00:00 AM |

Record: 1 of 7

Figure 6.8: OR Setup Form



Save and Close | Restore Default

Edit Physicians' Experience Types

| Experience_Selection | Attending |
|---|---|
| 1-year Resident | ☐ |
| 2-year Resident | ☐ |
| 3-year Resident | ☐ |
| 4-year Resident | ☐ |
| Attending Physician | ☐ |

Record: 1 of 5 No Filter

Edit Physicians' Info

| Physician Name | Experience_Selection | PGY | Service |
|---|---|---|---|
| Resident A | 1-year Resident | 1 | |
| Resident B | 2-year Resident | 2 | |
| Resident C | 3-year Resident | 3 | |
| Resident D | 4-year Resident | 4 | |
| Attending A | Attending Physician | | Orthopedics |
| Attending B | Attending Physician | | Orthopedics |

Record: 1 of 6 No Filter Search

Figure 6.9: Physicians Setup Form

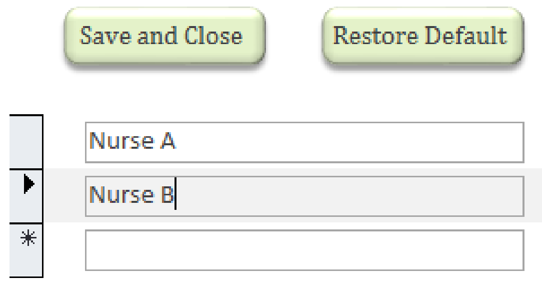



Figure 6.10: Nurses Setup Form

### 6.1.2 Test Schedule Settings

Once the schedule specs have been set, the users can use the form shown in Figure 6.11 to test whether the specs are set in such a way that the system generated schedule is desired. The users can set a date range in the history, then click "Test" to let the system generate schedules for all blocks within the date range. After this is done, the users can click "View Test Results" to bring up the form shown in Figure 6.12 to review the schedule results. The users can select a date from the previous date range and select an OR to locate a schedule block, after the button "View" is clicked, the system generated schedule, schedule used in the history and actual usage of the schedule block will be shown on the right of the form, and statistics about the system generated schedule and what actually happened in the history will be shown below.

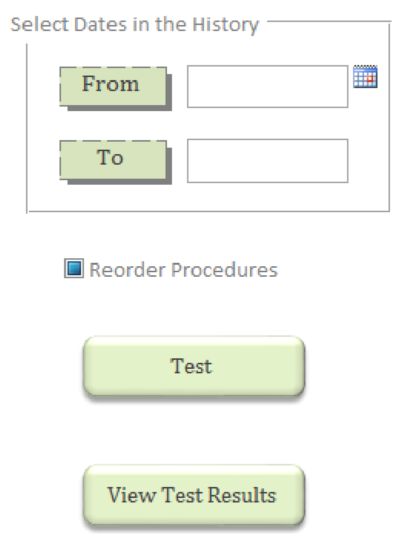



Figure 6.11: Test Schedule Settings Form

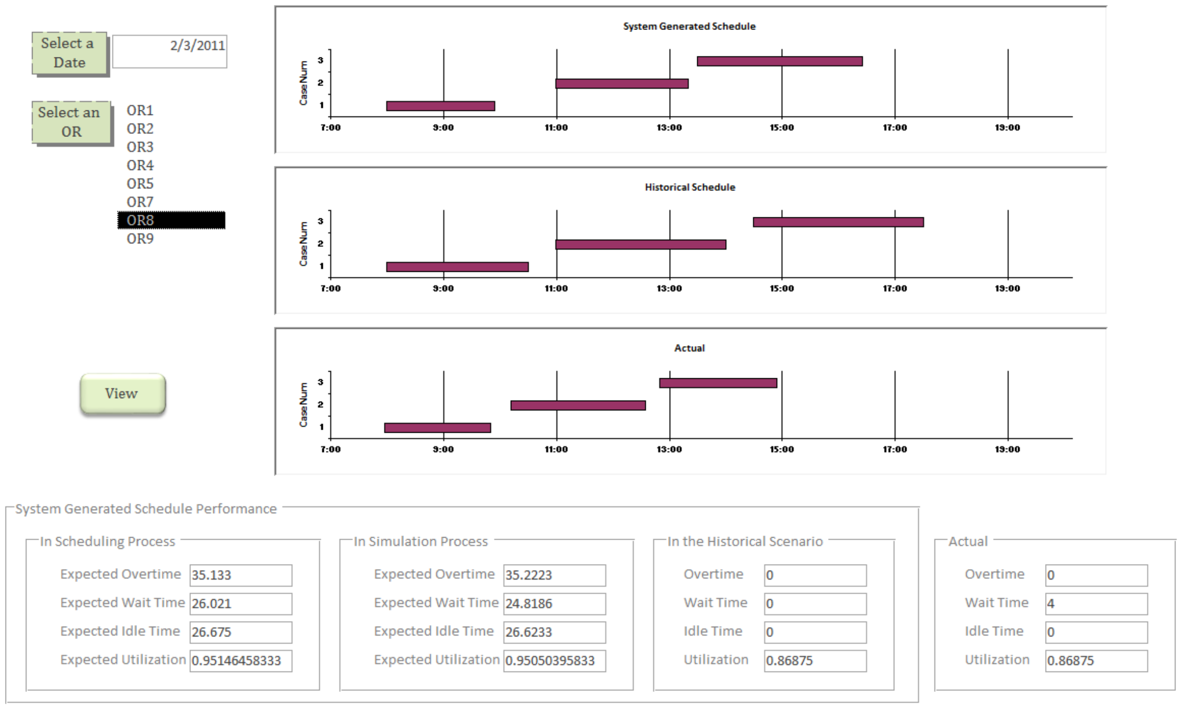



Figure 6.12: Test Schedule Settings - View Results Form

### 6.1.3 Create, View and Edit Patient Information

Figure 6.13 illustrates the patient information form for the orthopedics service. This form is used to record the detailed information of each surgical procedure. It includes patients' basic information, patients' visit record, procedure descriptions, special surgical needs, pre-surgery requirements, post-surgery arrangements, and the surgical team. It provides information for both the scheduler, for making schedules, and physicians, for practicing medicine. The information canl be entered through the user interface and stored in the database. Some information also serves as input for the data analysis engine and the schedule engine.

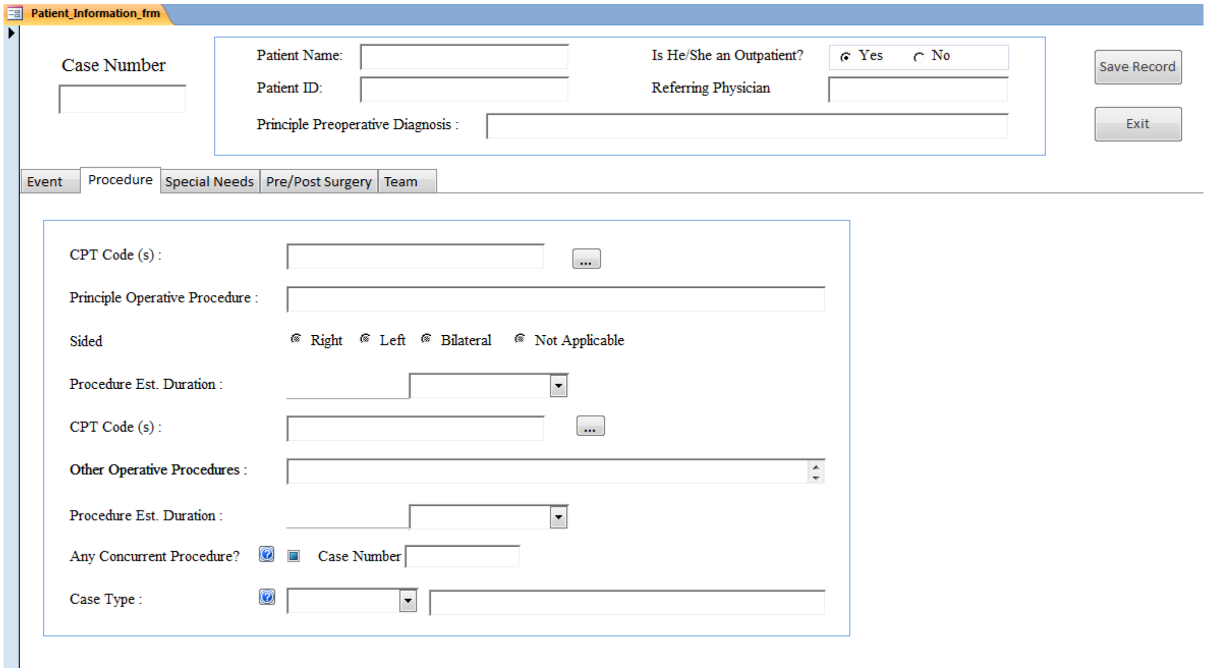



Figure 6.13: Patient Information Form

A commonly seen problem in filling out such a form in reality is that surgeons sometimes choose an inexact CPT code. It can be difficult to remember all the CPT codes and their descriptions, so the system will also provide help if they need to look up the CPT codes. Figure 6.14 shows the CPT finder form for orthopedics service. In addition, the estimation for procedure duration, length of stay in PACU and ICU will be displayed as a reference to the users.

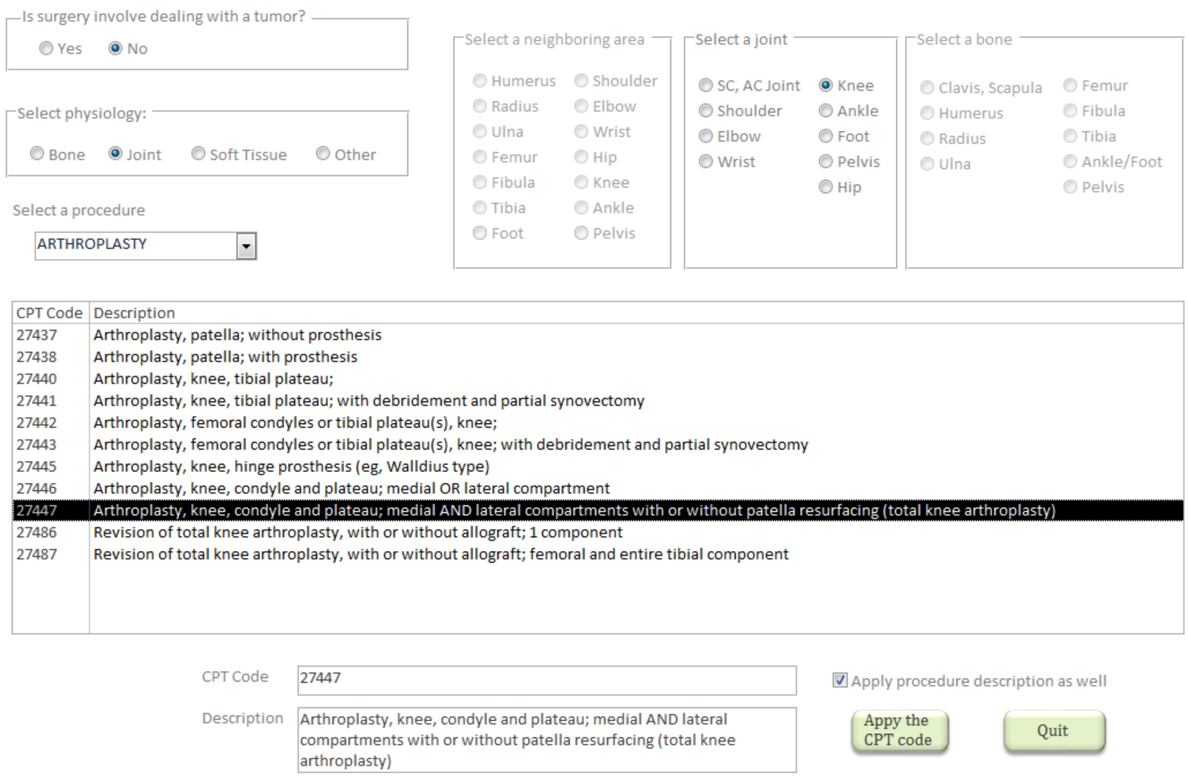



Figure 6.14: CPT Finder Form

### 6.1.4 View and Edit Priority Schedule Table

As introduced in Chapter 1, the service type for each schedule block should follow the guidance in “Priority Schedule Table”. In order to accomplish this, the information needs to be included in the system and should be treated as a reference during the entire scheduling process. Figure 6.15 illustrates how this table could be edited, and Figure 6.16 illustrates how the information is shown to the interested users.

| OR | Date | Service | Weekday |
|---|---|---|---|
| OR1 | 10/4/2012 | Cardiac | Thursday |
| OR1 | 10/5/2012 | Cardiac | Friday |
| OR2 | 10/1/2012 | General | Monday |
| OR2 | 10/2/2012 | GYN | Tuesday |
| OR2 | 10/3/2012 | Neurology | Wednesday |
| OR2 | 10/4/2012 | NSU | Thursday |
| OR2 | 10/5/2012 | Orthopedics | Friday |
| OR3 | 10/1/2012 | Orthopedics | Monday |
| OR3 | 10/2/2012 | Plastic | Tuesday |
| OR3 | 10/3/2012 | Oral | Wednesday |
| OR3 | 10/4/2012 | Orthopedics | Thursday |
| OR3 | 10/5/2012 | Urology | Friday |
| OR1 | 10/1/2012 | Cardiac | Monday |
| OR1 | 10/2/2012 | Cardiac | Tuesday |
| OR1 | 10/3/2012 | Cardiac | Wednesday |

Figure 6.15: Edit Priority Schedule Table Form

| _Year_ | _Month_ | _Week_ |
|---|---|---|
| 2012 | Oct | 10/1/2012 ~ 10/7/2012 |

| | Weekday | | | | |
|---|---|---|---|---|---|
| | Monday | Tuesday | Wednesday | Thursday | Friday |
| OR | Service | Service | Service | Service | Service |
| OR1 | Cardiac | Cardiac | Cardiac | Cardiac | Cardiac |
| OR2 | General | GYN | Neurology | NSU | Orthopedics |
| OR3 | Orthopedics | Plastic | Oral | Orthopedics | Urology |

Figure 6.16: View Priority Schedule Table Form

### 6.1.5 Build a Schedule

As mentioned in Chapter 1, the basic workflow for creating a schedule is as follows. First, the surgeon comes up with an initial list of procedures to be performed and maybe also provide orders for these procedures three days prior to the scheduled procedure. This is called an initial schedule. The scheduler then schedule the procedures based on this initial schedule, and the schedule will be published at 1pm one day before the scheduled day. On the day of the scheduled procedures, the scheduler may need to adjust the schedule to handle different situations that occur on that day, this is referred to as the dynamic schedule. So, there are actually three schedules considered, the initial schedule, the published schedule and the dynamic schedule.

### Build Initial Schedule

When the surgeon starts to build the initial schedule, they can start with either selecting procedures from the patient waiting list (as illustrated in Figure 6.17) or manually adding procedure information (as illustrated in Figure 6.18). After the procedures are selected, the form shown in Figure 6.19 can be used to edit the initial schedule and enter additional information for each procedure. Once the surgeon is satisfied with the initial schedule, the form shown in Figure 6.20 can be used to select the finished initial schedule and send it to the scheduler.

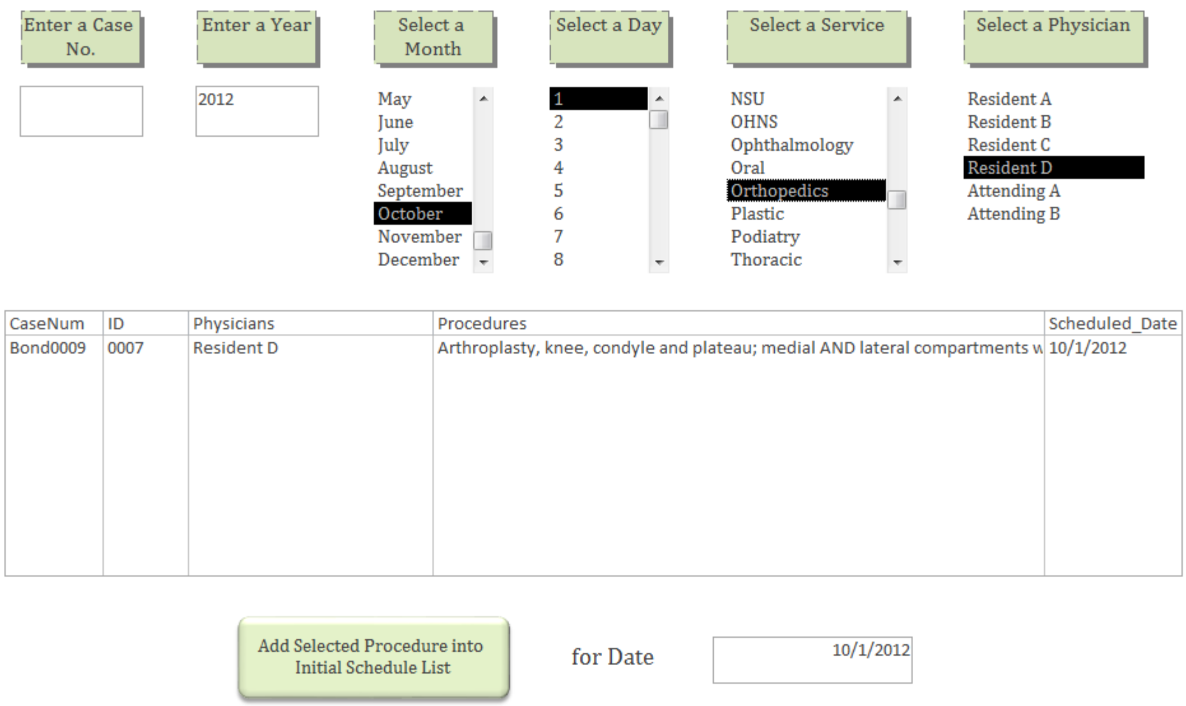



Figure 6.17: Build Initial Schedule - Add Procedures from Waiting List

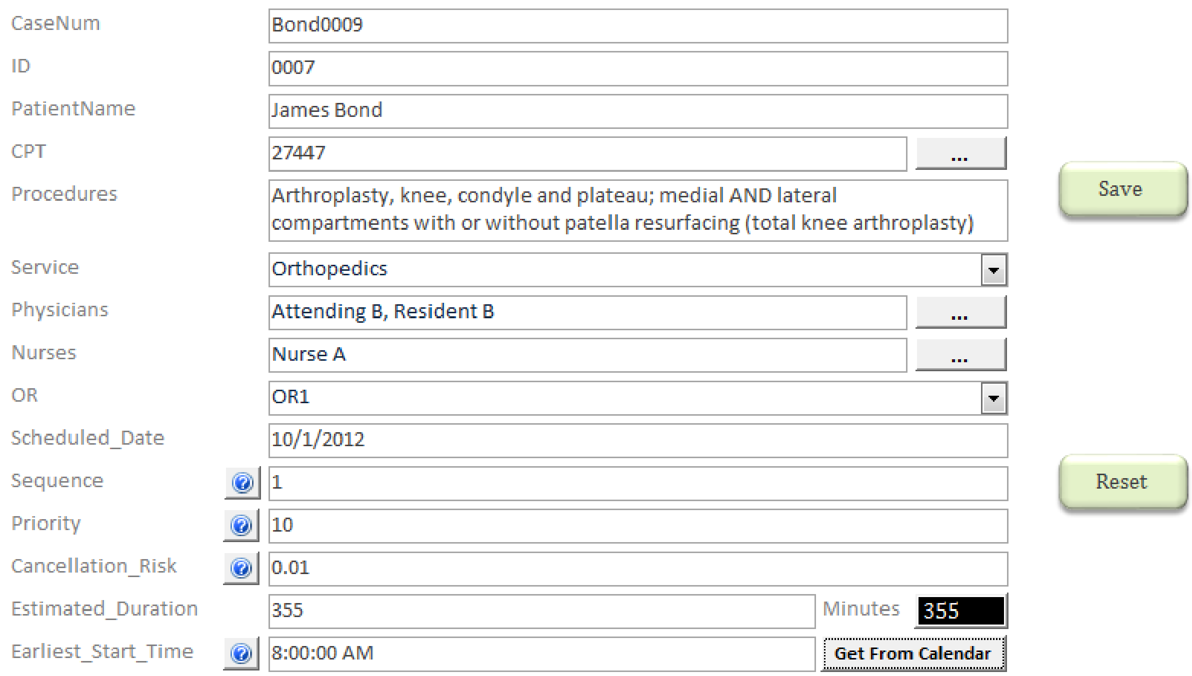



Figure 6.18: Build Initial Schedule - Add Procedures Manually

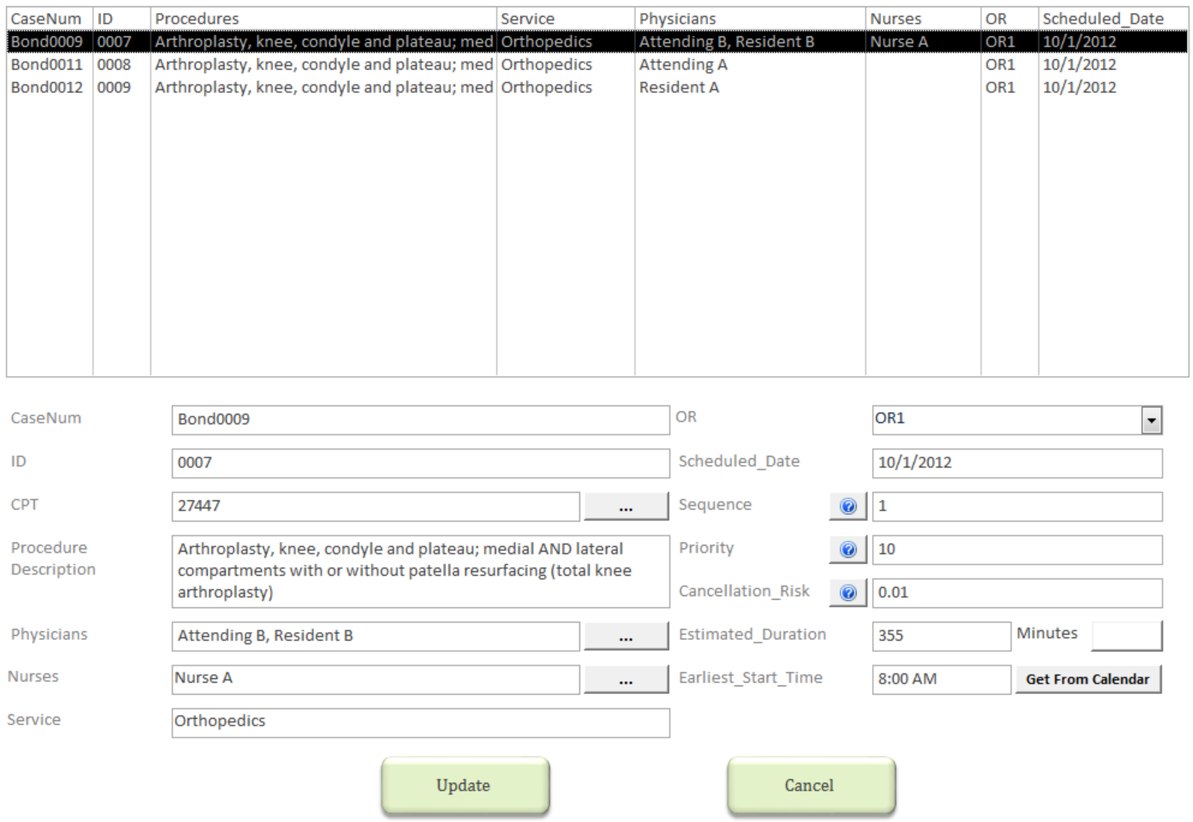



Figure 6.19: Build Initial Schedule - Edit Information

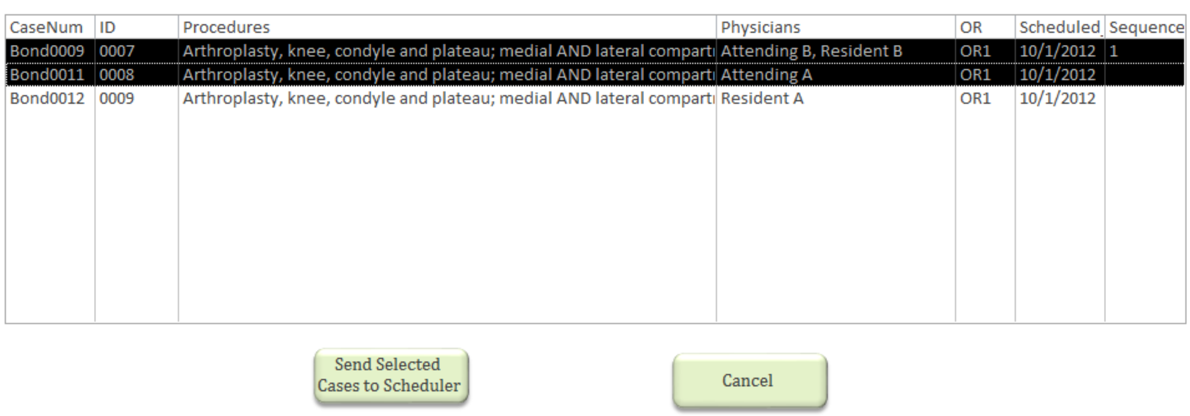



Figure 6.20: Build Initial Schedule - Send List to Scheduler

CPT code lookup (with procedure names filled out), procedure duration estimates, and the earliest start time of the procedure based on physicians' calendar are all provided in the corresponding forms to assist the users in filling out the form.

### Build Published Schedule

The initial schedule the surgeon sends to the scheduler is in the form shown in Figure 6.21. The scheduler can then click on the corresponding procedure, update the information if necessary, then click "Update and Add into Schedule List" button to add the information into the schedule list. This is an un-published schedule, which is a local schedule draft that will not be seen by other users. The scheduler can also manually enter information without selecting any information from the initial schedule list, and click the "Add into Schedule List" button to serve the same purpose.

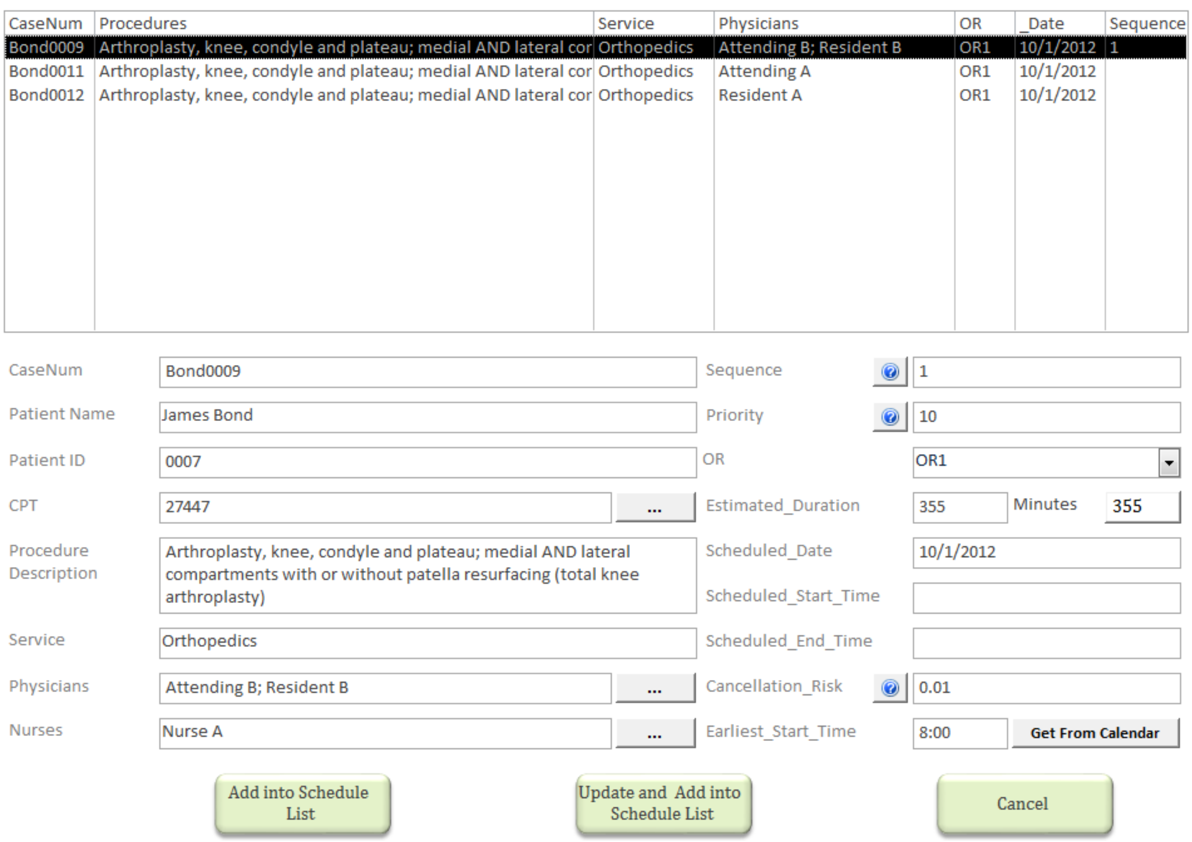



Figure 6.21: Build Schedule - Add Procedures

Figure 6.22 shows the interface for generating schedules automatically by the system. First, the user needs to select which group of procedures is going to be considered. It can be from the initial schedule surgeons provide, or from the un-published schedule, or from the published schedule, or any combination of these sources. When the group is selected, the procedures will be shown in the list below, and the user can select procedures they want the system to consider in generating the schedule. After the button "Schedule Selected procedures" is clicked, the system will call schedule engine to generate a best schedule based on the model and algorithm we described in Chapter 5. When it is finished, the form illustrated in Figure 6.23 will pop-up on the screen and the user can select one schedule block by choosing the date/OR combination and click "View" button to take a look at the generated schedule for the block and the expected performance of the schedule in the scheduling process shown on the right part of the figure. Then the user can decide if they want to "Save Schedule", which saves the schedule as an un-published schedule, or "Cancel" the operation.

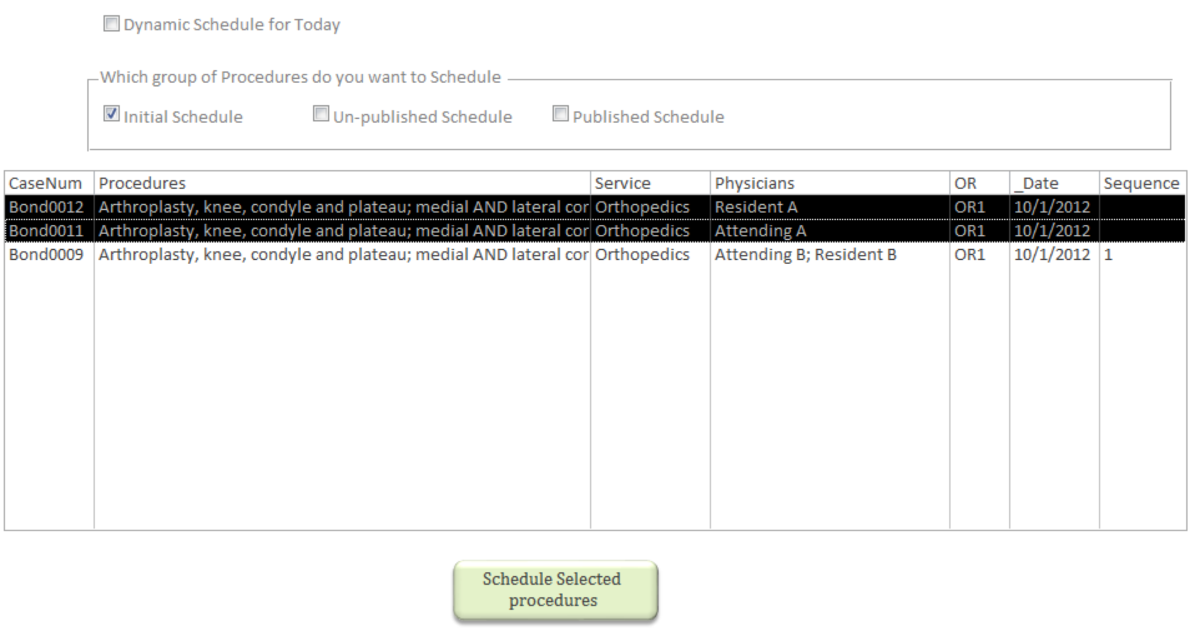



Figure 6.22: Build Schedule - Generate Schedule Automatically

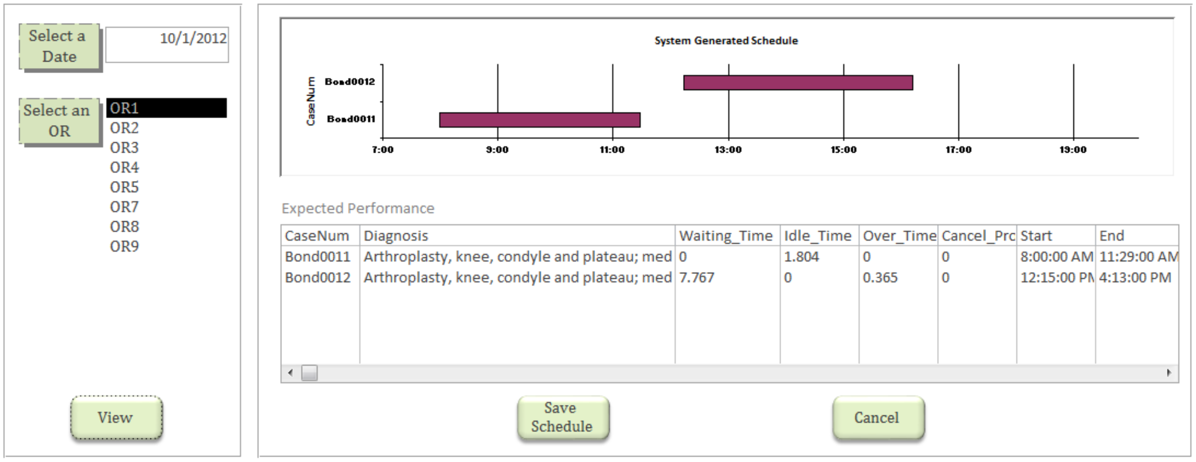



Figure 6.23: Build Schedule - View Generated Schedule and Expected Performance

A similar form shown in Figure 6.24 can be used to check other un-published or published schedules that are input by the user instead of generated by the system. The expected performance is obtained by calling the simulation engine to evaluate the schedule. Based on the visual display of the schedule and the expected performance, the user can adjust the schedule as they want.

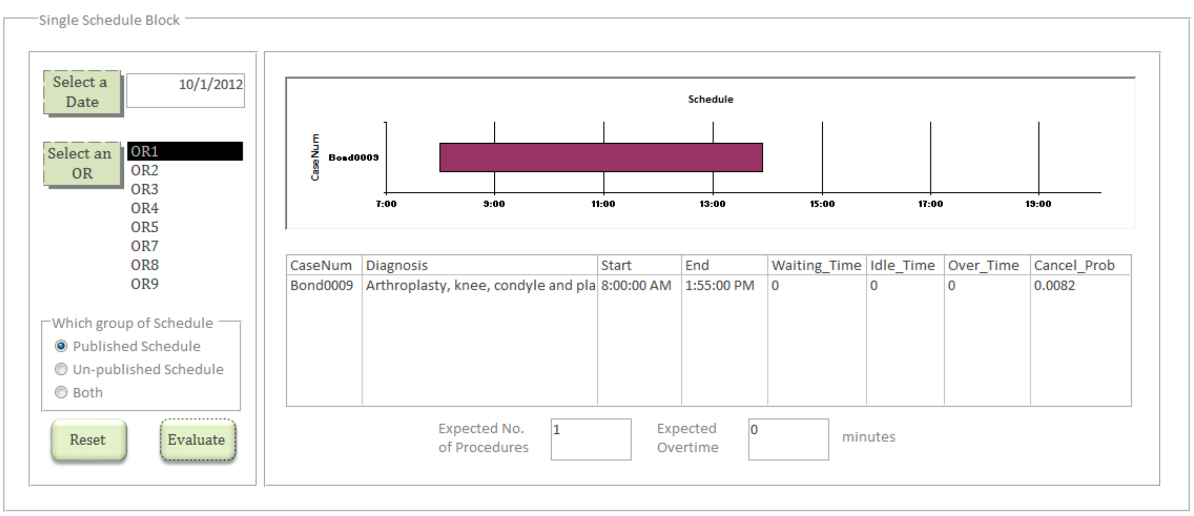



Figure 6.24: Build Schedule - View Schedule and Evaluation Results

The user can use the form shown in Figure 6.25 to locate all un-published and published schedules, then edit a schedule and save the information by clicking the “Save” button, or remove the schedule and take the procedure back to the waiting list by clicking ”Remove Schedule”, or completely delete the schedule without putting the records back into the waiting list by clicking “Delete Records”.

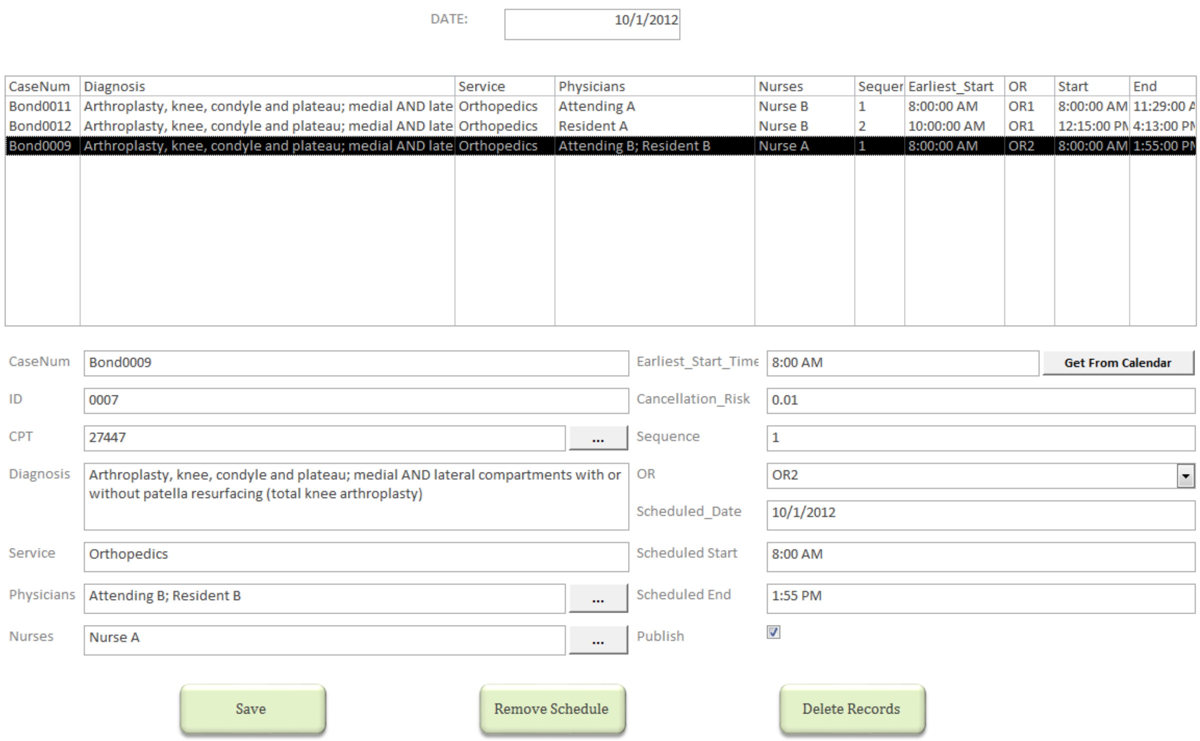



Figure 6.25: Build Schedule - Edit/Publish Schedule

### Build Dynamic Schedule

The dynamic schedule can be easily generated using the form illustrated in Figure 6.22 by checking the “Dynamic Schedule for Today” button and then perform the steps that have already been described for this form. The system will check the status of all procedures and only show procedures that have not been performed yet. The earliest start time of each procedure will also be adjusted to be no earlier than the current time when the system generates schedules and the total time remaining for the block will also be adjusted as well. To better utilize this functionality, for example, handle the scenarios mentioned in Section 5.4, the user should first edit the procedure information accordingly using the forms illustrated in Figure 6.21 and Figure 6.25 before coming to this form for generating a dynamic schedule.

### 6.1.6 Evaluate Schedule

In addition to the single block schedule evaluation form shown in Figure 6.24, a multi-block schedule evaluation form illustrated in Figure 6.26 is also provided to the user to check schedules for several blocks so that they can re-allocate procedures into a more desired block to balance the usage of several blocks.

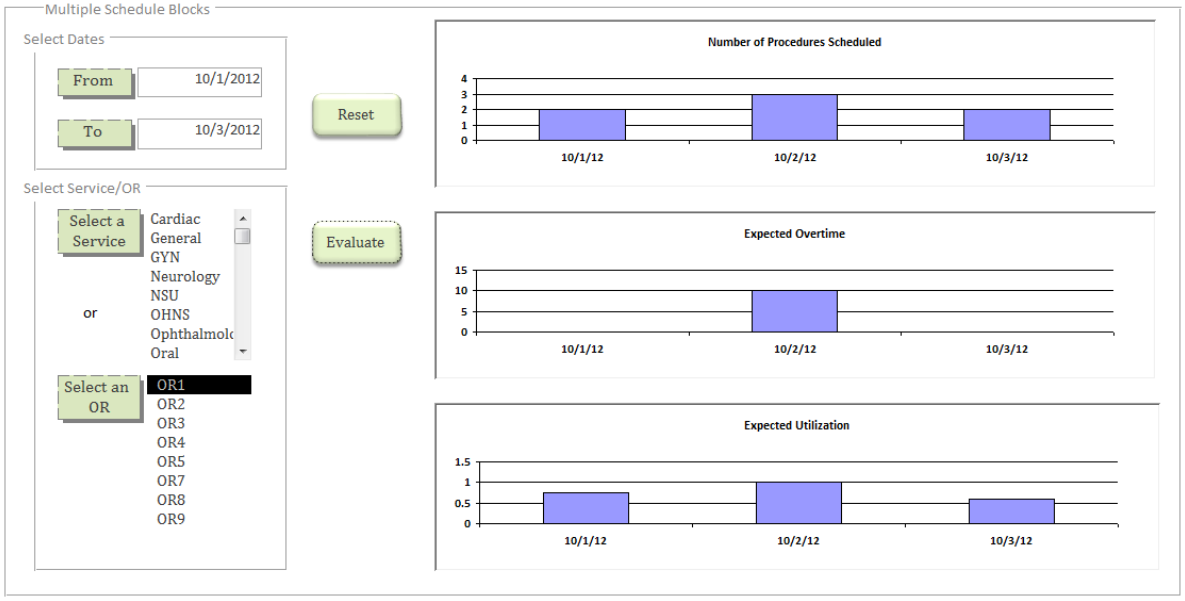



Figure 6.26: View Multiple Block Schedule Evaluation Results

### 6.1.7 View Schedule

Once the schedule is published, the users can view the contents as illustrated in Figure 6.27. They can also filter the results by providing case number, or providing patient name, selecting service, selecting physician name, selecting nurses, selecting OR, or choosing a schedule date.

| CaseNum | PatientName | CPT | Diagnosis | Service | Physicians | Nurses | OR | _Date | Start | End | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | James | | | Orthopedics | | | | 10/1/2012 | | | Search |
| Bond0011 | James B Bond | 27447 | Arthroplasty, knee, condyle and plateau; medial AND lateral | Orthopedics | Attending A | Nurse B | OR1 | 10/1/2012 | 8:00:00 AM | 11:29:00 AM | |
| Bond0012 | James C Bond | 27447 | Arthroplasty, knee, condyle and plateau; medial AND lateral | Orthopedics | Resident A | Nurse B | OR1 | 10/1/2012 | 12:15:00 PM | 4:13:00 PM | |
| Bond0009 | James Bond | 27447 | Arthroplasty, knee, condyle and plateau; medial AND lateral | Orthopedics | Attending B; Resident B | Nurse A | OR2 | 10/1/2012 | 8:00:00 AM | 1:55:00 PM | |

Figure 6.27: View Published Schedule

### 6.1.8 Make/View Calendar

The form shown in Figure 6.28 provides physicians with a platform where they can indicate their planned absence or other obligations which may impact the OR schedules. It will be used as a reference when the physicians are trying to build the initial schedule and when the scheduler is trying to finalize the schedule. The physician can add or update their own calendar and can view others' calendars.

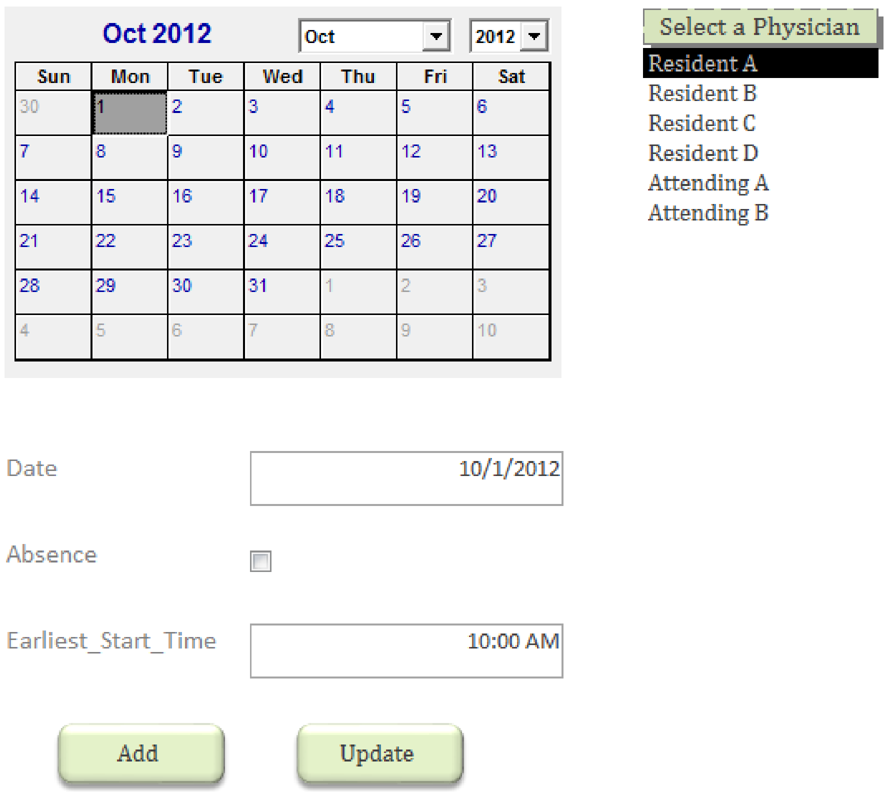



Figure 6.28: Make/View Calendar From

### 6.1.9 Case Tracking

At the beginning of a shift, the nurse or other users can bring up the form illustrated in Figure 6.29, and all scheduled procedures for this OR for the current date will be displayed. When the patient is brought into the OR, the user can select the corresponding procedure and click the "Patient In OR" button, so that the time the patient entered the OR will be recorded. After this button is clicked, this procedure record will show up in the list below, and the user can update the progress of the case by clicking the corresponding buttons ("Operation Begins", “Operation Ends”, “Patient Out OR”). Updating the progress of the cases is important because this information is needed in order to track the block usage and update the historical data.


Figure 6.29: Update Case Status Form

Figure 6.30 illustrates the form that can be used to check the status of each OR. The user can first select an OR and then click the “View Status” button to check the progress of the cases performed in the OR and how the OR is used. As illustrated in the figure, the first graph shows how the OR is scheduled, and the second graph shows how the OR is actually being used. By checking this information, we can have a valid idea of whether the OR is well utilized and whether we should adjust the schedule (dynamic schedule) accordingly.

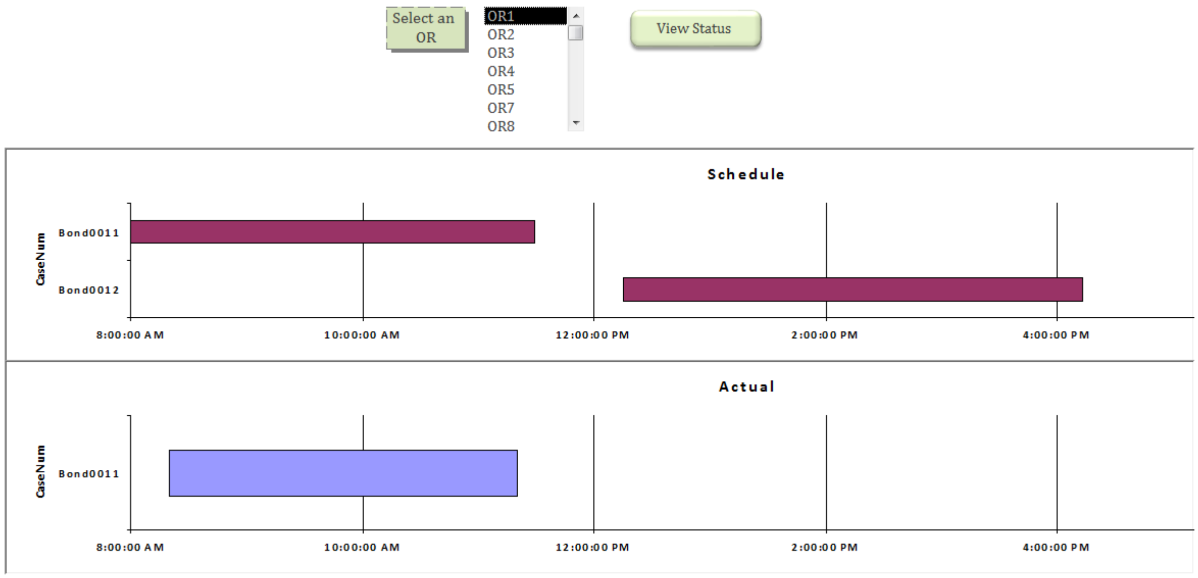



Figure 6.30: Check OR Status Form

### 6.1.10 Historical Performance Evaluation

The administrators of the hospital may need to look at the performance of each service each month to help them make some facility and/or personnel decisions. The form shown in Figure 6.31 provides such functionality to the users. They can select a date range, select a service or an OR, then click the "Evaluate" button to check how many procedures are performed, what is the overtime, and what is the utilization of each block as shown in the right side of the figure.

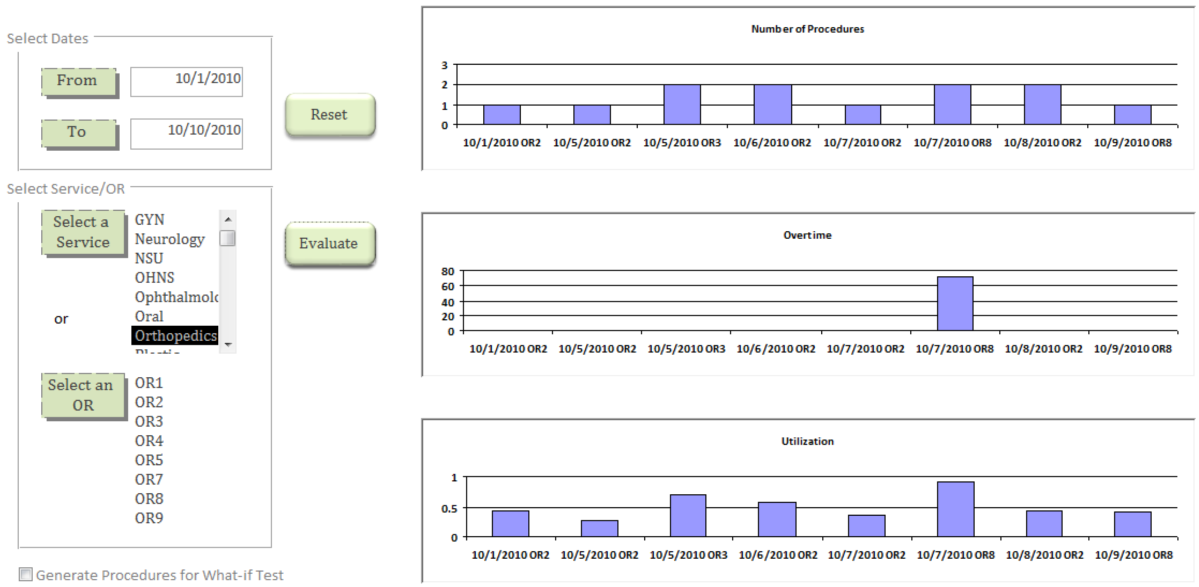



Figure 6.31: Historical Performance Evaluation and "What if" Test Form

### 6.1.11 "What if" Scenario Test

The form shown in Figure 6.31 can also be used in the "what if" simulation test we mentioned in Chapter 4. By checking this checkbox, the system will not compute the historical performance based on the historical data. It will randomly generated procedures based on the procedure occurrence probability in the history, generate schedules for the requested blocks, and then call the simulation engine to simulate the expected behavior of these blocks. By comparing the simulated behavior under current hospital setup with that of the desired setup, the users can test "what if" decisions about the hospital setting up options. For example, if the user want to study how many extra procedures will the hospital be able to perform if the standard operating hours of all ORs are extended to 10 hours, they can check the "Generate Procedures for What-if Test" checkbox after setting up the date ranges and service/OR, and then click "Evaluate" to see expected simulated behaviors for the corresponding blocks, and then they can change the OR daily closed time in the form shown in Figure 6.8, and click the "Evaluate" button again to check the expected simulated behaviors after this OR operating time change.

## 6.2 Multi-User Design

Since the users, administrators, physicians, nurses, etc., have different functions and different concerns in the hospital, some functions are only available to particular users. For example, if all the surgeons can edit schedules, it is quite likely that they will change the schedules according to their own wishes, thus you can imagine what chaos this could cause. We designed four different main menus according to the user's functions in the hospital. They are specifically designed for administrators, schedulers, nurses, and physicians. Based on the information the user enters when they log in to the system using the form illustrated in Figure 6.32, they will be led to a menu designed for their functions. The sequences of the functions in the menus are designed to cater to their different concerns as well.


Figure 6.32: Login Form

### 6.2.1 Administrator's Menu

Figure 6.33 shows the main menu for administrators. Since the administrators are more interested in what is going on in the hospital, in general, such as whether the resources are properly utilized, and they have better knowledge of the general information of the hospital, we provide case/OR tracking, view schedule, view schedule evaluation summary, view hospital historical performance summary, try "what if" scenarios, view calendar and setting up hospital basic information functionalities for these users. The administrators can click the blue navigation tabs below the main menu caption to select interfaces for desired functions. The initial (default) function displayed, "Case Status Tracking", helps the administrators know what is going on in the hospital quickly.


Figure 6.33: Main Menu for Administrators

### 6.2.2 Scheduler's Menu

On the scheduler's menu (Figure 6.34), users can use all functionalities related to building the published schedule as mentioned in Section 6.1.5 and dynamic schedule as mentioned in Section 6.1.5, editing "Priority Schedule Table", view schedules, view patient information, tracking case/OR, view physicians' calendar, setting up schedule specs, and testing schedule settings as mentioned in Section 6.1.2. Since making schedules is their primary task, "Build Schedule" is the initial (default) function displayed.

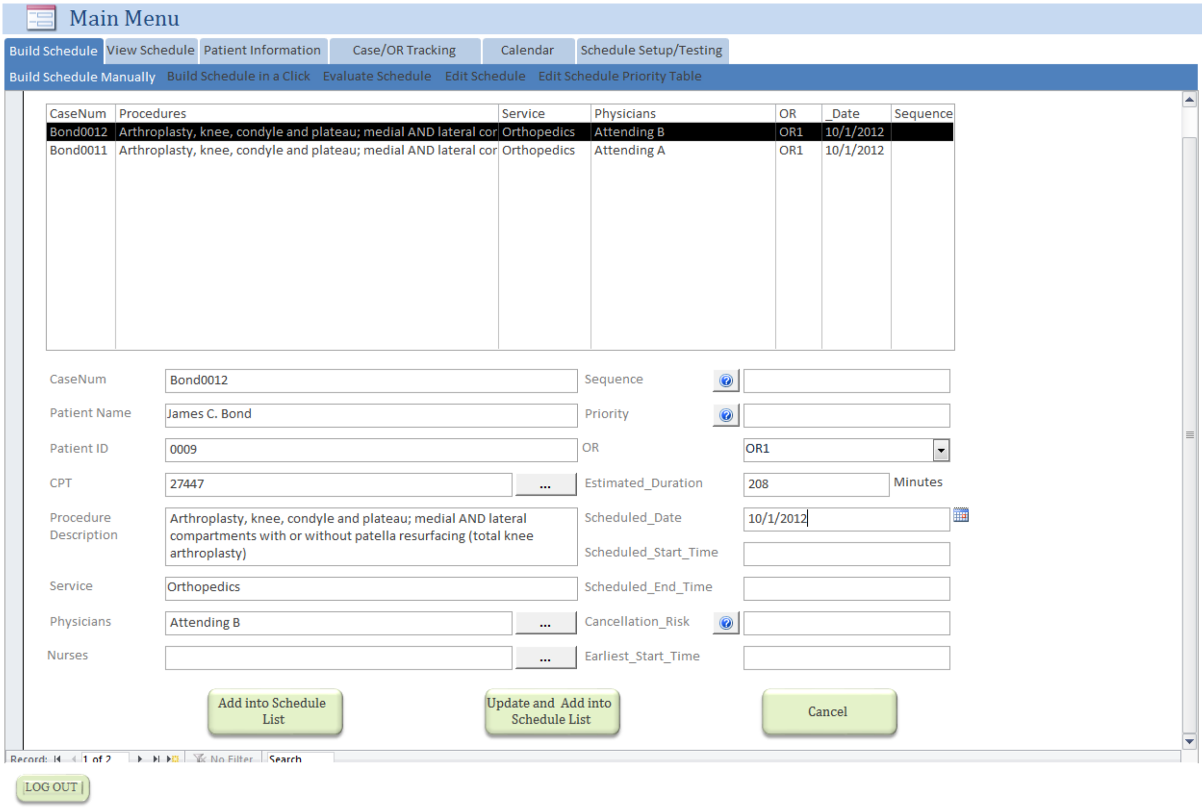



Figure 6.34: Main Menu for Schedulers

### 6.2.3 Physician's Menu

On the physician's menu (illustrated in Figure 6.35), physicians can view schedules, build initial schedule as described in Section 6.1.5, enter and check patient information, track case/OR status and build their own calendar. Unlike the menus for other types of users, the default screen is the view schedule form with their own schedule displayed. They can use the filters on this form to display other schedules of interest as well.

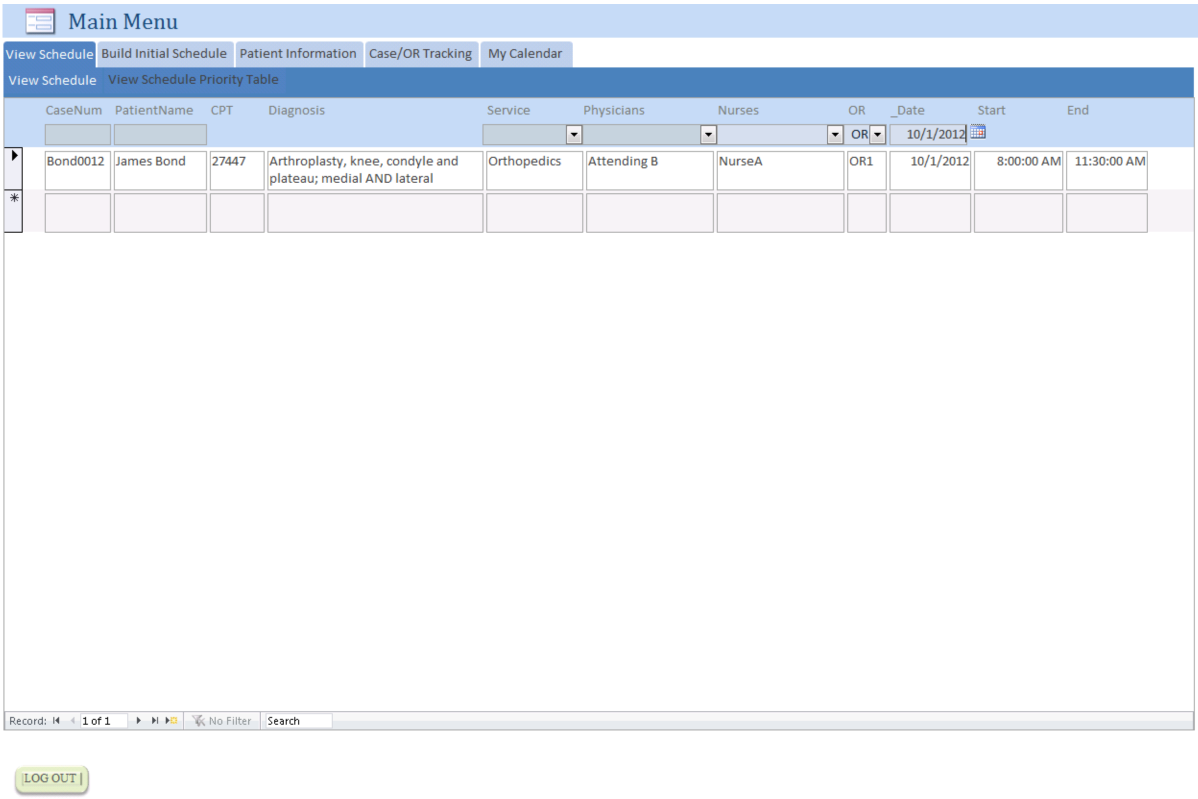



Figure 6.35: Main Menu for Physicians

### 6.2.4 Nurse's Menu

Nurses will be able to update case/OR status, view the schedule, enter or edit patients' information, and check physicians' calendars. One of their main task would be updating the case status, so that staff outside of the OR would know an accurate case status. See the illustration in Figure 6.36

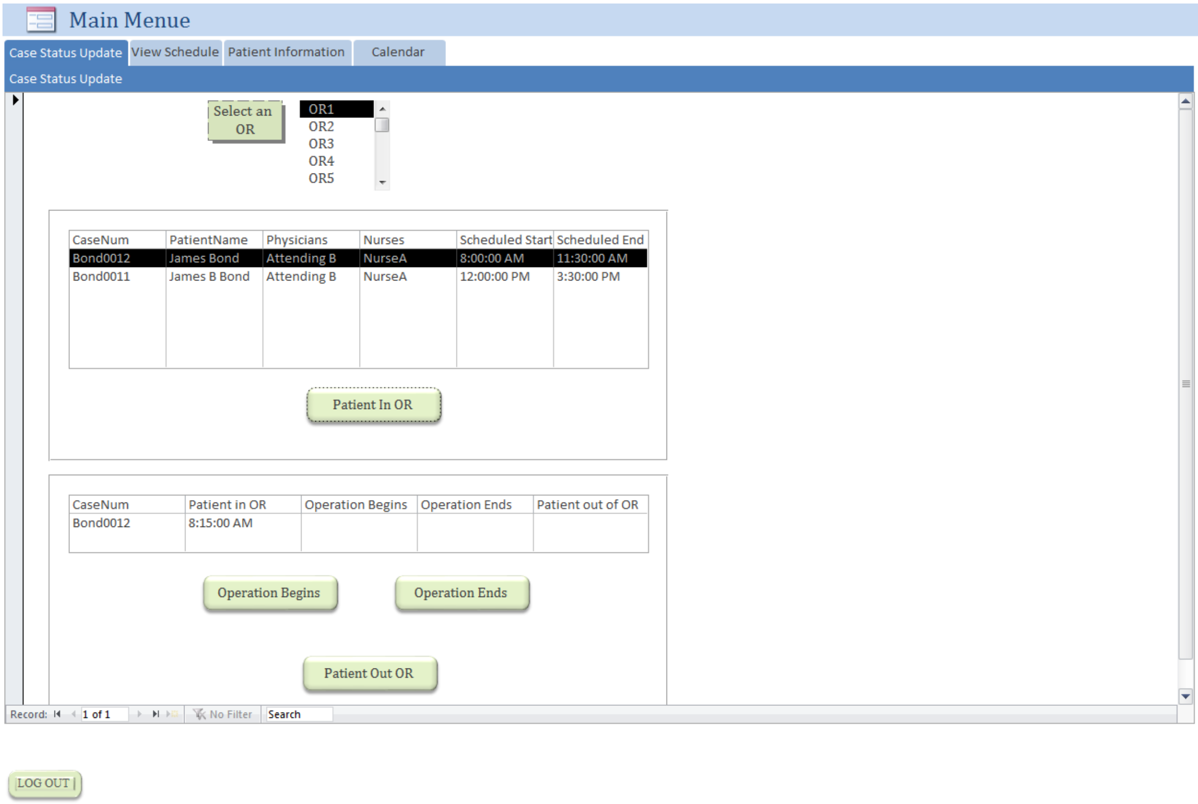



Figure 6.36: Main Menu for Nurses

# CHAPTER 7

# TWO STAGE STOCHASTIC PROGRAMMING MODEL WITH LINKING CONSTRAINTS

In this section, we first generalize and relax the form of the two-stage integer stochastic programming model developed in Section 5.1.2. Then we introduce a Two-Stage L-shaped algorithm to solve the optimal solution for problems with this particular structure.

## 7.1 Problem Definition

The generalized and relaxed two-stage stochastic linear programming model with linking constraints can be defined as follows.

$$
\begin{aligned}
Min \quad & c^T x + \textstyle\sum_{k=1}^{K} p_k q_k^T y_k && (7.1)\\
s.t. \quad & Ax = b, && (7.2)\\
& \textstyle\sum_{k=1}^{K} B_k y_k \leq BG, && (7.3)\\
& T_k x + W_k y_k = h_k, \quad k = 1, \ldots, K, && (7.4)\\
& x \geq 0, y_k \geq 0, \quad k = 1, \ldots, K && (7.5)
\end{aligned}
$$

, where $c$ is a known vector in $\mathbb{R}^{n_1}$, $b$ is a known vector in $\mathbb{R}^{m_1}$, $BG$ is a known vector in $\mathbb{R}^{m_3}$, $A$ is known matrix of size $m_1 \times n_1$. Here we assume there are finite ($K$) realizations, for each possible realization $k \in \{1, \ldots, K\}$, $p_k \in [0, 1]$ is its probability, $q_k \in \mathbb{R}^{n_2}$, $B_k$ is a $m_3 \times n_2$ matrix, $T_k$ is $m_2 \times n_1$, $W_k$ is $m_2 \times n_2$, and $h_k \in \mathbb{R}^{m_2}$. So we have $n_1 + Kn_2$ decision variables and $m_1 + m_3 + Km_2$ linear constraints. It is a large-scale linear programming problem.

Compared with traditional two-stage stochastic programming problems, we have one additional constraint (7.3) which makes our problem quite different. It links all the realizations in the second stage. In other words, in the commonly-seen stochastic programming model, the coefficients corresponding to the decision variables $(x, y_1, \ldots, y_k)$ in the standard formulation form the following structure:

$$\begin{bmatrix} A & & & & \\ T_1 & W & & & \\ T_2 & & W & & \\ \vdots & & & \ddots & \\ T_k & & & & W \end{bmatrix}$$

In our model, the structure becomes:

$$\begin{bmatrix} A & & & & & \\ & B_1 & B_2 & \ldots & B_k & I \\ T_1 & W & & & & \\ T_2 & & W & & & \\ \vdots & & & \ddots & & \\ T_k & & & & W & \end{bmatrix}$$

for decision variables $(x, y_1, \ldots, y_k, slack)$.

Because of the special structure, the existing algorithm (e.g. L-shaped method), which does not consider the linkage between different realizations, does not apply to our problem ([Birge and Louveaux, 1997]). We need to develop a new algorithm to solve problems with this special structure.

In addition to this general form, the first stage decision variables ( $x$ in our problem) may also be integer variables, which gives us an additional integrality constraint:

$$x \in \mathbb{Z}_+^{n_1}. \tag{7.6}$$

## 7.2 Two-Stage L-Shaped Algorithm

In this section, we introduce a new Two-Stage L-shaped algorithm, to solve for problem (7.1)-(7.5). The basic idea of this algorithm is to apply L-shaped method nested within L-shaped method.

We can simplify L-shaped method as the following three steps:

1. Solve the master problem;

2. Solve feasibility sub problems, generate cuts based on dual information of the sub problems, and add the cuts into the master problem;

3. Solve optimality sub problems, generate cuts based on dual information of the sub problems, and add the cuts into the master problem;

The Two-Stage L-shaped algorithm also follows this logic, but since the feasibility sub problems and the optimality sub problems, which we need to solve in Steps 2 and 3, are both two-stage stochastic programming problems, we apply the L-shaped method (all three steps above) in both Steps 2 and 3 for solving these sub problems. The details of this algorithm are listed below.

---

**Algorithm 3** Two-Stage L-Shaped Method

---

*Step 0.* Set $r = s = v = 0$.
*Step 1.* Set $v = v + 1$. Solve the linear program

$$\begin{aligned} Min \quad & c^T x + \theta && (7.7) \\ s.t. \quad & Ax = b, && (7.8) \\ & D_l x \geq d_l \quad l = 1, \ldots, r, && (7.9) \\ & E_l x + \theta \geq e_l, \quad l = 1, \ldots, s, && (7.10) \\ & x \geq 0, \theta \in \mathbb{R}. && (7.11) \end{aligned}$$

Let $(x^v, \theta^v)$ be an optimal solution. If in the first iteration when no constraint (7.10) is present, $\theta^v$ is set equal to $-\infty$ and is not considered in the computation of $x^v$.
*Step 2.*
*Step 2.0* Set $t = w = 0$.
*Step 2.1.* Set $w = w + 1$. Solve the linear program.

$$\begin{aligned} Min \quad & BG^T \gamma + \delta && (7.12) \\ s.t. \quad & F_m \gamma + \delta \geq f_m, \quad m = 1, \ldots, t, && (7.13) \\ & \gamma \geq 0, \delta \in \mathbb{R}. && (7.14) \end{aligned}$$

Let $(\gamma^w, \delta^w)$ be an optimal solution. If no constraint (7.13) is present, $\delta^w$ is set equal to $-\infty$ and is not considered in the computation of $\gamma^w$.
*Step 2.2.* For $k = 1, \ldots, K$ solve the linear program.

$$\begin{aligned} Min \quad & -(h_k - T_k x^v)^T \alpha_k && (7.15) \\ s.t. \quad & W_k^T \alpha_k \leq B_k^T \gamma^w, && (7.16) \\ & I\alpha_k \leq e^T, && (7.17) \\ & -I\alpha_k \leq e^T. && (7.18) \end{aligned}$$

---

**Algorithm 2** Two-Stage L-Shaped Method (continued)

Let ${d_k}^w$ be the simplex multipliers (dual variables) associated with the optimal solution of Problem $k$ of type (7.16). Define

$$F_{m+1} = -\sum_{k=1}^{K} d_k^{wT} B_k^T \tag{7.19}$$

and

$$f_{m+1} = 0. \tag{7.20}$$

If $\delta^w \geq f_{m+1} - F_{m+1}\gamma^w$, continue. Otherwise, set $m = m+1$, add to the constraint set (7.13), and return to *Step 2.1.*
*Step 2.3.* Define

$$D_{r+1} = -\sum_{k=1}^{K} \alpha_k^{wT} T_k \tag{7.21}$$

and

$$d_{r+1} = \sum_{k=1}^{K} \alpha_k^{wT} h_k - \gamma^{wT} BG. \tag{7.22}$$

to generate a constraint (feasibility cut) of type (7.9). Set $r = r+1$, add to the constraint set (7.9)and continue.
*Step 3.*
*Step 3.0* Set $t = w = 0$.
*Step 3.1.* Set $w = w+1$. Solve the linear program.

$$Min \qquad BG^T\sigma + \eta \tag{7.23}$$
$$s.t. \quad G_m\sigma + \eta \geq g_m, \quad m = 1, \dots, t, \tag{7.24}$$
$$\sigma \geq 0, \eta \in \mathbb{R}. \tag{7.25}$$

Let $(\sigma^w, \eta^w)$ be an optimal solution. If no constraint (7.24) is present, $\eta^w$ is set equal to $-\infty$ and is not considered in the computation of $\sigma^w$.
*Step 3.2.* For $k = 1, \dots, K$ solve the linear program.

$$Min \qquad -(h_k - T_k x^v)^T \pi_k \tag{7.26}$$
$$s.t. \quad W_k^T \pi_k \leq B_k^T \sigma^w + p_k q_k. \tag{7.27}$$

Let ${a_k}^w$ be the simplex multipliers associated with the optimal solution of Problem $k$ of type (7.27). Define

$$G_{m+1} = -\sum_{k=1}^{K} a_k^{wT} B_k^T \tag{7.28}$$

and

$$g_{m+1} = \sum_{k=1}^{K} a_k^{wT} p_k q_k. \tag{7.29}$$

**Algorithm 1** Two-Stage L-Shaped Method (continued)

If $\eta^w \geq g_{m+1} - G_{m+1}\sigma^w$, continue. Otherwise, set $m = m+1$, add to the constraint set (7.24), and return to *Step 3.1.*
*Step 3.3.* Define

$$E_{s+1} = \sum_{k=1}^{K} \pi_k^{wT} T_k \tag{7.30}$$

and

$$e_{s+1} = \sum_{k=1}^{K} \pi_k^{wT} h_k - \sigma^{wT} BG. \tag{7.31}$$

If $\theta^v \geq e_{s+1} - E_{s+1}x^v$, stop; $x^v$ is an optimal solution. Otherwise, set $s = s+1$, add to the constraint set (7.10)and return to *Step 1.*

Since in most stochastic programming problems, $n_1 << Kn_2$, we can also use this algorithm to solve for problems with integrality constraints on first stage decision variables. To do this, we can simply add constraints (7.6) into the problem (7.7)-(7.11) solved in Step 1.

## 7.3 Convergence of the Algorithm

Problem (7.1)-(7.5) is equivalent to the so-called deterministic equivalent problem:

$$Min \quad c^T x + Q(x) \tag{7.32}$$

$$s.t. \quad Ax = b, \tag{7.33}$$

$$x \geq 0. \tag{7.34}$$

where

$$Q(x) = min_{\{y_k, k=1,\ldots,K\}}\{\sum_{k=1}^{K} p_k {q_k}^T y_k | \sum_{i=1}^{K} B_i y_i \leq BG, T_k x + W_k y_k = h_k, y_k \geq 0, k = 1,\ldots,K.\} \tag{7.35}$$

We define $K_1 = \{x\,|Ax = b, x \geq 0\}$ be the set determined by the fixed constraints, and $K_2 = \{x\,|Q(x) < \infty\}$ be the second-stage feasibility set. We may now redefine the deterministic equivalent program as follows:

$$Min \quad c^T x + Q(x), \tag{7.36}$$

$$s.t. \quad x \in K_1 \cap K_2. \tag{7.37}$$

Observe that solving (7.36)-(7.37) is equivalent to solving:

$$
\begin{aligned}
Min \quad & c^T x + \theta && (7.38)\\
s.t. \quad & Q(x) \le \theta, && (7.39)\\
& x \in K_1 \cap K_2. && (7.40)
\end{aligned}
$$

We are thus looking for a finitely convergent algorithm for solving (7.36)-(7.37) or (7.38)-(7.40).

**Lemma 5.** *The set $K_2$ is a closed and convex polyhedron.*

*Proof.* By definition, $x \in K_2$ is equivalent to

$$x \in \{x | \, \exists y_{k,k=1,\dots,K} \ge 0 \, s.t. \sum_{i=1}^{K} B_i y_i \le BG, W_k y_k = h_k - T_k x, k = 1, \dots, K.\}$$

, which is defined by a set of linear constraints.

For any two different vectors, $x^1, x^2 \in K_2$, let $x^\lambda = \lambda x^1 + (1-\lambda)x^2$, $\lambda \in (0,1)$. Let $y^1_{k,k=1,\dots,K}$ and $y^2_{k,k=1,\dots,K}$ be some vector in the set $Y = \{y_{k,k=1,\dots,K} | y_{k,k=1,\dots,K} \ge 0, \sum_{i=1}^{K} B_i y_i \le BG, W_k y_k = h_k - T_k x, k = 1, \dots, K.\}$ for $x = x^1$ and $x = x^2$, respectively. Let $y^\lambda_k = \lambda y^1_k + (1-\lambda) y^2_k \ge 0, \forall k = 1, \dots, K$. We can see that $W_k y^\lambda_k = \lambda W_k y^1_k + (1-\lambda) W_k y^2_k = h_k - T_k(\lambda x^1 + (1-\lambda)x^2) = h_k - T_k x^\lambda, \forall k = 1, \dots, K.$ and $\sum_{i=1}^{K} B_i y^\lambda_i = \lambda \sum_{i=1}^{K} B_i y^1_i + (1-\lambda)\sum_{i=1}^{K} B_i y^2_i \le BG$. Thus, $x^\lambda \in K_2$. □

**Lemma 6.** *For a stochastic programming problem in the form of (7.1)-(7.5), $Q(x)$, as defined in (7.35), is a piecewise linear convex function in $X$ for $x$ in $\mathbb{K} = K_1 \cap K_2$.*

*Proof.* For any two different vectors, $x^1$ and $x^2$ in $\mathbb{K}$, let $x^\lambda = \lambda x^1 + (1-\lambda)x^2$, $\lambda \in (0,1)$. It is easy to see that $\mathbb{K}$ is closed and convex, so $x^\lambda \in \mathbb{K}$. Let $y^1_{k,k=1,\dots,K}$ and $y^2_{k,k=1,\dots,K}$ be some optmial solution of $min_{\{y_{k,k=1,\dots,K}\}}\{\sum_{k=1}^{K} p_k {q_k}^T y_k | \sum_{i=1}^{K} B_i y_i \le BG, W_k y_k = h_k - T_k x, y_k \ge 0, k = 1, \dots, K.\}$ for $x = x^1$ and $x = x^2$, respectively. Then $\lambda y^1_{k,k=1,\dots,K} + (1-\lambda) y^2_{k,k=1,\dots,K}$ is a feasible solution of $min_{\{y_{k,k=1,\dots,K}\}}\{\sum_{k=1}^{K} p_k {q_k}^T y_k | \sum_{i=1}^{K} B_i y_i \le BG, W_k y_k = h_k - T_k x^\lambda, y_k \ge 0, k = 1, \dots, K.\}$. Let $y^\lambda_{k,k=1,\dots,K}$ be an optimal solution of this last problem. We thus have

$$
\begin{aligned}
Q(x^\lambda) &= \sum_{k=1}^{K} p_k {q_k}^T y^\lambda_k \le \sum_{k=1}^{K} p_k {q_k}^T (\lambda y^1_k + (1-\lambda) y^2_k) \\
&= \lambda \sum_{k=1}^{K} p_k {q_k}^T y^1_k + (1-\lambda) \sum_{k=1}^{K} p_k {q_k}^T y^2_k = \lambda Q(x^1) + (1-\lambda) Q(x^2).
\end{aligned}
\tag{7.41}
$$

□

For a particular solution $x^v \in K_1$,

$$Q(x^v) = Min \quad \sum_{k=1}^{K} p_k q_k^T y_k \tag{7.42}$$

$$s.t. \quad \sum_{k=1}^{K} B_k y_k \leq BG, \tag{7.43}$$

$$W_k y_k = h_k - T_k x^v, \quad k = 1, \ldots, K, \tag{7.44}$$

$$y_k \geq 0, \qquad k = 1, \ldots, K. \tag{7.45}$$

Assume the optimal simplex multipliers (dual variables) for this problem corresponding to constraint (7.43) and (7.44) are $-\sigma$ and $\pi_{k,k=1,\ldots,K}$, in which $\sigma \in \mathbb{R}^{m_3}$ and $\pi_{k,k=1,\ldots,K} \in \mathbb{R}^{m_2}$, then the dual problem can be written as:

$$(-)Min \quad BG^T\sigma - \sum_{k=1}^{K} (h_k - T_k x^v)^T \pi_k \tag{7.46}$$

$$s.t. \quad -B_k^T\sigma + W_k^T \pi_k \leq p_k q_k, \qquad k = 1, \ldots, K, \tag{7.47}$$

$$\sigma \geq 0 \tag{7.48}$$

It follows from duality in linear programming that,

$$Q(x^v) = -BG^T\sigma + \sum_{k=1}^{K} (h_k - T_k x^v)^T \pi_k. \tag{7.49}$$

By convexity of $Q(x)$ (Lemma 6), it follows that

$$Q(x) \geq -BG^T\sigma + \sum_{k=1}^{K} h_k^T \pi_k - \sum_{k=1}^{K} \pi_k^T T_k x. \tag{7.50}$$

Since $\theta \geq Q(x)$, it follows that a pair $(x, \theta)$ is feasible for (7.38)-(7.40) only if

$$\theta \geq -B^T\sigma + \sum_{k=1}^{K} h_k^T \pi_k - \sum_{k=1}^{K} \pi_k^T T_k x, \tag{7.51}$$

which corresponds to (7.10) where $E_l$ and $e_l$ are defined in (7.30) and (7.31).

On the other hand, if $(x^v, \theta^v)$ is optimal for (7.38)-(7.40), it follows that $Q(x^v) = \theta^v$, because $\theta$ is unrestricted in (7.38)-(7.40) except for $\theta \geq Q(x)$. This happens when $\theta = -BG^T\sigma + \sum_{k=1}^{K} h_k^T \pi_k - \sum_{k=1}^{K} \pi_k^T T_k x^v$, which justifies the termination criterion in Step 3.3.

This means that at each interation, either $\theta^v \geq Q(x^v)$ implying termination, or $\theta^v < Q(x^v)$. In the latter case, none of the already defined optimal cuts (7.10) adequately imposes $\theta \geq Q(x)$, so a new set of multiplier, $\sigma$ and $\pi_{k,k=1,\ldots,K}$, will be defined at $x^v$ to generate an appropriate constraint (7.10). The finite convergence of the algorithm follows from the fact that there is only a finite number of such constraints (7.10) because there are only a finite number of optimal bases to the problem (7.46)-(7.48).

Since problem (7.46)-(7.48) itself is also a large scale problem, solving for $\sigma$ and $\pi_{k,k=1,\ldots,K}$ also requires a fast algorithm. We can treat $\sigma$ as the first stage variable and $\pi_{k,k=1,\ldots,K}$ as the second stage variable, then the problem becomes a two-stage stochastic linear programming problem and we can use the L-shaped method to solve it, which is described in (Step 3.0 - Step 3.2). Notice that the first stage feasibility set of this two-stage stochastic problem problem is $F_1 = \{\sigma | \, \sigma \geq 0\}$ and the second stage feasibility set $F_2$ is equivalent to $\{\sigma | \, \exists \pi_k \, s.t. W_k^T \pi_k \leq p_k q_k + B_k^T \sigma, k = 1, \ldots, K\}$, which is equivalent to $\mathbb{R}^{m_3}$ if $W_k \neq \mathbf{0}, \forall k = 1, \ldots, K$. So $F_2 \supseteq F_1$, which shows that the stochastic program has relatively complete recourse and we do not need to add feasibility cut in the L-shaped method.

We now have to prove that at most a finite number of constraints (7.9) is needed to guarantee $x \in K_2$.

By definition, $x \in K_2$ is equivalent to

$$x \in \{x | \, \exists y_{k,k=1,\ldots,K} \geq 0 \, s.t. \sum_{i=1}^{K} B_i y_i \leq BG, W_k y_k = h_k - T_k x, k = 1, \ldots, K.\}. \tag{7.52}$$

So in Step 2, a sub-problem

$$\begin{aligned}
w' = Min \quad & \textstyle\sum_{k=1}^{K} (e^T v_k^+ + e^T v_k^-) && (7.53)\\
s.t. \quad & \textstyle\sum_{k=1}^{K} B_k y_k \leq BG, && (7.54)\\
& W_k y_k + I v_k^+ - I v_k^- = h_k - T_k x^v, \quad k = 1, \ldots, K, && (7.55)\\
& y_k \geq 0, v_k^+ \geq 0, v_k^- \geq 0 \qquad k = 1, \ldots, K. && (7.56)
\end{aligned}$$

is solved that tests whether $x^v \in K_2$ because $x^v \in K_2$ if and only if $w' = 0$.

Assume the optimal simplex multipliers for this problem corresponding to constraint (7.54) and (7.55) are $-\gamma$ and $\alpha_{k,k=1,\ldots,K}$, in which $\gamma \in \mathbb{R}^{m_3}$ and $\alpha_{k,k=1,\ldots,K} \in \mathbb{R}^{m_2}$, then the dual problem can be written as:

$$\begin{aligned}
(-)Min \quad & BG^T \gamma - \textstyle\sum_{k=1}^{K} (h_k - T_k x^v)^T \alpha_k && (7.57)\\
s.t. \quad & -B_k^T \gamma + W_k^T \alpha_k \leq 0, \qquad k = 1, \ldots, K, && (7.58)\\
& I \alpha_k \leq e^T, \qquad k = 1, \ldots, K, && (7.59)\\
& -I \alpha_k \leq e^T, \qquad k = 1, \ldots, K, && (7.60)\\
& \gamma \geq 0. && (7.61)
\end{aligned}$$

By duality, $w'$ being strictly positive is the same as $BG^T \gamma - \sum_{k=1}^{K} (h_k - T_k x^v)^T \alpha_k < 0$. A necessary condition for $x$ belonging to $K_2$ is that $BG^T\gamma - \sum_{k=1}^{K}(h_k - T_k x)^T\alpha_k \geq 0$. There is at most a finite number of such constraints (7.9) because there are only a finite number of optimal bases to the problem ( 7.57)-( 7.61).

Since problem (7.57)-(7.61) itself is also a large scale problem, we can treat $\gamma$ as the first stage variable and $\alpha_{k,k=1,\ldots,K}$ as the second stage variable, then use L-shaped method to solve the two-stage stochastic programming problem, which is described in (Step 2.0 - Step 2.2). The stochastic program also has relatively complete recourse so we do not need to add feasibility cut in the L-shaped method as well.

We thus have proved the following theorem.

**Theorem 1.** *The Two-Stage L-shaped algorithm finitely converges to an optimal solution when it exists or proves the infeasibility of problem (7.36)- (7.37).*

## 7.4 Numerical Experiments

In this section, we modify and model the well-known Newsvendor problem into a two-stage stochastic programming problem with linking constraints, and use numerical test results to show the characteristics of the algorithm. Since the problem we described in this chapter is a linear programming problem, the Two-Stage L-Shaped algorithm works for problems with no integer variables in the second stage, while the hospital scheduling problem in Section 5.1.2 is an integer programming problem with integer variables in the second stage, we will not use the scheduling problem for numerical tests. We use the Newsvendor problem for numerical test for several reasons: it is a well-known problem; it can be easily modified into a meaningful stochastic programming problem with linking constraints; and most importantly, the optimal solution of the Newsvendor problem can be easily computed, which helps us in studying if the tightness of the added constraints has any effect on the algorithm performance.

All cases were tested using on a Windows 7 environment computer with Intel Core i7 2.93 GHz CPU and 8GB RAM. The Two-Stage L-Shaped Algorithm is implemented using CPLEX Concert Technology with Visual Studio c++ API. We also solved the original problem in the same environment for comparison.

The Newsvendor Problem (or Newsboy Problem) originally described the situation faced by a newspaper vendor who must decide how many copies of the day's paper to stock in the face of uncertain demand and knowing that unsold copies will be worth less at the end of the day.

In the test cases, we assume that the unit purchase price is \$2 and unit retail price is \$6, and the unit salvage price at the end of the day is \$1. When there are no other constraints, the optimal solution $x^*$ of such Newsvendor problem should satisfy:

$$p\{D <= x^*\} = \frac{RetailPrice - Cost}{RetailPrice - SalvagePrice} = 0.8 \quad (7.62)$$

, in which $D$ represents the demand per day, which is a random variable.

Assume that the demand follows a normal distribution with mean of 1000. We can control the variation of the demand by setting the standard deviation to be 10, 100 and 500. Thus the corresponding optimal solution would be around 1008.4, 1084.2 and 1420.8.

Now we modify this original Newsvendor Problem into the problem we discussed in this chapter by introducing constraints. First, we introduce a purchase upper limit as a first-stage constraint, i.e. the amount of copies of daily newspaper the newspaper vendor buys cannot exceed the purchase upper limit. The tightness of the first stage problem can be controlled by setting the purchase upper limit to be 2000 or 1000. Second, we introduce the expected salvage upper limit into the linking constraint, i.e. on average the salvage capacity cannot exceed a salvage upper limit.The tightness of this linking constraint can be controlled by setting the expected salvage upper limit to be 500, 100 or 10.

The size of the problem is controlled by the number of scenarios, which is 10000 or 100000. Without considering the slack variables, the number of variables for the problem equals $2n+1$, where $n$ is the number of scenarios, and the number of constraints equals $2n+2$. The linear programming formulation of the modified Newsvendor Problem is included in Appendix B.

By varying the variation of the demand, the tightness of the both constraints and the number of scenarios generated, we came up with the test cases listed in Table 7.1. The parameters in cases N6I and N6 are the same, except that the decision variable in case N6I is integer. For each test case, we randomly generated 20 datasets, solve the problem using the proposed algorithm for each dataset and compare it with solving the problem in CPLEX. The results are shown in the Table 7.1. The confidence interval is at 95% significance level. The computation time is in seconds.

Table 7.1: Numerical Test Results for Newsvendor Problem

| Index | No. of Scenarios | Purchase Upper Limit | Demand Std. | Salvage Upper Limit | Optimal Solution CI | Algorithm Computation Time CI | CPLEX Computation Time CI | Step 1 Iterations CI | Step 3.2 Iterations CI |
|---|---|---|---|---|---|---|---|---|---|
| N2 | 10000 | 2000 | 10 | 10 | (1008.3, 1008.5) | (0.80, 0.83) | (3.98, 4.60) | (15.9, 16.4) | (32.3, 33.7) |
| N3 | 10000 | 2000 | 100 | 100 | (1083.9, 1085.0) | (0.82, 0.86) | (4.22, 4.52) | (16.2, 16.9) | (35.3, 36.7) |
| N4 | 10000 | 2000 | 100 | 10 | (1042.8, 1043.8) | (1.16, 1.20) | (2.39. 2.53) | (16.6, 17.1) | (54.8, 57.6) |
| N6 | 10000 | 2000 | 500 | 500 | (1418.9, 1424.8) | (0.67, 0.70) | (4.66, 4.90) | (16.1, 16.7) | (33.8, 35.1) |
| N7 | 10000 | 1000 | 10 | 10 | (1000, 1000) | (0.13, 0.13) | (0.30, 0.33) | (3, 3) | (6, 6) |
| N9 | 100000 | 2000 | 100 | 10 | (1042.9, 1043.2) | (4.12, 4.27) | (204.22, 219.67) | (18.1, 18.6) | (60.8, 63.0) |
| N6I | 10000 | 2000 | 500 | 500 | (1419.6, 1425.5) | (1.40, 1.51) | (5.04, 5.18) | (12.9, 13.6) | (27.3, 28.9) |

In all cases, the Two-Stage L-Shaped Algorithm converged to the optimal solution, and the computational time is significantly less than solving the problem directly in CPLEX. In the tests from N2 to N4, the first stage constraints are so loose that the solutions are controlled by the second stage problems. We can see that when the linking constraints are active, the algorithm tends to take longer, while solving the problem in CPLEX tends to take less time. When the solution is mainly controlled by the first stage constraints, both solving the problem using the algorithm and solving the problem in CPLEX take much less time as seen in test case N7. N9 shows that the algorithm is more advantageous as the size of the problem grows, and N6I shows that the algorithm takes less cycles when the first stage variables are integer.

# CHAPTER 8

# CONCLUSIONS AND FUTURE WORK

In this dissertation, we discussed the development of an operating room scheduling support information system that can be generalized and applicable for VA medical centers. Based on the structure of the system, we describe the development in four aspects: data analysis, simulation, scheduling and system design.

In the data analysis part, we focus on three processes with high uncertainty: operating room turnover, surgical procedure operation and patients stay in the PACU and/or ICU. The distributions for the processes are used in simulating the system and generating schedules, and the estimates for turnover time and procedure durations are provided to the users as a guideline for making schedules. We start the data analysis from studying turnover time, which is rarely discussed in the literature. Since there is no direct data for turnover time, we utilize the time gap between two consecutive procedures to obtain turnover time. The steps for getting the filtered time gap data are first described. We consider two factors in estimating turnover time: service type and procedure complexity. Service type is shown to be a classification factor with significant effect in estimating turnover time, and the 20th percentiles of the service type classified time gap data are verified to be a reasonable estimates of the turnover time for each service. As for procedure complexity, we first show that, based on the limited data we have, if two surgeries are major or minor surgeries does not significantly affect the turnover time between them, then we extend the study to show that procedure durations have no significant impact over the turnover time either. Based on assumptions that the filtered time gap is composed of the actual turnover time and the idle time, and the turnover time should follow a bounded distribution, possibly a Uniform distribution or a Triangular distribution, we describe the applicable methods for getting the distribution of turnover time analytically by matching moments or matching the CDF. Since the data of time gaps could contain noise, we also introduced a method for detecting and removing the noise. Better test results are shown for determining the distribution of turnover time using the filtered data. For procedure duration, we conclude that the CPT code is a classification factor with significant effect in describing the duration, and the empirical distribution works better for our data. When estimating the procedure durations, we first study the seasonality and trend of procedure duration due to the learning curve of resident doctors but find the seasonality and trend is neither significant nor consistent. We also study how to obtain an estimate by considering economic effects. The early start of a procedure impacted by the previous procedure's early finish is also taken into consideration. With respect to the distribution of LoS in the PACU/ICU, we find that it is more robust and helpful to pool the LoS to the service level. We also point out that the correlation between procedure duration and LoS in the PACU and ICU should be considered and the Norta method [Cario and Nelson, 1997] should be applied if the correlation is significant.

In the simulation part, we describe our OR centered hospital process including turnover, surgical procedure and the recovering in the PACU. Resources include the OR, PACU, and the surgical team. We demonstrate how to use the simulation model to answer some "what if" questions. Then we discuss how to extend the model to evaluate schedules with both surgeons' procedure duration estimates and procedures' cancelation risk taken into consideration.

In Chapter 5, we first model the single OR scheduling problem in a two stage stochastic integer programming problem with linking budget and expected cancelation constraints that balances between scheduling more procedures and generating more desired schedule by considering the relative costs of OR idle, surgical team's waiting and procedure performing overtime. Our model is flexible enough to include patients' early arrival requirement, surgeon's planned absence, PACU resource limitation, procedures' cancelation outside regular OR operation time and overtime cost control as a character of VA medical centers. How to compute the schedule solution practically is also described and numerical tests results are shown. Then we discuss how to utilize the static model for generating dynamic schedules based on the real time scenarios that occur during the scheduled day.

We describe the functionalities available in the scheduling support information system. The system allows users to set up hospital basic information, test schedule engine settings, create/view/edit patient information, view/edit the priority schedule table, build an initial schedule, generate the final schedule, obtain a dynamic schedule, evaluate schedules, view schedules, edit/view calendar, track/update case status. We consider four types of users based relative to their role in the hospital: administration, scheduler, physician and nurse. Different types of users have different user interfaces and they get access to different functionalities.

Besides the above study in the development of the scheduling system, we also extend the literature in stochastic linear programming. We generalize and relax the static scheduling model we brought up in Chapter 5 into a generalized two-stage stochastic linear programming problem with constraints that link the second stage decisions variables in different scenarios. Such structure has not been studied in the literature before according to the author's knowledge. We develop an algorithm called the Two-Stage L-Shaped method that utilizes the special structure of the problem. The algorithm is proved to converge to the optimal solution. Numerical tests using the Newsvendor Problem shows that the algorithm always converges to the optimal solution significantly faster than solving the problem in CPLEX, and it is especially advantageous when the problem size grows.

Future research can further explore the data analysis part in many aspects. Although our study shows there is no significant relationship between procedure complexity and turnover time, the intuition of such relationship may come for a reason. Such relationships can be explored further with other types of definition and classification of procedure complexity. Another interesting intuition, the seasonality and trend due to residents' learning curve, can also be further explored. Such seasonality and trend may only occur in a certain types of procedures. Future research can address this issue by studying how to identify such characteristics. Due to limitations of the data, we cannot further classify the turnover time, procedure duration and LoS in PACU and ICU to make the estimation and description more accurate. It would be especially helpful if some other factors can be identified to further classify the data. Factors including the surgical team, type of anesthetic, severity of the patient's illness, patient's age, gender, etc. all seem to be reasonable candidates. Here, the whole surgical team, instead of individuals, should be considered as a factor because performing a surgery requires teamwork and the cooperation of the team is important to the procedure duration.

In our simulation model, we only consider three locations: OR, PACU and ICU, and four processes: turnover, performing the surgical procedure and stays in the PACU and ICU. It is designed with consideration of the current available data and the other systems that have already implemented. Future research may extend the model by considering the whole hospital system in a more detailed manner. Locations like wards and processes like preoperative activities and postoperative activities could be considered.

In such an early stage of our operating room scheduling study, we use a simple and practical method for solving the optimal solution of the single OR static scheduling problem. Future research can improve the method by adopting ideas from meta-heuristic or heuristic methods. The ideas can also be incorporated into generating dynamic schedules, for example, instead of utilizing a static model, some rules or simple heuristic steps can be used in generating dynamic schedules under particular scenarios.

An extension of the stochastic linear programming problem with linking constraints will go to the integer case. Since the Two-Stage L-Shaped algorithm we developed originates from the L-Shaped method, which does not apply to problems with integer variables in the second stage, the algorithm only works for the case where there are no integrality constraints on the second stage variables as well. To solve for such integer problem, the ideas in the Integer L-Shaped method could be borrowed and modified.

## REFERENCES


[Aida Jebalia, 2006] Aida Jebalia, Atidel B. Hadj Alouane, P. L. (2006). Operating rooms scheduling. *International Journal of Production Economics*, 99(1-2):52–62.

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

[Baumgart et al., 2007] Baumgart, A., Zoeller, A., Denz, C., Bender, H.-J., Heinzl, A., and Badreddin, E. (2007). Using computer simulation in operating room management: Impacts on process engineering and performance. In *Proceedings of the 40th Annual Hawaii International Conference on System Sciences*, HICSS '07, pages 131–, Washington, DC, USA. IEEE Computer Society.

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

[Belin et al., 2006] Belin, J., Demeulemeester, E., and Cardoen, B. (2006). Visualizing the demand for various resources as a function of the master surgery schedule: A case study. *Journal of Medical Systems*, 30:343–350. 10.1007/s10916-006-9012-5.

[Belin et al., 2009] Belin, J., Demeulemeester, E., and Cardoen, B. (2009). A decision support system for cyclic master surgery scheduling with multiple objectives. *Journal of Scheduling*, 12:147–161. 10.1007/s10951-008-0086-4.

[Birge and Louveaux, 1997] Birge, J. R. and Louveaux, F. (1997). *Introduction to Stochastic Programming*. Springer-Verlag.

[Brailsford, 2005] Brailsford, S. (2005). Overcoming the barriers to implementation of operations research simulation models in healthcare. *Clinical & Investigative Medicine*, 28(6):312 – 315.

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

[Cario and Nelson, 1997] Cario, M. C. and Nelson, B. L. (1997). Modeling and generating random vectors with arbitrary marginal distributions and correlation matrix. Technical report, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Illinois.

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

[Denton et al., 2006] Denton, B., Rahman, A., Nelson, H., and Bailey, A. (2006). Simulation of a multiple operating room surgical suite. In *Simulation Conference, 2006. WSC 06. Proceedings of the Winter*, pages 414 –424.

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

[Dexter et al., 2006] Dexter, F., Davis, M., Halbeis, C. E., Marjamaa, R., Marty, J., McIntosh, C., Nakata, Y., Thenuwara, K. N., Sawa, T., and Vigoda, M. (2006). Mean operating room times differ by 50% among hospitals in different countries for laparoscopic cholecystectomy and lung lobectomy. *Journal of Anesthesia*, 20:319–322. 10.1007/s00540-006-0419-4.

[Dexter et al., 2008] Dexter, F., Dexter, E. U., Masursky, D., and Nussmeier, N. A. (April 2008). Systematic review of general thoracic surgery articles to identify predictors of operating room case durations. *Anesthesia & Analgesia*, 106(4):1232–1241.

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

[Dexter and Traub, 2002] Dexter, F. and Traub, R. D. (2002). How to schedule elective surgical cases into specific operating rooms to maximize the efficiency of use of operating room time. *Anesthesia & Analgesia*, 94(4):933–942.

[Ernst et al., 1977] Ernst, Edward A.and Hoppel, C. L., Lorig, J. L., and Danielson, R. A. (1977). Operating room scheduling by computer. *Anesthesia & Analgesia*, 56(6):831–835.

[Hancock et al., 1988] Hancock, W. M., Walter, P. F., More, R. A., and Glick, N. D. (1988). Operating room scheduling data base analysis for scheduling. *Journal of Medical Systems*, 12:397–409. 10.1007/BF00992688.

[Hanson, 1982] Hanson, K. H. (1982). Computer-assisted operating room scheduling. *Journal of Medical Systems*, 6(3):311–314.

[Harper and Gamlin, 2003] Harper, P. and Gamlin, H. (2003). Reduced outpatient waiting times with improved appointment scheduling: a simulation modelling approach. *OR Spectrum*, 25:207–222. 10.1007/s00291-003-0122-x.

[Harper, 2002] Harper, P. R. (2002). A framework for operational modelling of hospital resources. *Health Care Management Science*, 5:165–173. 10.1023/A:1019767900627.

[HFMA, 2003] HFMA (2003). Achieving operating room efficiency through process integration. *Healthcare financial management journal of the Healthcare Financial Management Association*, 57(3):suppl 1–7 following 112.

[Kalman, 1960] Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME Journal of Basic Engineering*, (82 (Series D)):35–45.

[Marcon and Dexter, 2006] Marcon, E. and Dexter, F. (2006). Impact of surgical sequencing on post anesthesia care unit staffing. *Health Care Management Science*, 9(1):87 – 98.

[Marcon and Dexter, 2007] Marcon, E. and Dexter, F. (2007). An Observational Study of Surgeons Sequencing of Cases and Its Impact on Postanesthesia Care Unit and Holding Area Staffing Requirements at Hospitals. *Anesthesia & Analgesia*, 105(1):119–126.

[Marcon et al., 2003] Marcon, E., Kharraja, S., Smolski, Nicoleand Luquet, B., and Viale, J. P. (2003). Determining the number of beds in the postanesthesia care unit: A computer simulation flow approach. *Anesthesia & Analgesia*, 96(5):1415–1423.

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

[Ozkarahan, 1995] Ozkarahan, I. (1995). Allocation of surgical procedures to operating rooms. *Journal of Medical Systems*, 19:333–352. 10.1007/BF02257264.

[Ozkarahan, 2000] Ozkarahan, I. (2000). Allocation of surgeries to operating rooms by goal programing. *Journal of Medical Systems*, 24:339–378. 10.1023/A:1005548727003.

[Paoletti and Marty, 2007] Paoletti, X. and Marty, J. (2007). Consequences of running more operating theatres than anaesthetists to staff them: a stochastic simulation study. *Anaesth*, 98:462–469.

[Persson and Persson, 2010] Persson, M. and Persson, J. (2010). Analysing management policies for operating room planning using simulation. *Health Care Management Science*, 13(2):182–191.

[VanBerkel and Blake, 2007] VanBerkel, P. and Blake, J. (2007). A comprehensive simulation for wait time reduction and capacity planning applied in general surgery. *Health Care Management Science*, 10(4):373 – 385.

[Weiss, 1990] Weiss, E. N. (1990). Models for Determining Estimated Start Times and Case Orderings In Hospital Operating Rooms. *IIE Transactions*, 22:143–150.

[Wright and Bonar, 1996] Wright, Ian H.and Kooperberg, C. and Bonar, Barbara A.and Bashein, G. (1996). Statistical Modeling to Predict Elective Surgery Time: Comparison with a Computer Scheduling System and Surgeon-provided Estimates. *Anesthesiology*, 85:1235–1245.

# APPENDICES

# Appendix A

# Solving for Parameters in Turnover Time Models

## A.1 Models for Matching Moments

In the following models, we use $C_m$, $C_v$ and $C_s$ to denote the sample mean, variance and skewness.

### A.1.1 1-Uniform + Exponential

Assume the Uniform distribution has positive density function value on the interval 10 to unknown parameter $b$, and the mean of the Exponential distribution is $\mu$. The initial problem for solving the problems is as follows:

$$\begin{aligned} Min \quad & |\mu^2 + \frac{(b-10)^2}{12} - C_v| \\ s.t. \quad & \frac{10+b}{2} + \mu = C_m, \\ & b \geq 10, \\ & \mu \geq 0. \end{aligned}$$

Let $\mu = C_m - \dfrac{10+b}{2}$, we convert the problem into the following forms to solve for $b$:

$$\begin{aligned} Min \quad & |\frac{b^2}{3} + (C_m - \frac{10}{3})b + (C_m - 5)^2 + \frac{25}{3} - C_v| \\ s.t. \quad & b \geq 10, \\ & b \leq 2C_m - 10. \end{aligned}$$

The optimization problem becomes a relatively simple bounded nonlinear programming problem with only one variable.

### A.1.2 1-Uniform + Erlang2

Assume the Uniform distribution has positive density function value on the interval 10 to unknown parameter $b$, and the mean of each Exponential distribution in the two stage Erlang distribution is $\mu$. The initial problem for solving the problems is as follows:

$$\begin{aligned} Min \quad & |2\mu^2 + \frac{(b-10)^2}{12} - C_v| \\ s.t. \quad & \frac{10+b}{2} + 2\mu = C_m, \\ & b \geq 10, \\ & \mu \geq 0. \end{aligned}$$

Let $\mu = \frac{C_m}{2} - \frac{10+b}{4}$, we convert the problem into the following forms to solve for $b$:

$$\begin{aligned} Min \quad & |\frac{5b^2}{24} + (\frac{C_m}{2} - \frac{5}{6})b + \frac{(C_m-5)^2}{2} + \frac{25}{3} - C_v| \\ s.t. \quad & b \geq 10, \\ & b \leq 2C_m - 10. \end{aligned}$$

### A.1.3 1-Uniform + Hypoexponential2

Assume the Uniform distribution has positive density function value on the interval 10 to unknown parameter $b$, and the means of the two Exponential distributions in the two stage Hypoexponential distribution are $\mu_1$ and $\mu_2$. The initial problem for solving the problems is as follows:

$$\begin{aligned} Min \quad & |\frac{2(\mu_1^2 + \mu_2^2)}{C_v^{\frac{3}{2}}} - C_s| \\ s.t. \quad & \frac{10+b}{2} + \mu_1 + \mu_2 = C_m, \\ & \frac{(b-10)^2}{12} + \mu_1^2 + \mu_2^2 = C_v, \\ & b \geq 10, \\ & \mu_1, \mu_2 \geq 0. \end{aligned}$$

It only has solutions when $4C_v \geq (C_m - 10)^2$.

Denote $K_1(b) = C_m - \dfrac{10+b}{2}$ and $K_2(b) = C_v - \dfrac{(b-10)^2}{12}$, which are functions of parameter $b$. Let $\mu_1 = \dfrac{K_1(b) + \sqrt{2K_2(b) - K_1(b)^2}}{2}$ and $\mu_2 = \dfrac{K_1(b) - \sqrt{2K_2(b) - K_1(b)^2}}{2}$, we convert the problem into the following forms to solve for $b$:

$$
\begin{aligned}
Min \quad & \left|\frac{K_1(b)(3K_2(b) - K_1(b)^2)}{C_v^{\frac{3}{2}}} - C_s\right| \\
s.t. \quad & b \geq 10 \\
& b \leq 2C_m - 10, \\
& b \leq 2\sqrt{3C_v} + 10, \\
& b \leq \frac{2(3C_m - 5 + \sqrt{30C_v - 6(C_m - 10)^2})}{5}, \\
& b \geq \frac{2(3C_m - 5 - \sqrt{30C_v - 6(C_m - 10)^2})}{5}, \\
& b \leq \frac{3C_m - 10 - \sqrt{12C_v - 3(C_m - 10)^2}}{2} \text{ or } b \geq \frac{3C_m - 10 + \sqrt{12C_v - 3(C_m - 10)^2}}{2}.
\end{aligned}
$$

### A.1.4 1-Uniform + Lognormal

Assume the Uniform distribution has positive density function value on the interval 10 to unknown parameter $b$, and $\mu$ and $\sigma$ are the mean and standard deviation, respectively, of the variables natural logarithm (by definition, the variables logarithm is normally distributed). The initial problem for solving the problems is as follows:

$$
\begin{aligned}
Min \quad & \left|\frac{(e^{\sigma^2} + 2)(e^{\sigma^2} - 1)^2 e^{3\mu + \frac{3\sigma^2}{2}}}{C_v^{\frac{3}{2}}} - C_s\right| \\
s.t. \quad & \frac{10+b}{2} + e^{\mu + \frac{\sigma^2}{2}} = C_m, \\
& \frac{(b-10)^2}{12} + (e^{\sigma^2} - 1)e^{2\mu + \sigma^2} = C_v, \\
& b \geq 10, \\
& \mu, \sigma \geq 0.
\end{aligned}
$$

Denote $K_1(b) = C_m - \dfrac{10+b}{2}$ and $K_2(b) = C_v - \dfrac{(b-10)^2}{12}$, which are functions of parameter $b$. Let $\mu = \ln(\dfrac{K_1(b)^2}{\sqrt{K_1(b)^2 + K_2(b)}})$ and $\sigma = \sqrt{\ln(\dfrac{K_2(b)}{K_1(b)^2} + 1)}$, we convert the problem into the following forms to solve for $b$:

$$\begin{aligned} Min \quad & |\frac{(\frac{K_2(b)}{K_1(b)^2}+3)\frac{K_2(b)^2}{K_1(b)}}{C_v^{\frac{3}{2}}} - C_s| \\ s.t. \quad & b \geq 10, \\ & b \leq 2C_m - 12, \\ & b \leq 2\sqrt{3C_v} + 10, \\ & \mu, \sigma \geq 0. \end{aligned}$$

### A.1.5 Sym-Triangular +Exponential

Assume the Symmetric Triangular distribution has positive density function value on the interval 10 to unknown parameter $c$, and the mean of the Exponential distribution is $\mu$. The initial problem for solving the problems is as follows:

$$\begin{aligned} Min \quad & |\mu^2 + \frac{(c-10)^2}{24} - C_v| \\ s.t. \quad & \frac{10+c}{2} + \mu = C_m, \\ & c \geq 10, \\ & \mu \geq 0. \end{aligned}$$

Let $\mu = C_m - \frac{10+c}{2}$, we convert the problem into the following forms to solve for $c$:

$$\begin{aligned} Min \quad & |\frac{7c^2}{24} + (C_m - \frac{25}{6})c + (C_m - 5)^2 + \frac{25}{6} - C_v| \\ s.t. \quad & c \geq 10, \\ & c \leq 2C_m - 10. \end{aligned}$$

### A.1.6 Sym-Triangular +Erlang2

Assume the Symmetric Triangular distribution has positive density function value on the interval 10 to unknown parameter $c$, and the mean of each Exponential distribution in the two stage Erlang distribution is $\mu$. The initial problem for solving the problems is as follows:

$$\begin{aligned} Min \quad & |2\mu^2 + \frac{(c-10)^2}{24} - C_v| \\ s.t. \quad & \frac{10+c}{2} + 2\mu = C_m, \\ & c \geq 10, \\ & \mu \geq 0. \end{aligned}$$

Let $\mu = \frac{C_m}{2} - \frac{10+c}{4}$, we convert the problem into the following forms to solve for $c$:

$$\begin{aligned} Min \quad & |\frac{c^2}{6} + (\frac{C_m}{2} - \frac{5}{3})c + \frac{(C_m-5)^2}{2} + \frac{25}{6} - C_v| \\ s.t. \quad & c \geq 10, \\ & c \leq 2C_m - 10. \end{aligned}$$

### A.1.7 Sym-Triangular +Hypoexponential2

Assume the Symmetric Triangular distribution has positive density function value on the interval 10 to unknown parameter $c$, and the means of the two Exponential distributions in the two stage Hypoexponential distribution are $\mu_1$ and $\mu_2$. The initial problem for solving the problems is as follows:

$$\begin{aligned} Min \quad & |\frac{2(\mu_1^2 + \mu_2^2)}{C_v^{\frac{3}{2}}} - C_s| \\ s.t. \quad & \frac{10+c}{2} + \mu_1 + \mu_2 = C_m, \\ & \frac{(c-10)^2}{24} + \mu_1^2 + \mu_2^2 = C_v, \\ & c \geq 10, \\ & \mu_1, \mu_2 \geq 0. \end{aligned}$$

It only has solutions when $7C_v \geq (C_m - 10)^2$.

Denote $K_1(c) = C_m - \frac{10+c}{2}$ and $K_2(c) = C_v - \frac{(c-10)^2}{24}$, which are functions of parameter $c$. Let $\mu_1 = \frac{K_1(c) + \sqrt{2K_2(c) - K_1(c)^2}}{2}$ and $\mu_2 = \frac{K_1(c) - \sqrt{2K_2(c) - K_1(c)^2}}{2}$, we convert the problem into the following forms to solve for $c$:

$$
\begin{aligned}
Min \quad & |\frac{K_1(c)(3K_2(c)-K_1(c)^2)}{C_v^{\frac{3}{2}}} - C_s| \\
s.t. \quad & c \geq 10 \\
& c \leq 2C_m - 10, \\
& c \leq 2\sqrt{6C_v} + 10, \\
& c \leq \frac{3C_m - 10 + \sqrt{24C_v - 3(C_m-10)^2}}{2}, \\
& c \geq \frac{3C_m - 10 - \sqrt{24C_v - 3(C_m-10)^2}}{2}, \\
& c \leq \frac{2(6C_m - 25 - \sqrt{42C_v - 6(C_m-10)^2})}{7} \text{ or } b \geq \frac{2(6C_m - 25 + \sqrt{42C_v - 6(C_m-10)^2})}{7}.
\end{aligned}
$$

### A.1.8 Sym-Triangular +Lognormal

Assume the Symmetric Triangular distribution has positive density function value on the interval 10 to unknown parameter $c$, and $\mu$ and $\sigma$ are the mean and standard deviation, respectively, of the variables natural logarithm (by definition, the variables logarithm is normally distributed). The initial problem for solving the problems is as follows:

$$
\begin{aligned}
Min \quad & |\frac{(e^{\sigma^2}+2)(e^{\sigma^2}-1)^2 e^{3\mu+\frac{3\sigma^2}{2}}}{C_v^{\frac{3}{2}}} - C_s| \\
s.t. \quad & \frac{10+c}{2} + e^{\mu+\frac{\sigma^2}{2}} = C_m, \\
& \frac{(c-10)^2}{24} + (e^{\sigma^2}-1)e^{2\mu+\sigma^2} = C_v, \\
& c \geq 10, \\
& \mu, \sigma \geq 0.
\end{aligned}
$$

Denote $K_1(c) = C_m - \frac{10+c}{2}$ and $K_2(c) = C_v - \frac{(c-10)^2}{24}$, which are functions of parameter $c$. Let $\mu = \ln(\frac{K_1(c)^2}{\sqrt{K_1(c)^2 + K_2(c)}})$ and $\sigma = \sqrt{\ln(\frac{K_2(c)}{K_1(c)^2} + 1)}$, we convert the problem into the following forms to solve for $c$:

$$\begin{aligned}
Min \quad & \left|\frac{\left(\frac{K_2(c)}{K_1(c)^2}+3\right)\frac{K_2(c)^2}{K_1(c)}}{C_v^{\frac{3}{2}}} - C_s\right| \\
s.t. \quad & c \geq 10, \\
& c \leq 2C_m - 12, \\
& c \leq 2\sqrt{6C_v} + 10, \\
& \mu, \sigma \geq 0.
\end{aligned}$$

### A.1.9 2-Uniform + Exponential

Assume the Uniform distribution has positive density function value on the interval $a$ to $b$, and the mean of the Exponential distribution is $\mu$. The initial problem for solving the problems is as follows:

$$\begin{aligned}
Min \quad & \left|\frac{2\mu^3}{C_v^{\frac{3}{2}}} - C_s\right| \\
s.t. \quad & \frac{10+b}{2} + \mu = C_m, \\
& \frac{(b-a)^2}{12} + \mu^2 = C_v, \\
& a, b, \mu \geq 0.
\end{aligned}$$

Let $a = C_m - \mu - \sqrt{3(C_v - \mu^2)}$ and $b = C_m - \mu + \sqrt{3(C_v - \mu^2)}$, we convert the problem into the following forms to solve for $\mu$:

$$\begin{aligned}
Min \quad & \left|\frac{2\mu^3}{C_v^{\frac{3}{2}}} - C_s\right| \\
s.t. \quad & \mu \leq C_m, \\
& \mu \leq \sqrt{C_v}, \\
& \mu \geq \frac{C_m + \sqrt{3(4C_v - C_m^2)}}{4} && \text{, when } C_m^2 \leq 3C_v; \\
& \mu \leq \frac{C_m - \sqrt{3(4C_v - C_m^2)}}{4} \text{ or } \mu \geq \frac{C_m + \sqrt{3(4C_v - C_m^2)}}{4} && \text{, when } 3C_v < C_m^2 \leq 4C_v; \\
& \mu \geq 0 && \text{, when } C_m^2 > 4C_v.
\end{aligned}$$

Notice that the occurrence of the last three constraints depends on the sample data. From the last constraint, we can see that when $3C_v < C_m^2 \leq 4C_v$, we should solve for two optimization problem, one with the constrain $\mu \le \frac{C_m - \sqrt{3(4C_v - C_m^2)}}{4}$ and another with the constrain $\mu \ge \frac{C_m + \sqrt{3(4C_v - C_m^2)}}{4}$, then compare the solutions to find the optimal solution.

### A.1.10 2-Uniform + Erlang2

Assume the Uniform distribution has positive density function value on the interval $a$ to $b$, and the mean of each Exponential distribution in the two stage Erlang distribution is $\mu$. The initial problem for solving the problems is as follows:

$$
\begin{aligned}
Min \quad & |\frac{4\mu^3}{C_v^{\frac{3}{2}}} - C_s| \\
s.t. \quad & \frac{10+b}{2} + 2\mu = C_m, \\
& \frac{(b-a)^2}{12} + \mu^2 = C_v, \\
& a, b, \mu \ge 0.
\end{aligned}
$$

Let $a = C_m - 2\mu - \sqrt{3(C_v - 2\mu^2)}$ and $b = C_m - 2\mu + \sqrt{3(C_v - 2\mu^2)}$, we convert the problem into the following forms to solve for $\mu$:

$$
\begin{aligned}
Min \quad & |\frac{4\mu^3}{C_v^{\frac{3}{2}}} - C_s| \\
s.t. \quad & \mu \le \frac{C_m}{2}, \\
& \mu \le \sqrt{\frac{C_v}{2}}, \\
& \mu \ge \frac{2C_m + \sqrt{6(5C_v - C_m^2)}}{10} \quad \text{, when } C_m^2 \le 3C_v; \\
& \mu \le \frac{2C_m - \sqrt{6(5C_v - C_m^2)}}{10} \text{ or } \mu \ge \frac{2C_m + \sqrt{6(5C_v - C_m^2)}}{10} \quad \text{, when } 3C_v < C_m^2 \le 5C_v; \\
& \mu \ge 0 \quad \text{, when } C_m^2 > 5C_v.
\end{aligned}
$$

### A.1.11 2-Triangular + Exponential

Assume the Triangular distribution has positive density function value on the interval 10 to $c$ with a peak at $b$, and the mean of the Exponential distribution is $\mu$. The initial problem for solving the problems is as follows:

$$
\begin{aligned}
Min \quad & \left|\frac{2\mu^3 + \dfrac{(10+b-2c)(20-b-c)(10-2b+c)}{270}}{C_v^{\frac{3}{2}}} - C_s\right| \\
s.t. \quad & \frac{10+b+c}{3} + \mu = C_m, \\
& \frac{(b-10)^2 + (c-10)^2 + (b-c)^2}{36} + \mu^2 = C_v, \\
& \mu \geq 0, \\
& b, c \geq 10.
\end{aligned}
$$

It only has solutions when $3C_v \geq (C_m - 10)^2$.

Let $b = \dfrac{3C_m - 3\mu - 10 - \sqrt{24(C_v - \mu^2) - 3(C_m - \mu - 10)^2}}{2}$ and

$c = \dfrac{3C_m - 3\mu - 10 + \sqrt{24(C_v - \mu^2) - 3(C_m - \mu - 10)^2}}{2}$, we convert the problem into the following forms to solve for $\mu$:

$$
\begin{aligned}
Min \quad & \left|\frac{2\mu^3 + \dfrac{[(10-C_m+\mu)^2 - 6(C_v-\mu^2)](10-C_m+\mu)}{10}}{C_v^{\frac{3}{2}}} - C_s\right| \\
s.t. \quad & \mu \leq C_m - 10, \\
& \mu \leq \sqrt{C_v}, \\
& \mu \leq \frac{C_m - 10 + \sqrt{72C_v - 8(C_m-10)^2}}{9}, \\
& \mu \geq \frac{C_m - 10 + \sqrt{6C_v - 2(C_m-10)^2}}{3}, \text{when } (C_m-10)^2 \leq 2C_v; \\
& \mu \geq \frac{C_m - 10 + \sqrt{6C_v - 2(C_m-10)^2}}{3} \text{ or} \\
& \mu \leq \frac{C_m - 10 - \sqrt{6C_v - 2(C_m-10)^2}}{3}, \text{when } 2C_v < (C_m-10)^2 \leq 3C_v.
\end{aligned}
$$

### A.1.12 2-Triangular + Erlang2

Assume the Triangular distribution has positive density function value on the interval 10 to $c$ with a peak at $b$, and the mean of each Exponential distribution in the two stage Erlang distribution is $\mu$. The initial problem for solving the problems is as follows:

$$\begin{aligned}
Min \quad & \left|\frac{4\mu^3 + \dfrac{(10+b-2c)(20-b-c)(10-2b+c)}{270}}{C_v^{\frac{3}{2}}} - C_s\right| \\
s.t. \quad & \frac{10+b+c}{3} + 2\mu = C_m, \\
& \frac{(b-10)^2+(c-10)^2+(b-c)^2}{36} + 2\mu^2 = C_v, \\
& \mu \geq 0, \\
& b, c \geq 10.
\end{aligned}$$

It only has solutions when $4C_v \geq (C_m - 10)^2$.

Let $b = \dfrac{3C_m - 6\mu - 10 - \sqrt{24(C_v - 2\mu^2) - 3(C_m - 2\mu - 10)^2}}{2}$ and

$c = \dfrac{3C_m - 6\mu - 10 + \sqrt{24(C_v - 2\mu^2) - 3(C_m - 2\mu - 10)^2}}{2}$, we convert the problem into the following forms to solve for $\mu$:

$$\begin{aligned}
Min \quad & \left|\frac{4\mu^3 + \dfrac{[(10-C_m+2\mu)^2 - 6(C_v-2\mu^2)](10-C_m+2\mu)}{10}}{C_v^{\frac{3}{2}}} - C_s\right| \\
s.t. \quad & \mu \leq \frac{C_m - 10}{2}, \\
& \mu \leq \sqrt{\frac{C_v}{2}}, \\
& \mu \leq \frac{C_m - 10 + 2\sqrt{10C_v - (C_m-10)^2}}{10}, \\
& \mu \geq \frac{C_m - 10 + \sqrt{4C_v - (C_m-10)^2}}{4}, \text{when } (C_m-10)^2 \leq 2C_v; \\
& \mu \geq \frac{C_m - 10 + \sqrt{4C_v - (C_m-10)^2}}{4} \text{ or} \\
& \mu \leq \frac{C_m - 10 - \sqrt{4C_v - (C_m-10)^2}}{4}, \text{when } 2C_v < (C_m-10)^2 \leq 4C_v.
\end{aligned}$$

## A.2 Models for Matching CDF

In the following models, we use $C_m$, $C_v$ and $C_s$ to denote the sample mean, variance and skewness. Assume that there are $N$ observations, we use $x_i$ to denote observation $i$, and $ECDF_i$ to denote the emprical CDF value for observation $i$ computed based on the observations, i.e. $ECDF_i = \dfrac{i}{N}$.

### A.2.1 1-Uniform + Exponential

Assume the Uniform distribution has positive density function value on the interval 10 to unknown parameter $b$, and the mean of the Exponential distribution is $\mu$. The initial problem for solving the problems is as follows:

$$\begin{aligned} Min \quad & \textstyle\sum_i (CDF(x_i) - ECDF_i)^2 \\ s.t. \quad & \frac{10+b}{2} + \mu = C_m, \\ & b \geq 10, \\ & \mu \geq 0. \end{aligned}$$

, in which $CDF(x) = \begin{cases} 0 & , x \leq 10 \\ \frac{1}{10-b}(10 - x + \mu - \mu e^{\frac{10-x}{\mu}}) & , 10 < x \leq b \\ 1 + \mu e^{\frac{b-x}{\mu}} - \frac{1}{10-b} e^{\frac{10-x}{\mu}} & , x \geq b \end{cases}$.

Let $\mu = C_m - \frac{10+b}{2}$, we convert the problem into a one-variable optimization problem to solve for $b$.

### A.2.2 1-Uniform + Erlang2

Assume the Uniform distribution has positive density function value on the interval 10 to unknown parameter $b$, and the mean of each Exponential distribution in the two stage Erlang distribution is $\mu$. The initial problem for solving the problems is as follows:

$$\begin{aligned} Min \quad & \textstyle\sum_i (CDF(x_i) - ECDF_i)^2 \\ s.t. \quad & \frac{10+b}{2} + 2\mu = C_m, \\ & b \geq 10, \\ & \mu \geq 0. \end{aligned}$$

, in which $CDF(x) = \begin{cases} 0 & , x \leq 10 \\ \frac{1}{10-b}[10 - x + 2\mu + (10 - x - 2\mu) e^{\frac{10-x}{\mu}}] & , 10 < x \leq b \\ 1 + \frac{1}{10-b}[(10 - x - 2\mu) e^{\frac{10-x}{\mu}} - (b - x - 2\mu) e^{\frac{b-x}{\mu}}] & , x \geq b \end{cases}$.

Let $\mu = \frac{C_m}{2} - \frac{10+b}{4}$, we convert the problem into a one-variable optimization problem to solve for $b$.

### A.2.3 2-Uniform + Exponential

Assume the Uniform distribution has positive density function value on the interval $a$ to $b$, and the mean of the Exponential distribution is $\mu$. The initial problem for solving the problems is as follows:

$$\begin{aligned} Min \quad & \textstyle\sum_i (CDF(x_i) - ECDF_i)^2 \\ s.t. \quad & \frac{10+b}{2} + \mu = C_m, \\ & \frac{(b-a)^2}{12} + \mu^2 = C_v, \\ & a, b, \mu \geq 0. \end{aligned}$$

, in which $CDF(x) = \begin{cases} 0 & , x \leq a \\ \frac{1}{a-b}(a - x + \mu - \mu e^{\frac{a-x}{\mu}}) & , a < x \leq b \\ 1 + \mu e^{\frac{b-x}{\mu}} - \frac{1}{a-b} e^{\frac{a-x}{\mu}} & , x \geq b \end{cases}$.

Let $a = C_m - \mu - \sqrt{3(C_v - \mu^2)}$ and $b = C_m - \mu + \sqrt{3(C_v - \mu^2)}$, we convert the problem into a one-variable optimization problem to solve for $\mu$.

### A.2.4 2-Uniform + Erlang2

Assume the Uniform distribution has positive density function value on the interval $a$ to $b$, and the mean of each Exponential distribution in the two stage Erlang distribution is $\mu$. The initial problem for solving the problems is as follows:

$$\begin{aligned} Min \quad & \textstyle\sum_i (CDF(x_i) - ECDF_i)^2 \\ s.t. \quad & \frac{10+b}{2} + 2\mu = C_m, \\ & \frac{(b-a)^2}{12} + \mu^2 = C_v, \\ & a, b, \mu \geq 0. \end{aligned}$$

, in which $CDF(x) = \begin{cases} 0 & , x \leq a \\ \frac{1}{a-b}[a - x + 2\mu + (a - x - 2\mu)e^{\frac{a-x}{\mu}}] & , a < x \leq b \\ 1 + \frac{1}{a-b}[(a - x - 2\mu)e^{\frac{a-x}{\mu}} - (b - x - 2\mu)e^{\frac{b-x}{\mu}}] & , x \geq b \end{cases}$.

Let $a = C_m - 2\mu - \sqrt{3(C_v - 2\mu^2)}$ and $b = C_m - 2\mu + \sqrt{3(C_v - 2\mu^2)}$, we convert the problem into a one-variable optimization problem to solve for $\mu$.

# Appendix B

# Formulation for Newsvendor Problem with Linking Constraints

The notations are as follows:

Table B.1: Parameters

| | | |
|---|---|---|
| $k$ | : | index for a scenario (realization) |
| $n$ | : | total number of scenarios |
| $p$ | : | unit purchase price for each copy of newspaper |
| $r$ | : | unit retail price for each copy of newspaper |
| $s$ | : | unit salvage price for each copy of newspaper |
| $PU$ | : | purchase upper limit |
| $SU$ | : | salvage upper limit |
| $D_k$ | : | actual demand for scenario $k$ |

Table B.2: Variables

| | | |
|---|---|---|
| $x$ | : | a decision variable for how many copies of the daily newspaper to stock |
| $y_k^+$ | : | a variable indicating how many copies of the newspaper is sold in scenario $k$ |
| $y_k^-$ | : | a variable indicating how many copies of the newspaper is salvaged in scenario $k$ |

The formulation is as follows:

$$
\begin{aligned}
Max \quad & px + \sum_{k=1}^{n} \frac{1}{k}(ry_k^+ + sy_k^-) \\
s.t. \quad & x \leq PU, \\
& \sum_{k=1}^{n} \frac{1}{k} y_k^- \leq SU, \\
& y_k^+ \leq D_k, \\
& y_k^+ + y_k^- \leq x, \\
& x, y_k^+, y_k^- \geq 0.
\end{aligned}
$$

# Appendix C

# Source Code for Solving Newsvendor Problem with Linking Constraints

All the code are implemented in C++ using Microsoft Visual Studio 2010. IBM ILOG CPLEX 12.4 is called by using CPLEX Concert Technology with Visual Studio c++ API. The code is adjusted to the page size.

## C.1 Source Code for Solving the Problem using CPLEX

```
/* Main function for solving the original problem*/
#include <ilcplex/ilocplex.h>
#include <ilconcert/iloexpression.h>
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <iostream>
#include <random>
#include <ctime>
#include "params.h"
ILOSTLBEGIN

typedef IloArray<IloNumVarArray> NumVarMatrix;
typedef std::tr1::ranlux64_base_01 Myeng;
typedef std::tr1::normal_distribution<double> Mydist;

int main(void)
```

```
{
char log_filenm[100];
strcpy(log_filenm, "results_n9.txt");
int casenum = 0;
ofstream output_file;   //ofstream: stream class to write on files
int data_K = NUM_SCENS;
int max_cut = MAX_CUT;
long my_infi = MY_INFINITY;
double my_tol = MY_TOLERENCE;
double demand_sd = DEMAND_SD;
double demand_mean = 1000;
double p_ul = PURCHANSE_UL;
double s_ul = SALVAGE_UL;
double rand_t = RAND_TIMES;
output_file.open(log_filenm);
output_file << "MY_TOLERENCE: " << my_tol << endl;
output_file << "NUM_SCENS: " << data_K << endl;
output_file << "DEMAND_SD: " << demand_sd << endl;
output_file << "PURCHANSE_UL: " << p_ul << endl;
output_file << "SALVAGE_UL: " << s_ul << endl;
output_file << "RAND_TIMES: " << rand_t << endl;
while(casenum<20){
casenum++;
output_file << "~~~~~~~~~~~:" << endl;
output_file << "Test Case: " << casenum << endl;
long seed_var = casenum*RAND_TIMES;

int data_n1 = 1;
int data_n2 = 2;
int data_m1 = 1;
int data_m2 = 2;
int data_m3 = 1;

double* data_c = (double *)malloc((data_n1)*sizeof(double));
double* data_p = (double *)malloc((data_K)*sizeof(double));
double** data_q = (double **)malloc((data_K*data_n2)*sizeof(double *));
double** data_A = (double **)malloc((data_m1*data_n1)*sizeof(double *));
```

```
double* data_b = (double *)malloc((data_m1)*sizeof(double));
double*** data_B = (double ***)malloc((data_K*data_m3*data_n2)*sizeof(double *));
double* data_BG = (double *)malloc((data_m3)*sizeof(double *));
double*** data_T = (double ***)malloc((data_K*data_m2*data_n1)*sizeof(double *));
double*** data_W = (double ***)malloc((data_K*data_m2*data_n2)*sizeof(double *));
double** data_h = (double **)malloc((data_K*data_m2)*sizeof(double *));

IloEnv env;

int scen, i, j, k;
double seed, temp;
clock_t start, finish;
long int cputime; // returns milliseconds
long int Time_limit=10000000; //10 thousand seconds
//double u1, u2, z1;
//double my_pi = PI;

for(i=0; i< data_m1; i++){
data_A[i] = (double *)malloc((data_n1)*sizeof(double));
}
for(k=0; k< data_K; k++){
data_q[k] = (double *)malloc((data_n2)*sizeof(double));
data_B[k] = (double **)malloc((data_m3*data_n2)*sizeof(double *));
data_T[k] = (double **)malloc((data_m2*data_n1)*sizeof(double *));
data_W[k] = (double **)malloc((data_m2*data_n2)*sizeof(double *));
data_h[k] = (double *)malloc((data_m2)*sizeof(double));
for(i=0; i<data_m3; i++){
data_B[k][i] = (double *)malloc((data_n2)*sizeof(double));
}
for(i=0; i<data_m2; i++){
data_T[k][i] = (double *)malloc((data_n1)*sizeof(double));
data_W[k][i] = (double *)malloc((data_n2)*sizeof(double));
}
}

/*Generate Data*/
```

```
//output_file << "Generate Data Size: " << data_K << endl;
seed = (long)seed_var;
Myeng eng;
eng.seed(seed);
Mydist dist(demand_mean,demand_sd);
//srand(seed_var);
data_c[0]=2;
data_A[0][0]=1;
data_b[0]=PURCHANSE_UL;
data_BG[0]=SALVAGE_UL;
for(scen=0; scen<data_K;scen++){
data_p[scen]=(1/(double)data_K);
data_q[scen][0] = -6;
data_q[scen][1] = -1;
data_B[scen][0][0] = 0;
data_B[scen][0][1] = (1/(double)data_K);
data_T[scen][0][0] = 0;
data_T[scen][1][0] = -1;
data_W[scen][0][0] = 1;
data_W[scen][0][1] = 0;
data_W[scen][1][0] = 1;
data_W[scen][1][1] = 1;

//u1 = rand()/(double(RAND_MAX)+1);
//u2 = rand()/(double(RAND_MAX)+1);
//z1 = double(sqrt(-2 * log(u1)) * sin(2 *my_pi * u2));
//temp = double(abs(demand_mean+z1*demand_sd));
temp = dist(eng);
if (temp<0) temp = -temp;
data_h[scen][0] = temp;
data_h[scen][1] = 0;
}

start = clock () ;
try{
IloModel model(env);
IloNumVarArray x(env, data_n1, 0, my_infi, ILOFLOAT);
```

```
//IloNumVarArray x(env, data_n1, 0, my_infi, ILOINT);
NumVarMatrix y(env, data_K);
IloExpr obj(env);
for(i = 0; i < data_K; i++){
y[i] = IloNumVarArray(env, data_n2, 0, my_infi, ILOFLOAT);
}
for(i = 0; i < data_n1; i++){
obj += data_c[i]*x[i];
}
for(k = 0; k < data_K; k++){
for(j = 0; j< data_n2; j++){
obj += data_p[k]*data_q[k][j]*y[k][j];
}
}
model.add(IloMinimize(env,obj));

for(i = 0; i < data_m1; i++) {
IloExpr v(env);
for(j = 0; j < data_n1; j++){
v += data_A[i][j]*x[j];
}
model.add(v <= data_b[i]);
v.end();
}
for(i = 0; i < data_m3; i++) {
IloExpr v(env);
for(j = 0; j < data_n2; j++){
for(k=0; k<data_K; k++){
v += data_B[k][i][j]*y[k][j];
}
}
model.add(v <= data_BG[i]);
v.end();
}
for(k = 0; k < data_K; k++){
for(i = 0; i < data_m2; i++){
IloExpr v(env);
```

```
for(j = 0; j < data_n1; j++){
v += data_T[k][i][j]*x[j];
}
for(j = 0; j < data_n2; j++){
v += data_W[k][i][j]*y[k][j];
}
model.add(v <= data_h[k][i]);
v.end();
}
}
   IloCplex cplex(model);
   cplex.setParam(cplex.Threads, 1);

   if ( !cplex.solve() ) {
  env.error() << "Failed to optimize LP." << endl;
  throw(-1);
   }
   finish = clock ();
   cputime = finish - start;

   double obj_value = cplex.getObjValue();
   int status = cplex.getStatus();
   IloNumArray vals(env, data_n1);
   cplex.getValues(vals, x);
   //output_file << "Parameter h: " << endl;
   //for(i=0; i<data_K; i++){
   //output_file << data_h[i][0] << endl;
   //}
   //env.out() << "Solution status = " << status << endl;
   //env.out() << "Objective value = " << obj_value << endl;
   //env.out() << "Values = " << vals << endl;
   //env.out() << "CPU Time = " << cputime << endl;
   output_file << "Solution status = " << status << endl;
   output_file << "Objective value = " << obj_value << endl;
   output_file << "Values = " << vals << endl;
   output_file << "CPU Time = " << cputime << endl;
```

```
}
catch (IloException& e) {
  cerr << "Concert exception caught: " << e << endl;
}
catch (...) {
cerr << "Unknown exception caught" << endl;
}

env.end();

free(data_c);
free(data_p);
free(data_b);
free(data_BG);
for(i=0; i< data_m1; i++){
free(data_A[i]);
}
for(k=0; k< data_K; k++){
for(i=0; i<data_m3; i++){
free(data_B[k][i]);
}
for(i=0; i<data_m2; i++){
free(data_T[k][i]);
free(data_W[k][i] );
}
free(data_q[k]);
free(data_B[k]);
free(data_T[k]);
free(data_W[k]);
free(data_h[k]);

}
free(data_A);
free(data_q);
free(data_B);
free(data_T);
free(data_W);
```

```
free(data_h);
}
output_file.close();
return 0;
}
```

## C.2 Source Code for Solving the Problem using Two-Stage L-Shaped Method

```
/* Main function for solving the general_two_stage_L problem*/
/*
  Current code only work for the case that only h differs in each scenario
  For other case: need to modify the initialization of the data,
 definition of the subsub parameters,
  the way subsub problem is solved.
*/
#include <ilcplex/ilocplex.h>
#include <ilconcert/iloexpression.h>
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <iostream>
#include <random>
#include <ctime>
#include "params.h"
ILOSTLBEGIN

typedef IloArray<IloNumVarArray> NumVarMatrix;
typedef std::tr1::ranlux64_base_01 Myeng;
typedef std::tr1::normal_distribution<double> Mydist;

int main(void)
{
char log_filenm[100];
strcpy(log_filenm, "results_n9.txt");
```

```
int casenum = 0;
ofstream output_file;   //ofstream: stream class to write on files
int data_K = NUM_SCENS;
int max_cut = MAX_CUT;
long my_infi = MY_INFINITY;
double my_tol = MY_TOLERENCE;
double demand_sd = DEMAND_SD;
double demand_mean = 1000;
double p_ul = PURCHANSE_UL;
double s_ul = SALVAGE_UL;
double rand_t = RAND_TIMES;
output_file.open(log_filenm);
output_file << "MY_TOLERENCE: " << my_tol << endl;
output_file << "NUM_SCENS: " << data_K << endl;
output_file << "DEMAND_SD: " << demand_sd << endl;
output_file << "PURCHANSE_UL: " << p_ul << endl;
output_file << "SALVAGE_UL: " << s_ul << endl;
output_file << "RAND_TIMES: " << rand_t << endl;
while(casenum<20){
casenum++;
output_file << "~~~~~~~~~~~:" << endl;
output_file << "Test Case: " << casenum << endl;
long seed_var = casenum*RAND_TIMES;

int data_n1 = 1;
int data_n2 = 2;
int data_m1 = 1;
int data_m2 = 2;
int data_m3 = 1;

double* data_c = (double *)malloc((data_n1)*sizeof(double));
double* data_p = (double *)malloc((data_K)*sizeof(double));
double** data_q = (double **)malloc((data_K*data_n2)*sizeof(double *));
double** data_A = (double **)malloc((data_m1*data_n1)*sizeof(double *));
double* data_b = (double *)malloc((data_m1)*sizeof(double));
double*** data_B = (double ***)malloc((data_K*data_m3*data_n2)*sizeof(double *));
double* data_BG = (double *)malloc((data_m3)*sizeof(double *));
```

```
double*** data_T = (double ***)malloc((data_K*data_m2*data_n1)*sizeof(double *));
double*** data_W = (double ***)malloc((data_K*data_m2*data_n2)*sizeof(double *));
double** data_h = (double **)malloc((data_K*data_m2)*sizeof(double *));

IloEnv env;

//char model_filenm[100];
int scen, i, j, k;
double seed, temp;
clock_t start, finish;
long int cputime = 0; // returns milliseconds
long int Time_limit=10000000; //10 thousand seconds
//double u1, u2, z1;
//double my_pi = PI;

//strcpy(model_filenm, "submodels.lp");

for(i=0; i< data_m1; i++){
data_A[i] = (double *)malloc((data_n1)*sizeof(double));
}
for(k=0; k< data_K; k++){
data_q[k] = (double *)malloc((data_n2)*sizeof(double));
data_B[k] = (double **)malloc((data_m3*data_n2)*sizeof(double *));
data_T[k] = (double **)malloc((data_m2*data_n1)*sizeof(double *));
data_W[k] = (double **)malloc((data_m2*data_n2)*sizeof(double *));
data_h[k] = (double *)malloc((data_m2)*sizeof(double));
for(i=0; i<data_m3; i++){
data_B[k][i] = (double *)malloc((data_n2)*sizeof(double));
}
for(i=0; i<data_m2; i++){
data_T[k][i] = (double *)malloc((data_n1)*sizeof(double));
data_W[k][i] = (double *)malloc((data_n2)*sizeof(double));
}
}

/*Generate Data*/
```

```
//output_file << "Generate Data Size: " << data_K << endl;
//output_file << "Generate Data Size: " << data_K << endl;
seed = (long)seed_var;
Myeng eng;
eng.seed(seed);
Mydist dist(demand_mean,demand_sd);
//srand(seed_var);
data_c[0]=2;
data_A[0][0]=1;
data_b[0]=PURCHANSE_UL;
data_BG[0]=SALVAGE_UL;
for(scen=0; scen<data_K;scen++){
data_p[scen]=(1/(double)data_K);
data_q[scen][0] = -6;
data_q[scen][1] = -1;
data_B[scen][0][0] = 0;
data_B[scen][0][1] = (1/(double)data_K);
data_T[scen][0][0] = 0;
data_T[scen][1][0] = -1;
data_W[scen][0][0] = 1;
data_W[scen][0][1] = 0;
data_W[scen][1][0] = 1;
data_W[scen][1][1] = 1;
//u1 = rand()/(double(RAND_MAX)+1);
//u2 = rand()/(double(RAND_MAX)+1);
//z1 = double(sqrt(-2 * log(u1)) * sin(2 *my_pi * u2));
//temp = double(abs(demand_mean+z1*demand_sd));
temp = dist(eng);
if (temp<0) temp = -temp;
data_h[scen][0] = temp;
data_h[scen][1] = 0;
}
/*generate mean parameters*/
double* h_mean = (double *)malloc((data_m2)*sizeof(double));
double** T_mean = (double **)malloc((data_m2*data_n1)*sizeof(double *));
double** W_mean = (double **)malloc((data_m2*data_n2)*sizeof(double *));
double** B_mean = (double **)malloc((data_m3*data_n2)*sizeof(double *));
```

```
double* pq_mean = (double *)malloc((data_n2)*sizeof(double));
for(i=0; i<data_m2; i++){
T_mean[i] = (double *)malloc((data_n1)*sizeof(double));
W_mean[i] = (double *)malloc((data_n2)*sizeof(double));
}
for(i=0; i<data_m3; i++){
B_mean[i] = (double *)malloc((data_n2)*sizeof(double));
}
for(i=0;i<data_m2;i++){
h_mean[i]=0;
for(j=0; j<data_n1; j++){
T_mean[i][j]=0;
}
for(j=0; j<data_n2; j++){
W_mean[i][j]=0;
}
}
for(i=0;i<data_n2;i++){
pq_mean[i]=0;
for(j=0; j<data_m3; j++){
B_mean[j][i]=0;
}
}
for(k=0; k<data_K; k++){
for(i=0; i<data_m2; i++){
h_mean[i] += data_h[k][i];
for(j=0; j<data_n1; j++){
T_mean[i][j] += data_T[k][i][j];
}
for(j=0; j<data_n2; j++){
W_mean[i][j] += data_W[k][i][j];
}
}
for(i=0;i<data_n2;i++){
pq_mean[i] += data_p[k]*data_q[k][i];
for(j=0; j<data_m3; j++){
B_mean[j][i] += data_B[k][j][i];
```

```
}
}
}
for(i=0;i<data_m2;i++){
h_mean[i] /= data_K;
for(j=0; j<data_n1; j++){
T_mean[i][j] /= data_K;
}
for(j=0; j<data_n2; j++){
W_mean[i][j] /= data_K;
}
}
for(i=0;i<data_n2;i++){
pq_mean[i] /= data_K;
for(j=0; j<data_m3; j++){
B_mean[j][i] /= data_K;
}
}

/* assign memory for master problem*/
IloModel mastermodel(env);
IloCplex mastercplex(env);
mastercplex.setParam( mastercplex.Threads, 1);
double** masterCutE = (double **)malloc((max_cut*data_n1)*sizeof(double *));
for(i=0; i<max_cut; i++){
masterCutE[i] = (double *)malloc((data_n1)*sizeof(double));
}
double* masterCute = (double *)malloc((max_cut)*sizeof(double));
int master_cut_status = 0;
/*0 for adding no cut; 1 for adding cut; 2 for duplicated cut*/
int master_cut_num = 0;
double master_obj_value;
double current_THETA = -my_infi;
double* sx = (double *)malloc((data_n1)*sizeof(double));
double master_LB = -my_infi;
int master_status;
int master_opt = 0;
```

```
int stop = 0;
int master_iter = 0;

/*assign memory for sub problem*/
double** subCutG = (double **)malloc((max_cut*data_m3)*sizeof(double *));
for(i=0; i<max_cut; i++){
subCutG[i] = (double *)malloc((data_m3)*sizeof(double));
}
double* subCutg = (double *)malloc((max_cut)*sizeof(double));
double* sSIGMA = (double *)malloc((data_m3)*sizeof(double));
double sub_opt;
int sub_iter = 0;

/*assign memory for subsub problems*/
int lhs_column_size = data_n2+data_m2;
double** sspi = (double **)malloc((data_K*data_m2)*sizeof(double *));
double** ssdual = (double **)malloc((data_K*data_n2)*sizeof(double *));
double** lhs = (double **)malloc((data_n2*(lhs_column_size))*sizeof(double *));
double* rhs = (double *)malloc((data_n2)*sizeof(double));
double* rdc = (double *)malloc((lhs_column_size+1)*sizeof(double));
double* sscost = (double *)malloc((lhs_column_size)*sizeof(double));
double* Tx = (double *)malloc((data_m2)*sizeof(double));
int* basis = (int *)malloc((data_n2)*sizeof(int));
double total_subsub;

for(k=0; k<data_K; k++){
sspi[k] = (double *)malloc((data_m2)*sizeof(double));
ssdual[k] = (double *)malloc((data_n2)*sizeof(double));
}
for(i=0; i<data_n2; i++){
lhs[i] = (double *)malloc((lhs_column_size)*sizeof(double));
}

start = clock () ;
/*initialize master problem*/
IloNumVarArray x(env, data_n1, 0, my_infi, ILOFLOAT);
//IloNumVarArray x(env, data_n1, 0, my_infi, ILOINT);
```

```
IloNumVar THETA(env, -my_infi, my_infi, ILOFLOAT);
IloExpr masterobj(env);
masterobj += THETA;
for(i = 0; i < data_n1; i++){
masterobj += data_c[i]*x[i];
}
mastermodel.add(IloMinimize(env,masterobj));
masterobj.end();
for(i = 0; i < data_m1; i++) {
IloExpr v(env);
for(j = 0; j < data_n1; j++){
v += data_A[i][j]*x[j];
}
mastermodel.add(v <= data_b[i]);
v.end();
}


while(!master_opt && !stop){


if(master_cut_num >= max_cut){
stop = 1;
output_file << "number of cuts in MASTER problem is out of the maximal!" << endl;
break;
}
/*Step 1: solve master problem*/
try{

if(master_cut_status == 1){
IloExpr v(env);
for(i=0; i<data_n1; i++){
v += masterCutE[master_cut_num-1][i]*x[i];
}
v += THETA;
mastermodel.add(v >=masterCute[master_cut_num-1]);
}
```

```
mastercplex.extract(mastermodel);

if ( !mastercplex.solve() ) {
   env.error() << "Failed to optimize the master LP." << endl;
   throw(-1);
}

    master_iter ++;
master_obj_value = mastercplex.getObjValue();
master_status = mastercplex.getStatus();
current_THETA = mastercplex.getValue(THETA);
//output_file << "Master Solution status = " << master_status << endl;
//output_file << "Master Objective value = " << master_obj_value << endl;
//output_file << "Master solution THETA: = " << current_THETA << endl;
//output_file << "Master solution x: = " << endl;
for(i=0; i<data_n1; i++){
sx[i] = mastercplex.getValue(x[i]);
//output_file << sx[i] << endl;
}


}/*try*/
catch (IloException& e) {
  cerr << "Concert exception caught in master problem: " << e << endl;
}
catch (...) {
cerr << "Unknown exception caught in master problem" << endl;
}

if(current_THETA > master_LB){
master_LB = current_THETA;
}

/* Step 3 of two stage L-shaped method*/
/* define sub problem*/
IloModel submodel(env);
IloCplex subcplex(env);
```

```
subcplex.setParam(subcplex.Threads, 1);
int sub_cut_status = 0;
/*0 for adding no cut; 1 for adding cut; 2 for duplicated cut*/
int sub_cut_num = 0;
double sub_obj_value;
double current_DELTA = -my_infi;
double sub_LB = -my_infi;
int sub_status;
double temp;
int subsub_solve = 0;

IloNumVarArray SIGMA(env, data_m3, 0, my_infi, ILOFLOAT);
IloNumVarArray pi_mean(env, data_m2, 0, my_infi, ILOFLOAT);
IloNumVar DELTA(env, -my_infi, my_infi, ILOFLOAT);
IloNumArray sub_var(env);
IloExpr subobj(env);
subobj += DELTA;
for(i = 0; i < data_m3; i++){
subobj += data_BG[i]*SIGMA[i];
}
submodel.add(IloMinimize(env,subobj));
subobj.end();

for(i = 0; i < data_n2; i++) {
IloExpr v(env);
for(j = 0; j < data_m2; j++){
v -= W_mean[j][i]*pi_mean[j];
}
for(j = 0; j < data_m3; j++){
v -= B_mean[j][i]*SIGMA[j];
}
submodel.add(v <= pq_mean[i]);
v.end();
}
IloExpr tempv(env);
for(i = 0; i < data_m2; i++){
tempv += h_mean[i]*pi_mean[i];
```

```
for(j = 0; j < data_n1; j++){
temp = T_mean[i][j]*sx[j];
tempv -= temp*pi_mean[i];
}
}
submodel.add(tempv >= DELTA);
tempv.end();

sub_opt = 0;
while(!sub_opt && !stop){
/*solve sub problem*/
if(sub_cut_num >= max_cut){
stop = 1;
output_file << "number of cuts in SUB problem is out of the maximal!" << endl;
break;
}
try{

if(sub_cut_status == 1){
IloExpr v(env);
for(i=0; i<data_m3; i++){
v += subCutG[sub_cut_num-1][i]*SIGMA[i];
}
v += DELTA;
submodel.add(v >=subCutg[sub_cut_num-1]);
}

subcplex.extract(submodel);
if ( !subcplex.solve() ) {
   env.error() << "Failed to optimize the sub LP." << endl;
   throw(-1);
}

sub_iter ++;
sub_obj_value = subcplex.getObjValue();
sub_status = subcplex.getStatus();
current_DELTA = subcplex.getValue(DELTA);
```

```
//output_file << "Sub Solution status = " << sub_status << endl;
//output_file << "Sub Objective value = " << sub_obj_value << endl;
//output_file << "Sub solution DELTA: = " << current_DELTA << endl;
//output_file << "Sub solution SIGMA: = " << endl;

subcplex.getValues(sub_var, SIGMA);
for(i=0; i<data_m3; i++){
sSIGMA[i] = sub_var[i];
//output_file << sSIGMA[i] << endl;
}
//subcplex.exportModel(model_filenm);
}/*try*/
catch (IloException& e) {
  cerr << "Concert exception caught in sub problem: " << e << endl;
}
catch (...) {
cerr << "Unknown exception caught in sub problem" << endl;
}

if (sub_obj_value > sub_LB){
sub_LB = sub_obj_value;
}

/*prepare the subsub problem*/
for(i=0; i<data_n2; i++){
for(j=0; j<data_m2; j++){
lhs[i][j] = -data_W[0][j][i];
}
for(j=data_m2; j<data_m2+data_n2; j++){
if(j==i+data_m2){
lhs[i][j] = 1;
}
else lhs[i][j] = 0;
}
}
//output_file << "Subsub rhs: " << endl;
for(i=0; i<data_n2; i++){
```

```
rhs[i] = data_p[0]*data_q[0][i];
for(j=0; j<data_m3; j++){
rhs[i] += data_B[0][j][i]*sSIGMA[j];
}
//output_file << rhs[i] << endl;
}
for(i=0; i<data_m2; i++){
Tx[i] = 0;
for(j=0; j<data_n1; j++){
Tx[i] += data_T[0][i][j]*sx[j];
}
}
for(i=0; i<data_n2; i++){
basis[i] = data_m2+i;
}
total_subsub = 0;

int prob = 0;
/*solve the first problem*/
for(i=0; i<data_m2; i++){
rdc[i] = data_h[prob][i]-Tx[i];
}
for(i=data_m2; i<data_m2+data_n2; i++){
rdc[i] = 0;
}
subsub_solve = 0;
while(!subsub_solve){
/*check whether optimal, if not, which basis should be out*/
int outv = -1;
subsub_solve = 1;
while(outv<data_n2-1 && subsub_solve){
outv++;
if(rhs[outv]<0) subsub_solve = 0;
}
if(subsub_solve){
/*store the solutions*/
for(i=0; i<data_m2; i++){
```

```
sspi[prob][i] = 0;
}
for(i=0; i<data_n2;i++){
if(basis[i]<data_m2) sspi[prob][basis[i]]=rhs[i];
ssdual[prob][i] = -rdc[data_m2+i];
}
total_subsub -= rdc[data_m2+data_n2];

}
else{
/*find the negative component: let basis outv into the basis*/
int inv = -1;
double temp1 = 0;
temp = my_infi;
for(i=0; i<data_m2+data_n2; i++){
if(lhs[outv][i]<0){
temp1 = -rdc[i]/(double)lhs[outv][i];
if(temp1<temp){
temp = temp1;
inv = i;
}
}
}
if(inv<0){
subsub_solve = 1;
//output_file << "Subsub problem " << prob  << " Unbounded!" << endl;
for(i=0; i<data_m2; i++){
sspi[prob][i] = my_infi;
}
for(i=0; i<data_n2;i++){
ssdual[prob][i] = my_infi;
}
total_subsub -= my_infi;
}/*if(inv<0)*/
else{
/*pivot*/
//output_file << "Subsub out index: " << outv << endl;
```

```
//output_file << "Subsub out basis: " << basis[outv] << endl;
//output_file << "Subsub in basis: " << inv << endl;
//output_file << "Subsub reduced cost: " << endl;
basis[outv] = inv;
temp = lhs[outv][inv];
for(j=0; j<data_n2+data_m2; j++){
lhs[outv][j] /= temp;
}
rhs[outv] /= temp;
for(i=0; i<data_n2; i++){
if(i != outv){
temp = lhs[i][inv];
for(j=0; j<data_n2+data_m2; j++){
lhs[i][j] -= temp*lhs[outv][j];
}
rhs[i] -= temp*rhs[outv];
}
}

temp = rdc[inv];
//output_file << "multiplier for rdc: " << temp <<endl;
for(j=0; j<data_n2+data_m2; j++){
rdc[j] -= (double)(temp*lhs[outv][j]);
}
rdc[data_n2+data_m2] -= (double)(temp*rhs[outv]);
//for(j=0; j<data_n2+data_m2+1; j++){
//output_file << rdc[j] << endl;
//}

//output_file << "Subsub lhs: " << endl;
//for(i=0;i<data_n2;i++){
//for(j=0; j<data_n2+data_m2;j++)
//output_file << lhs[i][j] << endl;
//output_file << "  ... end row." << endl;
//}
//output_file << "Subsub rhs: " << endl;
//for(i=0;i<data_n2;i++){
```

```
//output_file << rhs[i] << endl;
//}
}/*else for if(inv<0)*/

}/*else for if(subsub_solve)*/

}/*while(!subsub_solve)*/

/*solve the following problem using known basis*/
prob = 1;
while(prob <data_K){ /*solve each subsub problem*/
//output_file << "SUBSUB problem " << prob << " :"<< endl;
//output_file << "SUBSUB cost: " <<endl;
for(i=0; i<data_m2; i++){
sscost[i] = data_h[prob][i]-Tx[i];
//output_file << data_h[prob][i] << "  " << sscost[i] <<endl;
rdc[i] = sscost[i];
}
for(i=data_m2; i<data_m2+data_n2; i++){
sscost[i] = 0;
rdc[i] = 0;
}
//output_file << "SUBSUB basis: " <<endl;
//for(i=0; i<data_n2; i++){
//output_file << basis[i]<<endl;
//}
for(i=0; i<data_m2+data_n2; i++){
for(j=0; j<data_n2; j++){
rdc[i] -= (double)(sscost[basis[j]]*lhs[j][i]);
}
}
//output_file << "SUBSUB reduced cost: " <<endl;
//for(j=0; j<data_m2+data_n2;j++){
//output_file << rdc[j] <<endl;
//}
i = data_m2+data_n2;
rdc[i] = 0;
```

```
for(j=0; j<data_n2; j++){
rdc[i] -= sscost[basis[j]]*rhs[j];
}
//output_file << rdc[data_m2+data_n2] <<endl;

subsub_solve = 0;
while(!subsub_solve){
/*check whether optimal, if not, which basis should be in*/
int inv = -1;
subsub_solve = 1;
while(inv<data_n2+data_m2-1 && subsub_solve){
inv++;
if(rdc[inv]<0) subsub_solve = 0;
}
if(subsub_solve){
/*store the solutions*/
for(i=0; i<data_m2; i++){
sspi[prob][i] = 0;
}
for(i=0; i<data_n2;i++){
if(basis[i]<data_m2) sspi[prob][basis[i]]=rhs[i];
ssdual[prob][i] = -rdc[data_m2+i];
}
total_subsub -= rdc[data_m2+data_n2];
}
else{
/*find the positive component: let basis outv out of the basis*/
int outv = -1;
double temp1 = 0;
temp = my_infi;
for(i=0; i<data_n2; i++){
if(lhs[i][inv]>0){
temp1 = rhs[i]/(double)lhs[i][inv];
if(temp1<temp){
temp = temp1;
outv = i;
}
```

```
}
}
if(outv<0){
subsub_solve = 1;
//output_file << "Subsub problem " << prob  << " Unbounded!" << endl;
for(i=0; i<data_m2; i++){
sspi[prob][i] = my_infi;
}
for(i=0; i<data_n2;i++){
ssdual[prob][i] = my_infi;
}
total_subsub -= my_infi;
}/*if(outv<0)*/
else{
/*pivot*/
//output_file << "Subsub out index: " << outv << endl;
//output_file << "Subsub out basis: " << basis[outv] << endl;
//output_file << "Subsub in basis: " << inv << endl;
//output_file << "Subsub reduced cost: " << endl;
basis[outv] = inv;
temp = lhs[outv][inv];
for(j=0; j<data_n2+data_m2; j++){
lhs[outv][j] /= temp;
}
rhs[outv] /= temp;
for(i=0; i<data_n2; i++){
if(i != outv){
temp = lhs[i][inv];
for(j=0; j<data_n2+data_m2; j++){
lhs[i][j] -= temp*lhs[outv][j];
}
rhs[i] -= temp*rhs[outv];
}

}
temp = rdc[inv];
//output_file << "multiplier for rdc: " << temp <<endl;
```

```
for(j=0; j<data_n2+data_m2; j++){
rdc[j] -= (double)(temp*lhs[outv][j]);
}
rdc[data_n2+data_m2] -= (double)(temp*rhs[outv]);
//for(j=0; j<data_n2+data_m2+1; j++){
//output_file << rdc[j] << endl;
//}

//output_file << "Subsub lhs: " << endl;
//for(i=0;i<data_n2;i++){
//for(j=0; j<data_n2+data_m2;j++)
//output_file << lhs[i][j] << endl;
//output_file << "  ... end row." << endl;
//}
//output_file << "Subsub rhs: " << endl;
//for(i=0;i<data_n2;i++){
//output_file << rhs[i] << endl;
//}
}/*else for if(outv<0)*/

}/*else for if(subsub_solve)*/

}/*while(!subsub_solve)*/
prob ++;
}/*while(prob <data_K)*/

if(sub_LB != 0){
//if(abs((double)(total_subsub - sub_LB)/(double)sub_LB) < my_tol){
if(abs((double)(total_subsub - sub_LB)) < my_tol){
output_file << "SUB problem LB and SUBSUB problem obj is within
_tolerence level. Stop." << endl;
sub_opt = 1;
}
}
if(total_subsub > current_DELTA && !sub_opt){
/*if DELTA< SUBSUBOBJ, add optimality cut for sub problem,
continue this while (current_DELTA <= Infinity) loop */
```

```
sub_cut_status = 1;
if(sub_cut_num>0) sub_cut_status = 2;
//output_file << "SUB Cut parameters: " << endl;
for(i=0;i<data_m3;i++){
subCutG[sub_cut_num][i] = 0;
for(k=0; k<data_K; k++){
for(j=0; j<data_n2; j++){
subCutG[sub_cut_num][i] -= data_B[k][i][j]*ssdual[k][j];
}
}
//output_file << subCutG[sub_cut_num][i] << endl;
if((sub_cut_status == 2) &&
_(abs(subCutG[sub_cut_num][i]-subCutG[sub_cut_num-1][i])>my_tol)){
sub_cut_status = 1;
}
}
subCutg[sub_cut_num] = 0;
for(i=0;i<data_n2;i++){
for(k =0;k<data_K;k++){
subCutg[sub_cut_num] += data_p[k]*data_q[k][i]*ssdual[k][i];
}
}
//output_file << subCutg[sub_cut_num] << endl;
if((sub_cut_status == 2)
&& (abs(subCutg[sub_cut_num] - subCutg[sub_cut_num-1])>my_tol)){
sub_cut_status = 1;
}
if(sub_cut_status == 2){
output_file << "Duplicated cuts for SUB problem." << endl;
break;
}
sub_cut_num ++;

}
else{
/*if DELTA>=SUBOBJ, sub optimal, add optimality cut for the master problem*/
sub_opt = 1;
```

```
}
}/*while(!sub_opt && !stop)*/

submodel.end();
subcplex.end();
if(stop) {break;}
if(current_THETA >= -sub_obj_value){
/*master problem optimality check in step 3.3*/

master_opt = 1;
output_file
<< "THETA is not less than second stage objective value, stop. SUB OBJECTIVE:  "
<< -sub_obj_value  << " THETA: " << current_THETA << endl;
break;
}
if(master_LB != 0){
//if(abs((double)(-sub_obj_value - master_LB)/(double)master_LB) < my_tol){
if(abs((double)(-sub_obj_value - master_LB)) < my_tol){
master_opt = 1;
output_file
<< "MASTER problem LB and SUB problem obj is within tolerence level. Stop."
<< endl;
break;
}
}
/*if not optimal, add cuts to master problem.*/
//output_file << "MASTER Cut parameters: " << endl;
master_cut_status = 1;
if(master_cut_num>0) master_cut_status = 2;
for(i=0; i<data_n1; i++){
masterCutE[master_cut_num][i] = 0;
for(k=0; k<data_K; k++){
for(j=0; j<data_m2; j++){
masterCutE[master_cut_num][i] -= data_T[k][j][i]*sspi[k][j];
}
}
if((master_cut_status == 2) &&
```

```
 (abs(masterCutE[master_cut_num][i] - masterCutE[master_cut_num-1][i])>my_tol)){
master_cut_status = 1;
}
//output_file << masterCutE[master_cut_num][i]  << endl;
}
masterCute[master_cut_num] = 0;
for(k=0; k<data_K; k++){
for(j=0; j<data_m2; j++){
masterCute[master_cut_num] -= sspi[k][j]*data_h[k][j];
}
}
for(i=0; i<data_m3; i++){
masterCute[master_cut_num] -= sSIGMA[i]*data_BG[i];
}
if((master_cut_status == 2) &&
(abs(masterCute[master_cut_num] - masterCute[master_cut_num-1])>my_tol)){
master_cut_status = 1;
}
//output_file << masterCute[master_cut_num] << endl;
if(master_cut_status == 2){
output_file << "Duplicated cuts for MASTER problem." << endl;
break;
}
master_cut_num ++;


}/*while(!master_opt && !stop)*/

finish = clock ();
cputime = finish - start;

output_file << "MASTER Objective value = " << master_obj_value << endl;
output_file << "x = " << sx[0] << endl;
output_file << "CPU Time = " << cputime << endl;
output_file << "MASTER Iterations = " << master_iter << endl;
output_file << "SUB Iterations = " << sub_iter << endl;
```

```
env.end();

for(k=0; k<data_K; k++){
free(sspi[k]);
free(ssdual[k]);
}
for(i=0; i<data_n2; i++){
free(lhs[i]);
}
free(sspi);
free(ssdual);
free(lhs);
free(rhs);
free(rdc);
free(sscost);
free(Tx);
free(basis);

for(i=0; i<max_cut; i++){
free(subCutG[i]);
}
free(subCutG);
free(subCutg);
free(sSIGMA);

for(i=0; i<max_cut; i++){
free(masterCutE[i]);
}
free(masterCutE);
free(masterCute);
free(sx);

free(data_c);
free(data_p);
free(data_b);
free(data_BG);
for(i=0; i< data_m1; i++){
```

```
free(data_A[i]);
}
for(k=0; k< data_K; k++){
for(i=0; i<data_m3; i++){
free(data_B[k][i]);
}
for(i=0; i<data_m2; i++){
free(data_T[k][i]);
free(data_W[k][i] );
}
free(data_q[k]);
free(data_B[k]);
free(data_T[k]);
free(data_W[k]);
free(data_h[k]);

}
free(data_A);
free(data_q);
free(data_B);
free(data_T);
free(data_W);
free(data_h);

for(i=0; i<data_m2; i++){
free(T_mean[i]);
free(W_mean[i]);
}
for(i=0; i<data_m3; i++){
free(B_mean[i]);
}
free(h_mean);
free(T_mean);
free(W_mean);
free(B_mean);
free(pq_mean);
}
```

```
output_file.close();
return 0;
}
```