View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by University of Tennessee, Knoxville: Trace

University of Tennessee, Knoxville
**Trace: Tennessee Research and Creative Exchange**

Doctoral Dissertations | Graduate School

8-2014

# Optimization of Healthcare Delivery System under Uncertainty: Schedule Elective Surgery in an Ambulatory Surgical Center and Schedule Appointment in an Outpatient Clinic

Zhaoxia Zhao
*University of Tennessee - Knoxville*, zhaoxia.zhao@gmail.com

## Recommended Citation

Zhao, Zhaoxia, "Optimization of Healthcare Delivery System under Uncertainty: Schedule Elective Surgery in an Ambulatory Surgical Center and Schedule Appointment in an Outpatient Clinic. " PhD diss., University of Tennessee, 2014.
https://trace.tennessee.edu/utk_graddiss/2879

This Dissertation is brought to you for free and open access by the Graduate School at Trace: Tennessee Research and Creative Exchange. It has been accepted for inclusion in Doctoral Dissertations by an authorized administrator of Trace: Tennessee Research and Creative Exchange. For more information, please contact trace@utk.edu.

To the Graduate Council:

I am submitting herewith a dissertation written by Zhaoxia Zhao entitled "Optimization of Healthcare Delivery System under Uncertainty: Schedule Elective Surgery in an Ambulatory Surgical Center and Schedule Appointment in an Outpatient Clinic." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Industrial Engineering.

Xueping Li, Xiaoyan Zhu, Major Professor

We have read this dissertation and recommend its acceptance:

James Ostrowski, Russell Zaretzki

Accepted for the Council:
Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

# Optimization of Healthcare Delivery System under Uncertainty: Schedule Elective Surgery in an Ambulatory Surgical Center and Schedule Appointment in an Outpatient Clinic

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Zhaoxia Zhao

August 2014

Copyright © 2014 by Zhaoxia Zhao
All Rights Reserved

*This dissertation is dedicated to my parents, Peifeng Liu and Pingxue Zhao, and my husband, Hongbiao Yang, for their love, support and encouragement.*

# Acknowledgements

First and foremost, I am deeply indebted to my two advisors Dr. Xueping Li and Dr. Xiaoyan Zhu. Their willingness to support my work and the guidance throughout my studies have allowed me to develop my skills as a researcher within a supportive team environment. I thank them for the precious opportunity. To my dissertation committee members, Dr. James Ostrowski and Dr. Russell Zaretzki, Thank you for generously offered your time and provided your insights on this work. Your inputs in the dissertation are highly appreciated.

I also would like to thank my fellow graduate students: Yu Huang, Cong Guo, Jiao Wang, Rui Xu, Qi Yuan, and Shima Mohebbi. Thank them for sharing the graduate school life together here and the supportive team environment. In addition, I would like to thank my friends at UT, Lanying Ma, Yi Zhao, Xie Xie, Caiqiao Xu, Shugang Ji, for listening, offering me advice, and always being there for me.

# Abstract

This work investigates two types of scheduling problems in the healthcare industry. One is the elective surgery scheduling problem in an ambulatory center, and the other is the appointment scheduling problem in an outpatient clinic.

The ambulatory surgical center is usually equipped with an intake area, several operating rooms (ORs), and a recovery area. The set of surgeries to be scheduled are known in advance. Besides the surgery itself, the sequence-dependent setup time and the surgery recovery are also considered when making the scheduling decision. The scheduling decisions depend on the availability of the ORs, surgeons, and the recovery beds. The objective is to minimize the total cost by making decision in three aspects, number of ORs to open, surgery assignment to ORs, and surgery sequence in each OR. The problem is solved in two steps. In the first step, we propose a constraint programming model and a mixed integer programming model to solve a deterministic version of the problem. In the second step, we consider the variability of the surgery and recovery durations when making scheduling decisions and build a two stage stochastic programming model and solve it by an L-shaped algorithm.

The stochastic nature of the outpatient clinic appointment scheduling system, caused by demands, patient arrivals, and service duration, makes it difficult to develop an optimal schedule policy. Once an appointment request is received, decision makers determine whether to accept the appointment and put it into a slot or reject it. Patients may cancel their scheduled appointment or simply not show up. The no-show and cancellation probability of the patients are modeled as the functions of the indirect waiting time of the patients. The performance measure is to maximize the expected net rewards, i.e., the revenue of seeing patients minus the cost of patients' indirect and direct waiting as well as the physician's overtime. We build a Markov Decision Process model and proposed a backward induction algorithm to obtain the optimal policy. The optimal policy is tested on random instances and compared with other heuristic policies. The backward induction algorithm and the heuristic methods are programmed in Matlab.

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 Overview of Optimization Under Uncertainty

Optimization by definition stands for making a single choice from a range of feasible ones to achieve the best results. The best choices that meet the objective the most are also called optimal solutions. Depends on the data we have, different techniques are applied to identify the optimal or near-optimal solutions. In literature, there are a number of well-established techniques and algorithms that have been proposed to solve the problems where all the data we need to make decisions are available and deterministic. To name a few, linear programming, nonlinear programming, mixed integer programming, constraint programming, and branch and bound algorithm are proposed to find the exact optimal solutions; while heuristic and meta-heuristic algorithms, such as genetic algorithm, tabu search, ant colony optimization, particle swarm optimization, and local search are proposed to find near optimal solutions.

However, in reality, due to variability of process and environmental data as well as unknown influential factors, uncertainty is prevalent in inventory control, supply chain, production planning, scheduling, location, transportation, and so forth. For instance, the demand for a certain product or service, the time needed to perform a surgery, the transportation time from station A to station B, the availability of machines in a production line, and so on. The deterministic optimization techniques mentioned above are not suitable to be applied here. In order to make reliable and efficient, hopefully optimal, decisions in the presence of the uncertainties, a number of research has been done on developing techniques for optimization under uncertainty.

There are two main branches to solve the problems under uncertainty. One is simulation, which is a very powerful tool to model the uncertainties in activities, perform the "What-If" analysis, and identify the bottleneck of the process. When the system is too complex to model or the system's behavior cannot be represented by mathematical equations, simulation models become the only tool we can resort to do the analysis. However, the simulation models cannot provide closed form solutions. In addition, the simulation models are computationally intensive and may require considerable computation time (Erdogan and Denton, 2011).

The other branch of studies is stochastic programming. Generally speaking, the probability distribution of random parameters is known or can be estimated. In most of the studies of stochastic programming models, the objective is to find the optimal decision that minimizes the expectation of some functions, and the decisions are required to be feasible for all (or almost all) possible data instance.

## 1.2 Motivation of the Study

In many postindustrial societies, health care industry is one of the largest and fast-growing industries in the service sector. According to statistics (Keehan et al., 2008), health care expenditures have been increased rapidly for the last few decades, and will reach 19.5% of the US GDP by 2017.

On the one hand, operating rooms (ORs) are usually the most critical resources in the hospital. In some hospital, more than 40% of the total revenue comes from ORs, and the ORs consist of a large proportion of the total expenses (Denton et al., 2010). Research indicates that in some situations, the ORs have not reached the target utilization, and better surgery-to-OR scheduling strategies are desired to improve the utilization level and thus achieve cost saving (Erdogan and Denton, 2011). On the other hand, the increasing expense of healthcare is due to more expensive treatment technologies and unfavorable population demographic trends. To deal with the increasing expense, many hospitals shorten the patient's length of stay and shift care from inpatient to outpatient, which in turn increases the pressure of outpatient clinic on improving healthcare access.

One potential solution is to develop better scheduling policies for ORs and outpatient clinics to improve the utilization level and thus achieve significant cost savings. A good scheduling enables the health care provider matching the available capacity and patient demands well. However, both the elective surgery scheduling and clinic appointment scheduling involve various resources, and the accessibility of these resources together with the condition of the patient lead to the uncertainty of patient arrival, service duration, etc. Taking all these factors into consideration, both elective surgery scheduling and clinic appointment scheduling are very complex problems.

In general, operations research is a really powerful tool, which employs various problem-solving techniques, such as mathematical model, simulation, queuing theory, and stochastic process model, for obtaining optimal or near-optimal solutions and providing insights for complex decision problems in many fields. As mentioned in Section 1.1, there are a number of research on developing efficient methods for solving problems under uncertainty. The efficiency of the healthcare system would get improved a great deal by applying these methods on solving the problems in the healthcare delivery system. Besides, along with the growth of the information technology used in the health care industry, more and more historical data on patient flow are available for research. Therefore, the collaboration between the health care community and operations research community will be expanded in the future (Gul, 2010).

## 1.3 Contribution and Document Organization

Two types of scheduling problems in the healthcare delivery system are investigated in this study. One is about scheduling elective surgeries to multiple ORs in an ambulatory surgical center with the objective of minimizing the (expected) total cost of the surgical center. The other problem is about scheduling appointments at an outpatient clinic with the objective of maximizing the long-run net reward of the clinic. The background, definition, as well as the healthcare environment of the two problems are unrelated. And the methods applied to solve these two problems are also different. Therefore, we address these two problems separately in two chapters.

Chapter 2 studied an elective scheduling problem in an ambulatory surgical center. In this problem, the Operating Room (OR) manager of an ambulatory surgical center is going to make decisions about scheduling a known set of surgeries to multiple ORs subject to the availabilities of the surgeons, nurses, anesthetist, ORs, recovery beds, as well as equipment and other resources. The scheduling decisions are needed to make include (1) number of ORs to open, (2) set of surgeries to assign to each OR, and (3) the surgery sequence within each opened OR. Besides the availability constraints, the type of surgeries and constraints of the surgery type of each OR due to the required equipment availability are also investigated. Three activities, including the setup before the surgery, the surgical operation itself, and the recovery after the surgery, are studied in this work. Although some existing studies on surgery scheduling problems considered some of the factors that we include in our model, none of them modeled them all. In other words, we study a new problem different from former work. The contributions of this work are: 1) we are the first few who model the sequence-dependent setup time in an analytical way, 2) we are the first few who consider the decisions of the three aspects simultaneously, 3) we are the first few who apply Constraint Programming (CP) methods on solving elective surgery scheduling problems, 4) we study the influence of the activity duration uncertainties on the scheduling decision by sensitivity analysis as well as modeling it explicitly in a two-stage stochastic programming model.

Chapter 3 investigated a problem of scheduling the appointments for one physician in an outpatient clinic. The demand request for making an appointment is unknown and arrives randomly. In addition, the service time of each patient is also a random parameter due to different physical conditions of patients. The uncertainties in both the demand arrivals as well as service times are considered in this study. This problem is an on-line optimization problem. When the decision maker receives a call for making an appointment of a certain day, he/she should make decisions whether to accept this appointment, if yes, which time slot to put this request on the schedule of this physician. The patients usually cannot get scheduled on the day they call. The time between the day they call for making the appointment and the day they are scheduled to get the service is called indirect waiting time of the patient. During their indirect waiting time, the patient may may cancel their appointment. Besides, on the day they are scheduled to, the patient many simply does not show up. The scheduling decisions should consider all these factors to maximize the long-run expected net rewards, i.e., the revenue of seeing patients minus the cost of patients' indirect and direct waiting as well as physician's overtime. The contributions of this work are: 1) we are the first few to consider the indirect waiting time of the patients, 2) we study the undesirable patient behavior to the appointment scheduling system, including walk-ins, no-shows, and cancellations, 3) we are among the first few who model the probabilities of no-show and cancellation as functions of the indirect waiting time, 4) we build a Markov Decision Process (MDP) model, develop an efficient algorithm to solve the model, and propose a good and easy-implemented policy for the schedule manager.

Chapter 4 concludes this work by summarized the finding of the two scheduling problems in the healthcare delivery system and pointing out the future research directions.

# Chapter 2

# Elective Surgery Scheduling Problem in an Ambulatory Surgical Center

## 2.1 Introduction

In general, surgeries could be divided into two categories, elective surgeries and urgent surgeries. In contrast to urgent surgeries, the elective surgeries do not require medical emergency. Therefore, elective surgeries can be scheduled in advance and most of them can be performed safely in an outpatient setting. And more complex surgeries that require emergency services and inpatient services are performed at hospitals. This study focuses on the elective surgeries which could be performed on an outpatient basis.

In the past few decades, there was a steady increase in the percentage of the outpatient procedures. According to a recent report (Papel, 2009), about 65% of the elective surgeries are done in outpatient settings comparing less than 10% in 1979. This increasing trend is partly due to the improvement of the technologies, including anesthesia as well as surgical. Specifically, the improved anesthesia techniques significantly reduce the postoperative nausea and vomiting, which allows the patients to take less time in the recovery area and go home early for convalescence; the improved surgical techniques lead to a decrease in the intra-operative time (Papel, 2009).

Along with the increasing demand of elective surgeries performed on an outpatient basis, a new surgery delivery system, ambulatory surgical centers, has emerged. The ambulatory surgical centers are designed to perform elective surgeries with minimal supporting resources and not prepared for emergencies or overnight stays (Denton et al., 2007). Despite the differences between the hospitals and ambulatory surgical centers, many characteristics of the OR environment are the same, for example, the fixed costs of the ORs, a large proportion of which is associated with the labor cost of the OR teams, in both environments are very high (Erdogan and Denton, 2011).

An ambulatory surgical center is usually opened 8 to 10 hours a day, and is equipped with an intake area, several ORs, and a recovery area. Figure 2.1 illustrates the patient flow through an ambulatory surgical center. When a patient arrives, he/she is allocated to the intake area to do some preparation work for the surgery. Then the patient is subsequently taken to surgery in an unoccupied OR, and after the surgery, the patient is sent to the recovery area for some post anesthesia care. And if there is no space available in the recovery area, the patient has to stay in the OR. In other words, the recovery area could also be a bottleneck for the ambulatory surgical center at certain times of the day. Usually, after staying a period of time in the recovery area, the patient leaves the surgical center and recovers at home. Since the ambulatory surgery centers are not prepared for overnight stays, all the patients need to leave the centers at the end of the day.

A surgery can last from minutes to hours, depending on the type of the procedures. Different types of surgeries may require different equipment and resources. Some mobile equipment and resources can be shared between several ORs, while some are dedicated to a particular OR. Therefore, the type of surgeries can be performed in each OR is constrained by the equipment and resources associated with that OR. When a surgery is finished, some setup work, for example, cleaning up the OR, changing equipment, refilling sterilization resources, and getting proper staff (surgeons, nurses, and anesthesiologists), has to be done before the start of the next surgery. The setup times depend on the types of two successive surgeries, and thus are sequence-dependent.

Figure 2.2 illustrates a schedule of elective surgeries in an OR and it is assumed that this schedule meets the constraints on the availability of the surgeons. In this example, there is only one recovery bed and the recovery time of surgery 1 is much longer than the surgery 2's duration. In this figure, the total length of the block represents the predefined maximum operating time, which is

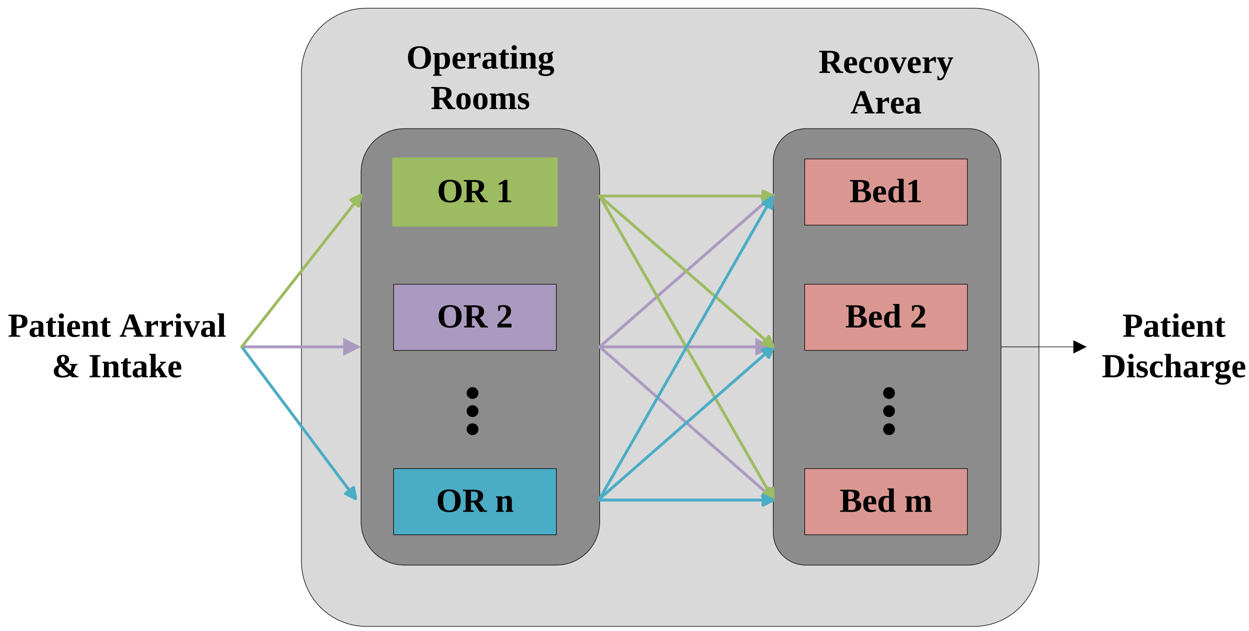


**Figure 2.1:** The patient flow through an ambulatory surgical center.

allowed by the labor legislation. The blocks shaded with the"\\" pattern represent the (expected) surgery durations, and the blocks shaded with the "||" pattern are the sequence-dependent setup times. When surgery 2 is finished, since there is no available bed in the recovery area, the patient has to stay a period of time in the OR where the recovery procedure begins. Later, the patient is transferred to the recovery area when a bed is discharged in the recovery area. The period of time that this patient stays in the OR is represented by the block shaded with the "++" pattern. Depending on the length of time the OR is planned to be available, some overtime may occur at the end of the day as the block shaded with the "××" pattern in Figure 2.2. Besides the overtime cost, the ORs usually have enormous fixed costs and various expensive resources, including surgeons, nurses, anesthesiologists, disposable supplies, medications, and different equipment, are involved (Erdogan and Denton, 2011).

The surgery durations and recovery times depend on lots of factors, including the type of the surgeries, the physician/nurse skills, and the patient physical conditions. Without considering these uncertainties may result in infeasible schedules, for example exceeding the maximum operating time, etc. Even for scheduling plan that are feasible, ignoring the uncertainties may lead to higher

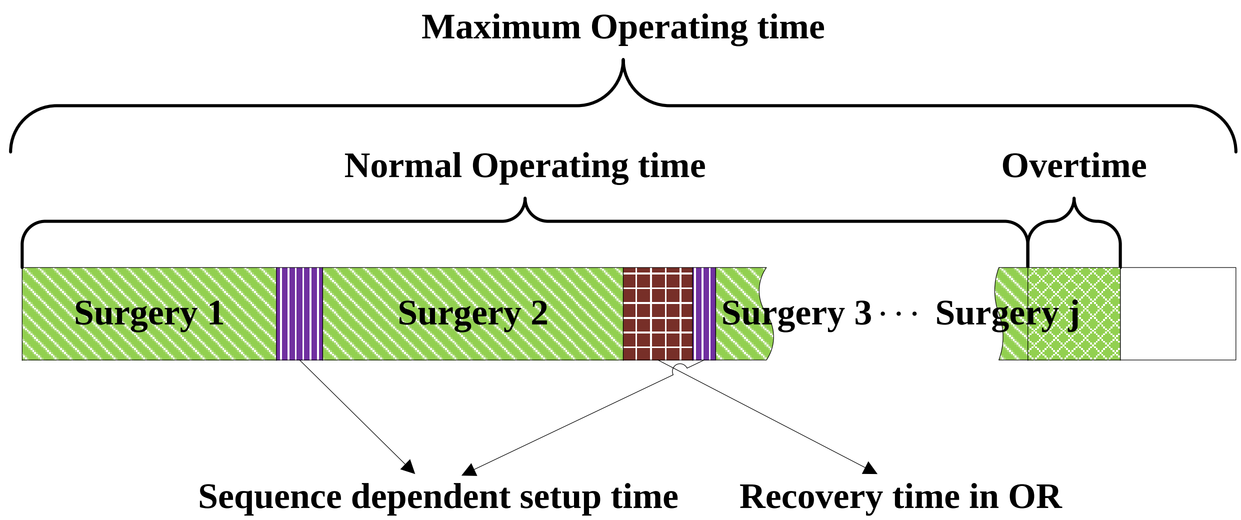


**Figure 2.2:** An example schedule of elective surgeries in an OR.

expected cost. Scheduling the elective surgeries in a single OR with uncertain surgery duration is already not easy (Sun and Li, 2011), scheduling the elective surgeries to multiple ORs when considering the uncertainty of the surgery and recovery duration as well as the availability of the ORs, surgeons, and recovery beds is a very hard problem. In addition, there are a number of competing performance criteria, such as the utilization of the ORs, surgeon's overtime and idle time, and the patient's waiting time, which bring more challenges for surgery scheduling.

This work studies problems of scheduling a set of known elective surgeries to an ambulatory surgical center, including multiple ORs and recovery beds, subject to the resource availabilities. Different from the previous problem settings, we model the setup times between two surgeries as sequence dependent. Three aspects of the scheduling decisions are needed to make, including 1) number of ORs to open, 2) set of surgeries to assign to each opened OR, and 3) the surgery sequence in each OR. Unlike previous researches, instead of solving this scheduling problem in multiple stages, we take these three aspects of decisions into account simultaneously. As stated in Chapter 1, we build both MIP and CP to solve this problem. In addition, we investigate the influence of the variations in surgery durations as well as the recovery time on the scheduling decisions.

The remainder of this chapter is organized as follows. Section 2.2 summarized the related research on surgery scheduling problems. Section 2.3 defines the problems and gives a brief introduction to the methodologies used to solve the problems. Section 2.4 presents the deterministic MIP and CP models, and their performance, computational time as well as the solution quality, are evaluated on random instances. Section 2.5 proposed a two stage stochastic constraint programming model to explicitly investigate the influence of the variations in activities, and a significant gain of the stochastic solutions have been proved. Section 2.6 concludes this chapter and summarizes the findings.

## 2.2 Literature Review

Surgery scheduling problems have been widely investigated in both medical and operations research fields. Several comprehensive reviews have been done on this research topic. Magerlein and Martin (1978) categorized previous literatures on hospital surgical suites by the stage of decision making: to determine the surgery date in advance and to determine the sequence of the patients on a given day. Cardoen et al. (2010) used several descriptive fields to structure the review, and these fields include patient characteristics, performance measures, decision delineation, methodology, uncertainty, and application of research. Within each field, the key concepts are elaborated and the important trends are pointed out. Erdogan and Denton (2011) provides a systematic review of the operations research methodologies used to solve the surgery scheduling problems. They divided these methods into four categories, including queuing models, simulation, optimization, and heuristic methods. Guerriero and Guido (2011) reviewed the operations research applications on three different decision levels in the surgical planning and scheduling processes, i.e., strategic, tactical, and operational.

In this review, we focus on the studies related to our problems. Specifically, we will first give an introduction on two popular surgery scheduling strategy. And then we will talk about the complicating factors in different surgery scheduling problem settings that have been exploited in previous work, including OR environment, number of ORs, resource availability, evaluation criteria, as well as setup times.

### 2.2.1 Surgery Scheduling Systems

There are two-well known surgery scheduling systems, the block-scheduling system and open-scheduling system. (Gupta and Denton, 2008). In the block-scheduling system, specific surgeons or surgical groups are assigned time periods of a particular OR and the surgeons can book cases into their allocated time periods. The allocated block-times assigned to surgeons are estimated based on the historical data. One disadvantage of the block scheduling is that there is no coordination between different surgeons. For a certain day, it is highly possible that the time period of one surgeon is under-utilized while the time period of another surgeon is over-utilized. Currently the block scheduling system is widely used in surgical suits, and many researches have been done on this topic to improve the policy. For example, Marcon and Dexter (2007) used simulation methods to do an observational study on the impact of surgery sequence on the staffing level of OR holding area and recovery area in a block-scheduling system. They mentioned that to reduce the patients' average waiting from scheduled start time, without violating other constraints, the surgeries should be sequenced based on the accuracy of the predication of their durations. Blake et al. (2002) designed a weekly schedule, under the block scheduling system, to minimize the difference between the actual allocated OR time and the targets.

Under the open-scheduling system, the requests for OR time are submitted by surgeons to the OR manager who will create the OR schedule prior to the day of surgery. In general, the requests are accepted based on the order of their arrival time, the available OR time, surgery's priorities, etc. However, the open-scheduling system also has a major disadvantage that the uncertain demand of the surgeries causes variable daily OR utilization rate. Fei et al. (2010) solved a weekly surgery scheduling problem in a surgical suite where an open scheduling strategy is used. They solve the problem in two steps, i.e., the surgery date is determined in the first step and daily decisions upon surgery sequence are made in the second step. Sufahani et al. (2011) conducted a similar study as Blake et al. (2002) to find a weekly schedule that minimizes the deviation of OR times allocated to different departments from their targets with open scheduling strategy.

The assumption that the set of surgeries to be scheduled are known in advance is reasonable for the open-scheduling system. Therefore, our model is applicable to the open-scheduling system described above, i.e., creating the OR schedule prior to the day of surgery. On the day of the surgery, the well-established schedule might not be implemented as expected due to lots of factors, for example, the surgery durations are longer or shorter than the estimated ones, the required equipment or resources are temporally unavailable at the time of the surgery, arrival of emergency cases or other add-on cases, etc. The OR managers need to make minor changes to the established schedule (Erdogan and Denton, 2011).

### 2.2.2 Complicating Factors

#### OR Environment

The hospital environment is more complex than the ambulatory surgical center in that the hospitals can accommodate inpatient surgeries, outpatient surgeries and emergent cases. The mixture of patient types brings two major differences for scheduling surgeries in hospitals than in ambulatory surgical centers. The first one is about the demand of emergency cases, which has been investigated in a number of previous researches. For example, Gerchak et al. (1996) addressed the reservation planning problem for elective surgeries when the ORs are shared by both elective and emergency patients. Lamiri et al. (2008a) and Lamiri et al. (2008b) built stochastic models to allocate a set of elective surgeries with uncertain demand for emergency surgeries.

The other difference of surgery scheduling in hospitals lies in the planning horizon. Scheduling surgeries in hospitals not only need to consider the constraints on the capacity of ORs but also the availability of hospital beds for patients' recovery periods, hospitalization date, patient deadlines, etc. Therefore most of the studies on scheduling surgeries in hospital settings are long planning periods. For example, Guinet and Chaabane (2003) investigated an inpatient surgery scheduling problem in an operating theater over a medium planning horizon with the objective of minimizing the fixed patient intervention costs. Testi et al. (2007) presented a three-phase, hierarchical approach for the weekly scheduling of ORs. In the first phase, the number of sessions to be weekly scheduled for each ward is determined, phase two solves the surgery-to-OR assignment, and the last phase works on the patients sequence in each session. Fei et al. (2008) worked on assigning a set of surgical cases to multiple ORs over one-week period with restrictions on ORs' capacity and patients' deadline and assumptions that all resources are available. The objective of their study is to minimize the total idle costs and overtime cost of ORs over the planning horizon. Pham and Klinkert (2008) analyzed a surgery scheduling problem over a long planning period for both elective and add-on cases of inpatients and outpatients. Lamiri et al. (2008a,b) addressed problems of scheduling several elective surgery cases into a one-week planning horizon with the objective of minimizing the patient-related costs and expected OR's utilization. As stated above, the ambulatory surgical centers are built for handling the elective surgeries. And since the patients leave on the same day of their surgeries and the ambulatory surgical centers did not accommodate the overnight stay of the patients, the scheduling decisions in this OR environment are mainly daily. Different from long planning horizon scheduling problems which focus on to which OR on which day, the daily surgery scheduling problems focus on the sequencing and start time of the surgeries.

**Number of ORs**

Depending on the number of ORs considered, previous studies on surgery scheduling problems can be divided into two branches. The single OR scheduling problems aim to determine the start times for a set of surgeries in an OR on a given day. Usually, the uncertainty of the surgery durations is considered in single OR scheduling problems, and the waiting time of patients, idle time and overtime of physicians are included. For previous studies on single OR scheduling problems, the readers are referred to (Wang, 1993; Denton et al., 2007; Sun and Li, 2011). Our study belongs to another branch, multiple ORs scheduling problems, and lots of work has been done on this topic. Batun (2011) proposed a two-stage stochastic MINLP model to examine the benefit of processing surgeries paralleled in multiple ORs. They considered the uncertainty in surgery durations, and the decisions of their model included the allocation and start time of each surgery. The objective is to minimize the cost of the ORs and the idle cost of the surgeons. Jebali et al. (2006) used a two-stage approach to solve a surgery scheduling problem within a surgical center with multiple ORs. The first step is to determine the surgeries assignment to the ORs, and the second step is to deal with the surgeries sequencing. In this paper, they considered the availability of different resources, surgeons, equipment, intensive care unit, and due date of the surgeries.

### Resource Availability

As stated above, the schedule planning process involves identifying and coordinating of all necessary resources, including surgical staff, anesthesiologist, nursing, rooms, equipment, supplies, instruments, support staff, interdepartmental prepared patient. Finding feasible schedules to satisfy all these constraints is not easy in some cases. Some of the previous studies made assumptions that the resourced needed during the surgeries are always available, while a portion of previous researches investigated the impact of availability of peripheral resources on the decisions of surgery schedules. Schmitz and Kwak (1972) used simulation to analyze a multi-OR surgical suite with recovery rooms to determine the number of ORs to open on a day given that the number of surgeries to be performed is known. They also studied the impact of an increase in the number of ORs on the recovery room usage. Hsu et al. (2003) built a deterministic no-wait, two-stage process shop scheduling model and developed a tabu search based heuristic algorithm to find the minimal number of nurses in the recovery area of an ambulatory surgical center. They pointed out that the surgery scheduling decision in the ORs determines the peak number of patients during the day, which further determines the number of nurses needed. Marcon and Dexter (2006) built simulation models to test the impact of seven different surgery sequencing rules on the staffing level of the recovery area. Guinet and Chaabane (2003), Kharraja et al. (2002), and Jebali et al. (2006) also took into account the constraints on the recovery room capacity when constructing the OR schedules.

### Evaluation Criteria

Different stakeholders in the healthcare, including physicians, anesthesiologists, nurse, administrators, patients, etc., view the OR scheduling very differently (Blake et al., 1997). Therefore, the "optimal" design of an OR scheduling on a particular day is significantly different for each stakeholder. For the surgeons, the ideal scheduling is having their own OR, staff, equipment, and an anesthesiologist available whenever they want. The anesthesiologists want the surgeries to staggered start. For the nurses, the ideal scheduling is one team per room, all cases finishing in time, and having scheduled lunch and breaks. The ideal scheduling for patients is starting surgery upon arrival and no need to wait. This study stands on the administrators' point of view, and the criterion used to evaluate the proposed schedule is finishing all the surgeries at the end of the day by minimum fixed cost as well as the overtime cost.

**Setup Time**

Despite various surgery scheduling problem settings, the setup times between surgeries are overlooked in previous studies. The setup time is a very important factor in elective surgery scheduling, denoting the time interval between one patient leaving the OR to the next patient entering the OR. It represents the "no-value added" time that leads to loss of revenue for the surgical center. Usually the setup times depend on the types of two successive surgeries (Arnaout, 2010), and thus are sequence-dependent. It is common that when two successive surgeries belong to different types, major setups are needed and time/work consuming changeover operations are required; while if the two successive operations are of the same type, only minor setups are needed. Little work has been done on the sequence-dependent setup time in the ORs scheduling. To the best of our knowledge, only Arnaout and Kulbashian Arnaout (2010); Arnaout and Kulbashian (2008) have included the sequence-dependent setup times in their work. They built simulation models to test the existing heuristics, and concluded that the longest expected processing with setup time first rule is the most appropriate one to maximize the utilization of the ORs. In this study, we build mathematical programming models to include the sequence-dependent setup times exclusively, and try to solve the problem to optimality.

## 2.3 Problem Definition

Three aspects are considered when scheduling a known set of elective surgeries in an ambulatory surgical center: (1) number of ORs to open, (2) set of surgeries to assign to each opened OR, and (3) the surgery sequence within each opened OR. The performance measure of the scheduling decisions is the (expected) total cost, including the fixed costs and overtime costs of the ORs.

We investigate both deterministic and stochastic versions of the elective surgery scheduling problem. In some situation, the estimated surgery durations and recovery times are very close to the actual values. Therefore the deterministic model is sufficient enough for solving the problem. In contrast, when the variations of estimated the surgery durations and recovery times are high, the optimal scheduling decisions obtained on estimated durations may result in much higher cost and even infeasible schedule. Therefore, it is worthwhile to build a stochastic model to include the variability of the surgery and recovery durations.

This study assumes that the intake area has enough capacity to accommodate the patients, and thus focuses on the scheduling decisions within the multiple ORs and recovery area. In addition, it is assumed that the total number of available ORs and the types of surgeries that can be operated in each OR are known, and the total number of surgeries and their types are also known in advance.

This problem is investigated in two steps. In the first step, we formulate deterministic CP and MIP models under the assumption that surgery durations and recovery times are deterministic and known, i.e., the surgery durations and recovery times are estimated based on historical data and the estimations are close to the actual values. The two deterministic mathematical programming models are solved by commercial software IBM® ILOG®, and their performances (both computational time and solution quality) are evaluated on a set of well-designed numerical instances. In addition, we conduct some sensitivity analysis on the optimal scheduling decisions to the difference between the estimated and actual surgery and recovery durations. In the second step, the assumption of deterministic durations is relaxed, and a two stage stochastic MIP model is developed to model these uncertainties explicitly. We use an L-shaped algorithm to solve the stochastic programming model. By comparing the results of the stochastic model with the deterministic ones, the value of the stochastic solutions is proved. The two problems' definition and structure, including the objective, decision variables, assumptions, constraints, models, and solving techniques, are presented in Figure 2.3.

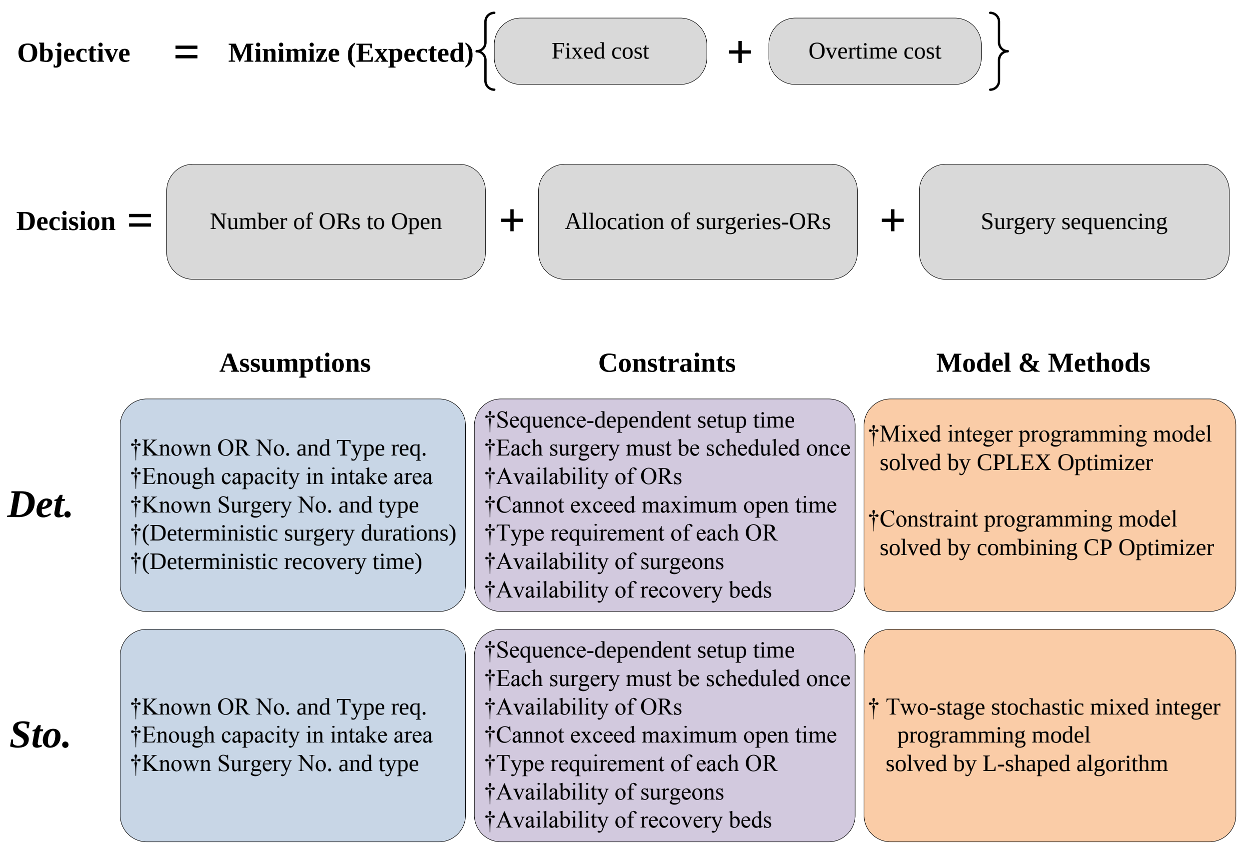


**Figure 2.3:** The definition and structure of the elective surgery scheduling problem.

## 2.4 Deterministic Mathematical Models

In this section, two deterministic models are presented, including an MIP model and a CP model, in Section 2.4.1 and Section 2.4.2. Both of them are modeled and solved by commercial solvers of IBM® ILOG®. The performance of the two models is compared in Section 2.4.3.

### 2.4.1 Deterministic Mixed Integer Programming Model

An MIP model is the minimization or maximization of a linear function subject to a set of linear constraints with some or all variables are integers. Usually speaking, linear programming problems can be solved in polynomial time while the MIP problems are NP-complete. One of the most widely used methods for solving MIP models is branch and bound. The basic idea is to divide the feasible region into several subsets. One can optimize over each subset separately, and the solution with the best objective value over all subsets is the global optimal solution. Many commercial software packages, for example, CPLEX and XPRESSMP, are developed based on LP based branch and bound, and they can solve the MIP models very efficiently. Besides branch and bound methods, other heuristic methods are also used to solve the MIP models, for example, genetic and evolutionary algorithms, simulated annealing algorithms, etc.

Many previous researches have built MIP models to solve surgery scheduling problems. Some of the models are solved by the commercial/open-sourced software package. Blake et al. (2002), Kharraja et al. (2006), Jebali et al. (2006), Persson and Persson (2007), Santibáñez et al. (2007), Zhang et al. (2008), Pham and Klinkert (2008), Persson and Persson (2009), and Cardoen et al. (2009) used CPLEX solver; Adan and Vissers (2002) and Vissers et al. (2005) used the solver MOMIP, which is developed based on branch-and bound algorithm and designed for solving middle-sized MIP problems. Chaabane et al. (2006) used the free GLPK solver to solve their developed linear programming model. Blake et al. (2002) called an add-in solver produced by Frontline Systems in Excel. Some of the models are solved by (customized) branch and bound algorithm. Fei et al. (2008) proposed a decomposition-based branch-and-bound algorithm and the linear relaxation of each node in the branch and bound algorithm is solved by a column generation procedure. van Oostrum et al. (2008) combined the column generation approach and CPLEX solver to solve their proposed two-phase decomposition model. Guinet and Chaabane (2003) proposed a primal-dual heuristic to solve the formulated MIP model.

Some previous research has built mixed integer nonlinear programming models which contain nonlinear objectives and/or constraints. Since these models are hard solved by the branch and bound based commercial software packages, some heuristic methods are proposed to solve them. Sier et al. (1997) and Beliën and Demeulemeester (2007) built a model with quadratic objectives and constraints, and used simulated annealing to find near optimal solutions. Roland et al. (2006) proposed a genetic algorithm to solve their model which contains nonlinear constraints.

In this study, we build a mixed integer linear programing model and solved it by IBM® ILOG® CPLEX solver. In the rest of this section, the notations for parameters and decision variables are defined first, followed by the model as well as the explanations of the objective functions and constraints.

- $i$: the index of the ORs, $i = 1, 2, \cdots, I$
- $k$: the index of the beds in the recovery area, $k = 1, 2, \cdots, K$
- $j$: the index of the patients, $j = 1, 2, \cdots, J$
- $l$: the index of the surgeons, $l = 1, 2, \cdots, L$
- $F_i$: the fixed cost of OR $i$
- $O_i$: the overtime cost of OR $i$ per unit time
- $H_i$: the normal operating time of OR $i$
- $G_i$: the maximum operating time of OR $i$
- $M$: a sufficiently large number
- $S_j$: the surgery duration of patient $j$
- $R_j$: the recovery time of patient $j$ after surgery
- $T_{jj'}$: the setup time of patient $j'$ when previous patient is $j$
- $\Phi_{ji}$: binary, equal to 1 if and only if surgery $j$ could be operated in OR $i$
- $P_l$: the moment that surgeon $l$ begins available
- $Q_l$: the moment that surgeon $l$ begins unavailable
- $\Theta_{jl}$: binary, equal to 1 if and only if patient $j$ is operated by surgeon $l$

- $y_i$: binary variable, equal to 1 if and only if OR $i$ is open
- $\alpha_{ji}$: binary variable, equal to 1 if and only if the surgery of patient $j$ is scheduled to OR $i$
- $\beta_{jj'}$: binary variable, equal to 1 if the start time of the surgery of patient $j$ is earlier than patient $j'$
- $\gamma_{jk}$: binary variable, equal to 1 if and only if patient $j$ is scheduled to bed $k$ in the recovery area
- $\delta_{jj'}$: binary variable, equal to 1 if the start time of patient $j$ in the recovery area is earlier than patient $j'$
- $a_j$: actual time that patient $j$ stays in an OR
- $b_{ji}$: start time of the surgery of patient $j$ in OR $i$, and equal to 0 if the surgery of patient $j$ is not operated in OR $i$
- $c_{ji}$: end time of the surgery of patient $j$ in OR $i$, and equal to 0 if the surgery of patient $j$ is not operated in OR $i$
- $d_{jk}$: start time of patient $j$ on bed $k$ in the recovery area, and equal to 0 if patient $j$ is not scheduled to bed $k$
- $e_{jk}$: end time of patient $j$ on bed $k$ in the recovery area, and equal to 0 if patient $j$ is not scheduled to bed $k$
- $x_i$: end time of OR $i$, and equal to 0 if OR $i$ is not open
- $z_i$: length of time that OR $i$ is over operated

The MIP model of the elective surgery scheduling problem is developed as:

$$\min W = \sum_{i=1}^{I} (F_i y_i + O_i z_i) \tag{2.1}$$

Subject to

$$\alpha_{ji} \leq \Phi_{ji} \quad \forall j, i \tag{2.2}$$

$$\sum_{i=1}^{I} \alpha_{ji} = 1 \quad \forall j \tag{2.3}$$

$$\sum_{j=1}^{J} \alpha_{ji} \leq |J| y_i \quad \forall i \tag{2.4}$$

$$\sum_{j=1}^{J} \alpha_{ji} \geq y_i \quad \forall i \tag{2.5}$$

$$\beta_{jj'} + \beta_{j'j} = 1 \quad \forall j, j'(j' \neq j) \tag{2.6}$$

$$\sum_{k=1}^{K} \gamma_{jk} = 1 \quad \forall j \tag{2.7}$$

$$\delta_{jj'} + \delta_{j'j} = 1 \quad \forall j, j'(j' \neq j) \tag{2.8}$$

$$a_j \geq S_j \quad \forall j \tag{2.9}$$

$$b_{ji} \leq G_i \alpha_{ji} \quad \forall j, i \tag{2.10}$$

$$c_{ji} \leq G_i \alpha_{ji} \quad \forall j, i \tag{2.11}$$

$$\sum_{i=1}^{I} c_{ji} = \sum_{i=1}^{I} b_{ji} + a_j \quad \forall j \tag{2.12}$$

$$x_i = \max_{\forall j} \{c_{ji}\} \quad \forall i \tag{2.13}$$

$$\sum_{i=1}^{I} b_{ji} - \sum_{i=1}^{I} b_{j'i} + M\beta_{jj'} \geq 1 \quad \forall j, j' \tag{2.14}$$

$$c_{ji} + T_{jj'} - b_{j'i} \leq (3 - \beta_{jj'} - \alpha_{ji} - \alpha_{j'i})M \quad \forall j, j'(j' \neq j), i \tag{2.15}$$

$$d_{jk} \leq M\gamma_{jk} \quad \forall j, k \tag{2.16}$$

$$e_{jk} \leq M\gamma_{jk} \quad \forall j, k \tag{2.17}$$

$$\sum_{k=1}^{K} e_{jk} = \sum_{k=1}^{K} d_{jk} + S_j + R_j - a_j \quad \forall j \tag{2.18}$$

$$\sum_{k=1}^{K} d_{jk} - \sum_{k=1}^{K} d_{j'k} + M\delta_{jj'} \geq 1 \quad \forall j, j' \tag{2.19}$$

$$e_{jk} - d_{j'k} \leq (3 - \delta_{jj'} - \gamma_{jk} - \gamma_{j'k})M \quad \forall j, j'(j' \neq j), k \tag{2.20}$$

$$\sum_{i=1}^{I} c_{ji} = \sum_{k=1}^{K} d_{jk} \quad \forall j \tag{2.21}$$

$$\sum_{l=1}^{L} \Theta_{jl} P_l \leq \sum_{i=1}^{I} b_{ji} \quad \forall j \tag{2.22}$$

$$\sum_{i=1}^{I} b_{ji} + S_j \leq Q_l \Theta_{jl} + (1 - \Theta_{jl})M \quad \forall j, l \tag{2.23}$$

$$\sum_{i=1}^{I} b_{ji} + S_j - \sum_{i=1}^{I} b_{j'i} \leq (3 - \beta_{jj'} - \Theta_{jl} - \Theta_{j'l})M \quad \forall j, j'(j' \neq j), l \tag{2.24}$$

$$x_i - H_i y_i \leq z_i \quad \forall i \tag{2.25}$$

$$x_i, z_i, a_j, b_{ji}, c_{ji}, d_{jk}, e_{jk} \geq 0 \quad \forall j, i, k \tag{2.26}$$

$$y_i, \alpha_{ji}, \beta_{jj'}, \gamma_{jk}, \delta_{jj'} \in \{0, 1\} \quad \forall j, j', i, k \tag{2.27}$$

The objective (2.1) is to minimize the fixed costs as well as the overtime costs of the ORs. Constraints (2.2) state that the surgeries scheduled to an OR must meet the type requirement of that OR. Constraints (2.3) indicate that each patient should be scheduled exactly once. Constraints (2.4) ensure that no surgery can be assigned to an unopened OR. Constraints (2.5) require that if an OR is open then at least one surgery is scheduled to that OR. Constraints (2.6) ensure that the surgery start time of two patients $j$ and $j'$ cannot be the same, i.e., patient $j$ either starts earlier or later than patient $j'$. Constraints (2.7) indicate that each patient should be scheduled to exact one recovery bed. Constraints (2.8) ensure that the recovery start time of two patients $j$ and $j'$ cannot be the same, i.e., patient $j$ enter the recovery area either earlier or later than patient $j'$. Constraints (2.9) specify that a patient can stay longer in an OR than the scheduled surgery time. This situation happens when there is no available bed in the recovery area and the patient's recovery process begins in the OR just after his/her surgery. Constraints (2.10) and (2.11) ensure that if patient $j$ is not scheduled to OR $i$ then the start time and end time of patient $j$ in OR $i$ are forced to be 0. Constraints (2.3), (2.10), and (2.11) imply that both $b_{ji}$ and $c_{ji}$ have only one non-zero value for all $i$, and the non-zero values of the two set variables are the start time and end time of patient $j$ in ORs. Constraints (2.12) state the end time of patient $j$ in ORs equal to the start time of patient $j$ in ORs plus the actual duration that patient $j$ stays in ORs. Constraints (2.13) state that the open time of an OR must cover all the surgery operation scheduled to that OR, i.e., the close time of an OR must equal to the end time of the last surgery scheduled to that OR. Constraints (2.14) specify that if the surgery start time of patient $j$ in an OR is earlier than patient $j'$ then $\beta_{jj'}$ is forced to be 1. Constraints (2.15) ensure an OR cannot operate two surgeries at the same time. If both patients $j$ and $j'$ are operated in OR $i$ ($\alpha_{ji} = \alpha_{j'i} = 1$) and patient $j$ starts earlier ($\beta_{jj'} = 1$), then the start time of $j'$ ($b_{j'i}$) must be greater than the end time of $j$ ($c_{ji}$) plus the setup time ($T_{jj'}$) between these two surgeries. Constraints (2.16) and Constraints (2.17) ensure that if patient $j$ is not scheduled to bed $k$ in the recovery area, then the start time and end time of patient $j$ on bed $k$ are forced to be 0. Constraints (2.7), (2.16), and (2.17) imply that both $e_{jk}$ and $f_{jk}$ have only one non-zero value for all $k$, and the non-zero values of the two set variables are the recovery start time and end time of patient $j$. Constraints (2.18) state the end time of patient $j$ in the recovery area equals the start time of patient $j$ in the recovery area plus the actual duration that patient $j$ stays in the recovery area. Constraints (2.19) specify that if the start time of patient $j$ in the recovery area is earlier than patient $j'$ then $b_{jj'}$ is forced to be 0. Constraints (2.20) ensure a bed in the recovery area cannot be occupied by two patients at the same time, which are similar as Constraints (2.14). Constraints (2.21) state that the time that a patient enters the recovery area equals the time that the patient leaves the OR area. Constraints (2.22) and Constraints (2.23) ensure that each surgeon conducts his/her surgeries during the interval when he/she is available. Constraints (2.24), similar to (2.14) and (2.20), ensure a surgeon cannot operate two surgeries at the same time. Constraints (2.25) define the length of overtime of each OR. Since the objective is to minimize the cost, if OR $i$ is not over-operated then $z_i$ is forced to be 0, and if it is over-operated then $z_i$ is forced to be equal to the length of overtime. Constraints (2.26) and Constraints (2.27) are integrity constraints.

### 2.4.2 Deterministic Constraint Programming Model

Before introducing the proposed the CP model for the elective surgery scheduling problem, we first talk about the difference between the CP and MIP as well as the techniques used to solve CP models.

In general, the CP models differ from the MIP models in three aspects. First, the main focus of the two types of models are different. The main focus of the MIP models is the objective function and optimality, while the CP models mainly focus on the constraints and feasibility. Second, the constraint representation is also different for the two types of models. The constraints in the MIP models are linear or nonlinear inequalities, while the CP models have logical constraints. Third, the search methods used by the two types of the models are different. The MIP models use relaxation of the models to eliminate suboptimal solutions, while the CP models use heuristics, including propagation and branch & prune, to eliminate infeasible configurations and reduce the search space.

The CP problem can be divided into two categories, one is constraint satisfaction problems, and the other is constraint optimization problems (used in our study). A solution to a constraint satisfaction problem is an assignment of values from the variables' domains which satisfy all the constraints. The main idea of solving a constraint satisfaction problem is to reduce the variable domains by eliminating infeasible configurations. Specifically, if a value from the domain of one variable cannot satisfy all the constraints related to this variable, the value will be eliminated from the domain. Execute this consistency check repeatedly for each value until no value can be removed for any variable. After this procedure, if the domain is empty for any variable, then the problem is infeasible; if only one value is left in the domain for each of the variable, then the single solution is obtained; otherwise, search methods are applied to find the best solutions for constraint optimization problems.

Several techniques are applied to solve the CP model, and these techniques can be divided into five major categories Bartak (2005):

- *Systematic search algorithm* The generate-and-test method, which generates each possible combination of the variable's value and test to see whether this combination satisfies all constraints, is the most popular one to solve the constraint satisfaction problem. Among the algorithms used to systematically search the value combinations, the backtracking algorithm is the most common one. Based on a partial solution, the backtracking algorithm attempts to develop a complete solution by repeatedly choosing a value for the remaining variables (which are not included in the partial solution).

- *Consistency techniques* When the value combination cannot satisfy any of the constraints, we say that there is an inconsistency in this value combination. It is important to identify the inconsistency soon to improve the efficiency of the search algorithm. The common consistency techniques include node-consistency, arc-consistency, and path consistency.

- *Constraint propagation* The systematic search method does not use the constraint to improve the efficiency, while the consistency techniques reduce the search space until a solution is found. The constraint propagation is introduced to embed a consistency algorithm inside a search algorithm, which takes the advantages of the two algorithms.

- *Variable and value ordering* Certain order of the variables and values in the domains can reduce the search efforts. The most common variable ordering is based on "fail-first" principle. The most common value ordering is based on "succeed-first" principle.

- *Constraint optimization* The constraint optimization model requires that all the solutions of constraint satisfaction problem are explored and compared based on the objective function. And the most widely used algorithm for solving optimization problems is the branch and bound algorithm.

The CP has become a promising approach for solving large-scale combinatorial problems. It has been applied to solve all kinds of scheduling and planning problems successfully, for example, airline crew scheduling problems, orchestra rehearsal scheduling problems, railway scheduling problems, and log-truck scheduling problems in the forest industry.

In the healthcare field, research has been done on applying CP to solve staff scheduling problem, especially nurse scheduling problems. Weil et al. (1995) proposed a CP model for a nurse scheduling problem. They considered constraints include personal policy, hospital policy, workload, nurse qualifications as well as individual requests. Abdennadher and Schlenker (1999) discussed a nurse scheduling problem and presented a prototype model that assists a planner in scheduling the nurse shifts for a hospital ward. The following constraints are taken into account in their model: legal regulations, organizational rules, personnel data, and personal requirements. Bourdais et al. (2003) developed a general CP approach for staff (doctor or nurses) scheduling in health care. The main categories of rules considered in their paper include staff demand, staff availability, shift distribution, and Ergonomics rules.

However, little work has been done on the CP for solving elective surgery scheduling problems. Hanset et al. (2010) built a CP model for ORs scheduling. In their model, the constraints on the surgeons and the material resources are considered.

In this study, we consider scheduling patients in both the OR area and recovery area, which is a new problem that never solved by the CP. We use the IBM® ILOG® CP Optimizer to model and solve the CP model. Some main aspects of building the CP model are discussed in details in the rest of this subsection.

**Decision Variables of the CP model**

In this CP model, two types of variables are needed. The interval variables are used to represent activities, which has five properties, including start time, end time, job size, duration, and status. An interval variable could be set to be optional, and if this is the case, its status could be present or absent. When an interval variable is absent, its other properties are all set to be 0. Some functions are built to retrieve the five properties of the interval variables. For example, $a$ is an interval variable, and the five functions $startOf(a)$, $endOf(a)$, $sizeOf(a)$, $lengthOf(a)$, and $presenceOf(a)$ can yield the five properties of $a$, respectively. Note that $sizeOf(a)$ represents the job size, while $lengthOf(a)$ represents the time spent to complete the job. Usually the two values are the same. However, the workers may take breaks when doing the jobs. In that case, $lengthOf(a)$ equals $sizeOf(a)$ plus the breaks of the workers.

In our model, the following interval variable arrays are defined. $\mathscr{A}$ contains $J$ members to represent the activity of each patient in ORs, one interval variable array. $\mathscr{B}$ contains $K$ members to represent the activity of each patient in the recovery area. $\mathscr{C}$ contains $I$ members to represent activities in each OR. $\mathscr{D}$ contains $K$ members to represent activities in each recovery bed. $\mathscr{E}$ contains $L$ members to represent activities of each surgeon. Besides, three interval variable arrays $\mathscr{U}$, $\mathscr{V}$, and $\mathscr{W}$, with size $J \times I$, $J \times K$ and $J \times L$, are defined to represent the allocation decisions of patients to ORs, recovery beds, and surgeons, respectively. Note that interval array $\mathscr{C}$, $\mathscr{U}$, $\mathscr{V}$, and $\mathscr{W}$ are set to be optional. The interval variable $\mathscr{C}_i$ is present if and only if OR $i$ is open. Similarly, the interval variable $\mathscr{U}_{ji}$ ($\mathscr{V}_{jk}$) is present if and only if the patient $j$ is allocated to OR $i$ (bed $k$), and the interval variable $\mathscr{W}_{jl}$ is present if and only if the patient $j$'s surgery is operated by surgeon $l$.

The other variable needed to complete the model is the interval sequence variable, which represents a total order over a set of interval variables (activities). Note that the absent interval variables are not considered in the ordering. A non-negative integer can be associated with each interval variable to represent its type.

In our model, we create three interval sequence variable arrays $\mathscr{X}$, $\mathscr{Y}$, and $\mathscr{Z}$, which contain $I$ members to represent the sequence of surgeries in each OR, $K$ members to represent the sequence of patients in each recovery bed, and $L$ members to represent the sequence of surgeries operated by each surgeon, respectively. Each interval sequence variable contains two parts, an array of interval variables and an integer array to represent the types of these intervals. For example, $\mathscr{X}_i(\{\mathscr{U}_{1i}, \cdots, \mathscr{U}_{Ji}\}, \mathscr{T})$ represents the total ordering of the surgeries in OR $i$, where the present interval variables in $\{\mathscr{U}_{1i}, \cdots, \mathscr{U}_{Ji}\}$ are the surgeries scheduled to OR $i$ and the integer array $\mathscr{T}$ contains $J$ members and each represents the type of the corresponding surgery.

**Constraints of the CP model**

In this subsection, we main discuss four types of important constraints in the CP Optimizer, *alternative*, *span*, *noOverlap*, and *endAtStart*, which show the advantages of the CP on modeling logic constraints.

The first one is the *alternative* constraint, which is used to model the selection of one interval from a set of optional intervals. We use *alternative* constraint to specify that patient $j$'s surgery can be assigned to one and only one OR as

$$alternative\left(\mathscr{A}_j, \{\mathscr{U}_{j1}, \cdots, \mathscr{U}_{jI}\}\right)$$

This constraint models an exclusive alternative between interval variables in set $\{\mathscr{U}_{j1}, \cdots, \mathscr{U}_{jI}\}$, i.e., exactly one of the interval variable in this set will be present and it starts and ends together with $\mathscr{A}_j$. This *alternative* constraint is equivalent to the following three constraints on the properties of the interval variables:

$$\begin{cases} \sum_{i=1}^{I} presenceOf(\mathscr{U}_{ji}) = 1 \\ startOf(\mathscr{A}_j) = \max\left\{startOf(\mathscr{U}_{ji}), i = 1, \cdots, I\right\} \\ endOf(\mathscr{A}_j) = \max\left\{endOf(\mathscr{U}_{ji}), i = 1, \cdots, I\right\} \end{cases}$$

Recall that when an interval variable is absent, all the five properties will be 0. The first constraint indicates that only one interval variable will be present, in other words, the surgery $j$ cannot be allocated to more than one OR. Since the start time and end time of an absent interval are 0, the right hand sides of the second and third constraint will return the start time and end time of the interval that is present. In the similar way, the constraints that each patient can assign to one and only one recovery bed can be specified as

$$alternative\left(\mathscr{B}_j, \{\mathscr{V}_{j1}, \cdots, \mathscr{V}_{jK}\}\right).$$

The second constraint *span* is used to specify the relationship between interval variables. We use the *span* constraint to model that the open hours of OR $i$ must exactly cover a set of surgeries allocated to it as

$$span\left(\mathscr{C}_i, \{\mathscr{U}_{1i}, \cdots, \mathscr{U}_{Ji}\}\right).$$

This constraint requires that the interval $\mathscr{C}_i$ starts at the start time of the first present interval from the set $\{\mathscr{U}_{1i}, \cdots, \mathscr{U}_{Ji}\}$ and ends together with the last one. Since the interval variables in the set are optional, if none of the interval variables in this set are present, the interval variable $\mathscr{C}_i$ is forced to be absent. This *span* constraint is equivalent to the following three constraints on the properties of the interval variables:

$$\begin{cases} \sum_{j=1}^{J} presenceOf(\mathscr{U}_{ji}) \leq |J| presenceOf(\mathscr{C}_i) \\ startOf(\mathscr{C}_i) = \min\left\{startOf(\mathscr{U}_{ji}), j = 1, \cdots, J\right\} \\ endOf(\mathscr{C}_i) = \max\left\{endOf(\mathscr{U}_{ji}), j = 1, \cdots, J\right\} \end{cases}$$

The first constraint indicates that when $\mathscr{C}_i$ is absent, then all $\mathscr{U}_{ji}$ will be forced to be 0, in other words, the surgeries cannot be scheduled to a closed OR. The second states that the start time of OR $i$ equals to the earlier start time of the surgeries assigned to this OR. The third one means that OR $i$ is close when the last allocated surgeries is finished. The second and third constraints together define the total occupied time of OR $i$. Similarly, the constraints that the occupied time of a recovery bed must exactly cover the set of patients allocated to it can be specified as

$$span\left(\mathscr{D}_k, \{\mathscr{V}_{1k}, \cdots, \mathscr{V}_{Jk}\}\right),$$

and the constraints that the surgeries must be operated during the available time of a surgeon can be written as

$$span\left(\mathscr{E}_l, \{\mathscr{W}_{1l}, \cdots, \mathscr{W}_{Jl}\}\right).$$

The third constraint *noOverlap* is used to guarantee that there is no overlap between any two activities, either surgery operations or setup work, in one OR. As we stated before, the setup times are sequence-dependent. And in CP optimizer the setup times are modeled as an instance of the *TransitionDistance* class, named $\mathscr{G}$. This instance has a $N \times N$ table of non-negative numbers, where $N$ is the total number of surgery types, and $\mathscr{G}_{n,n'}$ represents the amount of time that must elapse before a type $n'$ surgery is started when a type $n$ surgery precedes it. The *noOverlap* constraint takes an interval sequence variable and an instance of object *TransitionDistance* as its arguments. And constraint

$$noOverlap\left(\mathscr{X}_i(\{\mathscr{U}_{1i}, \cdots, \mathscr{U}_{Ji}\}, \mathscr{T}), \mathscr{G}\right)$$

requires that the surgeries scheduled to OR $i$ are non-overlapping, and the minimum time must elapse between two surgeries in the sequence are specified in the table $\mathscr{G}$. Note that the absent interval variables are automatically removed from the sequence in the solution. This constraint is easier to explain by using the propositional logic. We define three propositions $p_{jj'}$, $r_{ji}$, and $q_{jj'}$ on the interval variable $\mathscr{U}_{ji}$ and $\mathscr{U}_{j'i}$ as:

$$\begin{cases} p_{jj'} : startOf\left(\mathscr{U}_{ji}\right) < startOf\left(\mathscr{U}_{j'i}\right); \\ r_{ji} : presenceOf(\mathscr{U}_{ji}) \\ q_{jj'} : endOf\left(\mathscr{U}_{ji}\right) < startOf\left(\mathscr{U}_{j'i}\right) - \mathscr{G}\left(\mathscr{T}_j, \mathscr{T}_{j'}\right) \end{cases}$$

And the above *noOverlap* constraint requires that the propositional sentence

$$\left(p_{jj'} \wedge r_{ji} \wedge r_{j'i}\right) \Rightarrow q_{jj'}$$

is true for any pair of interval variables $\mathscr{U}_{ji}$ and $\mathscr{U}_{j'i}$, which means that for any two pair of surgeries $j$ and $j'$ that allocated to OR $i$, if surgery $j$ starts earlier, then surgery $j'$ must wait at least $\mathscr{G}\left(\mathscr{T}_j, \mathscr{T}_{j'}\right)$ to start after surgery $j$ is finished. In the similar way, the constraints that there is no overlap between any two patients' recovery in one bed can be specified as

$$noOverlap\big(\mathscr{Y}_k(\{\mathscr{V}_{ik}, \cdots, \mathscr{V}_{Jk}\})\big);$$

and the constraints that there is no overlap between any two surgery operations for each surgeon can be written as

$$noOverlap\big(\mathscr{Z}_l(\{\mathscr{W}_{il}, \cdots, \mathscr{W}_{Jl}\})\big).$$

Note that in the last two *noOverlap* constraints, since the turnover times for recovery beds and surgeons are not considered here, the interval sequence variables only have one argument.

The last constraint is the *endAtStart* constraint, which is used to model the relative position of interval variables. We use *endAtStart* constraint to specify that the patient $j$ is sent to the recovery area right after leaving the OR as

$$endAtStart(\mathscr{A}_j, \mathscr{B}_j, 0),$$

which is equivalent to $startOf(\mathscr{B}_j) = endOf(\mathscr{A}_j) + 0$, i.e., the start time of the recovery of patient $j$ equals to the time when patients leave the OR area.

The rest of the constraints can be expressed by the properties of the interval variables easily. $sizeOf(\mathscr{A}_j) + sizeOf(\mathscr{B}_j) = S_j + R_j$ states that the sum of the actual times patient $j$ spending in OR area and recovery area should be equal to the sum of the surgery durations and recovery time. $presenceOf(\mathscr{W}_{jl}) = \Theta_{jl}$ ensures that if surgery $j$ cannot be operated by surgeon $l$ then the interval variable $\mathscr{W}_{jl}$ must be absent, otherwise it should be present. $startOf(\mathscr{W}_{jl}) = startOf(\mathscr{A}_j)\Theta_{jl}$ $endOf(\mathscr{W}_{jl}) = (startOf(\mathscr{A}_j) + S_j)\Theta_{jl}$ specifies the surgeon's start time and end time on a surgery. Constraint $presenceOf(\mathscr{U}_{ji}) <= \Phi_{ji}$ ensures that if surgery $j$ cannot be operated in OR $i$ then the interval variable $\mathscr{U}_{ji}$ is forced to be absent. Note that in the above constraints, the parameters $G_i$, $S_j$, $R_j$, $P_l$, $Q_l$, $\Theta_{jl}$, $\Phi_{jl}$ are defined in Section 2.4.1.

**Objective function of the CP model**

The objective function, minimizing the fixed costs as well as the overtime costs of the ORs, is fairly easy by the use of the properties of the interval variables. The status of an interval variable $\mathscr{C}_i$ can be obtained by using the function $presenceOf(\mathscr{C}_i)$. In addition, the actual operating time of OR $i$ can be obtained by using the function $lengthOf(\mathscr{C}_i)$, and the overtime of OR $i$ can be obtained by comparing the actual operating time with the normal operating time. Then the cost of OR $i$ is expressed as

$$F_i \times presenceOf(\mathscr{C}_i) + O_i \times \mathrm{Max}(0, (lengthOf(\mathscr{C}_i) - H_i)),$$

where the $F_i$ is the fixed cost of opening OR $i$, $H_i$ is the normal operating time of OR $i$, and $O_i$ is the unit overtime cost of OR $i$. And the total cost of the surgical center is the sum of the costs over all opened ORs.

### 2.4.3 Computational Results

In the first part of this section, we compare the computational performance of the MIP model with the CP model. And the second part presents the result about the sensitivity analysis of the total costs obtained by the CP model under small and large variations in surgery durations as well as recovery times. The last part of this section shows the performance of the CP models on a relatively large instance.

The purpose of the numerical experiment is to test the performance of the two models, not solve the actual problems of a certain surgical center. Therefore, these two models are tested on 18 randomly generated instances, and the sizes of these instances are listed in Table 2.2. The total number of OR and the types of surgeries that could be operated in each OR are fixed for all instances, and the total number of ORs is set to be 10 to make sure the largest instance (30 surgeries, 8 surgeons, 5 recovery beds) feasible. The 18 instances are divided into 3 sets with 4, 6, and 8 surgeons, respectively. In the first set, the 4 surgeons are identical, and the first ten surgeries of instance“14S-4S-3B” are the same as the instance “10S-4S-2B”. Similar settings apply to other instances. There are 5 types of surgeries, and for each surgery, we randomly generate an integer between 1 and 5 and assign it as the type of the surgery. Previous research shows that the log-normal distributions are suitable to model the surgery durations (Spangler et al., 2004; Jebali et al., 2006; Pandit and Carey, 2006). And thus, for the 5 types of surgeries, we generate the surgery durations by following log-normal distributions with means equal to $72, 49, 103, 173, 135$ minutes and standard deviations equal to $23, 13, 25, 33, 32$ minutes, respectively. Note that the parameters are set by referring the data reported in the study of Pandit and Carey (2006). The duration of a surgery is set to be no less than 25 minutes nor more than 240 minutes. The setup time for a surgery is sequence dependent, which is shown in Table 2.1 in details. For example, if a surgery is type 2 and its preceding one is type 4, then the setup time is 20 minutes. Depending on the surgery type, the recovery time of each surgery is generated from uniform distributions $U(30,72)$, $U(25,49)$, $U(65,103)$, $U(120,173)$, $U(90,135)$, respectively. The surgeons' start times and end times are generated from uniform distributions $U(0,50)$ and $U(400,720)$, respectively. For all ORs, the normal operating time $H_i = 480$ minutes and the maximal operating time $G_i = 720$ minutes, $1 \leq i \leq I$. The cost coefficients for all the ORs are the same: the fixed cost of an OR during the normal operating time $F_i = 8000$; and the overtime cost per minute is about 1.75 times of the regular costs (Dexter and Traub, 2002), i.e., $1.75\frac{8000}{480} = 29.17 \approx 30 = O_i$, $1 \leq i \leq I$.

Both the two models are built and solved by IBM® ILOG® on a personal computer with Intel® Core™i7 2.20 GHz processor and 8.0 GB RAM. We apply the CPLEX Optimizer to solve the MIP model and CP Optimizer to solve the CP model. Currently, the time limits of running the two models are set to be 600 seconds. Besides the time limits, both the two models are terminated

**Table 2.1:** Sequence dependent setup time (minutes).

| Setup time | | Surgery Type 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| Surgery Type | 1 | 10 | 15 | 20 | 25 | 30 |
| | 2 | 15 | 10 | 15 | 20 | 25 |
| | 3 | 20 | 15 | 10 | 15 | 20 |
| | 4 | 25 | 20 | 15 | 10 | 15 |
| | 5 | 30 | 25 | 20 | 15 | 10 |

when a feasible solution proved to be within 1% of the optimal solution, i.e., relative gap is set to be 1%.

### The Comparison of the Two Models

The performance of the two models on the 18 instances are compared from four aspects, including the number of constraints and variables, objective values, and computational time. The results are shown in Table 2.2. The first column of the table shows the name of the instances, for example, *"10S-4S-2B"* means there are total 10 surgeries, 4 surgeons, and 2 recovery beds considered in this instance. For each model, the first two columns *"#Cons."* and the *"#Var."* represent the number of constraints and variables, respectively. The column of *"Obj."* shows the best objective value found when the program terminated. The column *"Time(s)"* is the computational time (in seconds) when the value in column *"Obj."* is obtained, i.e., the time either at which the optimal solution is found or after that the solution does not get improved. The column *"Imp%"* presents the percentage improvement of the CP solution over the MIP solution. The column *"BestLB"* is the best lower bound of the objective value obtained by the Cplex solver. And the *"MIPGap"* and *"CPGap"* are relative gaps between the objective value obtained of the two models and the best lower bound.

For all the instances, both the MIP model and the CP model can give feasible integer solutions, however, both of them don't perform very well on finding optimal ones. Specifically, the MIP model only finds optimal solutions for *"10S-4S-1B"* and *"10S-4S-2B"*, while the CP model only finds optimal solutions for *“10S-4S-2B”*, and all of them are marked with ”*” in column *“Time(s)”*. Note that for instance *“10S-4S-1B”*, the CP model also finds the optimal solution, but the program is only terminated when reaching the time limit.

For the rest of the 16 instances, the best lower bounds are relatively small, which lead to large relative gaps of the two solutions. Note that in our MIP model, all the scheduling decision variables, including OR status, allocation, and sequence, are binary. The best lower bound is obtained by relaxing these binary variables to continuous ones. After relaxing, starting all surgeries in one OR at the beginning of the day will be a feasible solution, i.e., the sequencing and no-overlap constraints are not binding any more. However, this relaxed solution could be far away from being feasible to the original problem. Therefore, the lower bound as well as the relative gap cannot be used as a strong reference to evaluate the quality of the solutions.

The CP technique is efficient to model and solve the surgery scheduling problem. Both the number of constraints and variables of the CP model are smaller than the MIP model. Moreover, the number of variables of the MIP model grows faster along with the increase of the instance size. Specifically, for the largest instance, i.e., *“30S-8S-5B”*, there are total 26280 constraints and 3151 variables in the MIP model, which are 14.5 and 3.8 times of the CP model, respectively. Partially due to the smaller set of constraints and variables, compared with the MIP model, the CP model is solved faster and the yield solutions are better. Among the 18 instances, the MIP model only performs slightly better than the CP model on one instance *“16S-4S-3B”*. On average, the best objective value of the CP model is 6.43% better. And along with the increase of the instance size, the advantage of the CP model is more significant. In addition to better solutions, the CP model takes less time in general. The average computational time of CP is about 182 seconds, while the average computational time of MIP is about 400 seconds. As shown above, CP finds the optimal solution for instance *“10S-4S-1B”* but does not report it as optimal. In other words, the CP model is not a perfect resort for solving the problem to optimality due to its search methods. However, the CP model does have advantages in getting sub-optimal solutions in short time, and these sub-optimal solutions are better than the MIP model.

One more advantage of the CP model is the form in which the solutions are represented. The CP Optimizer not only gives the value of the variables but also provides the Gantt chart for each corresponding solution. Figure 2.4 shows Gantt charts of ORs, recovery beds, and surgeons for the solution of instance *"10S-4S-1B"*. In the three Gantt charts, each row represents the schedule of an OR, a recovery bed, or a surgeon, and each block represents a patient's activity. Different types of surgeries are represented with different colors and patterns. The legends at the end of the first two rows show the types of the surgeries can be performed in the two ORs.

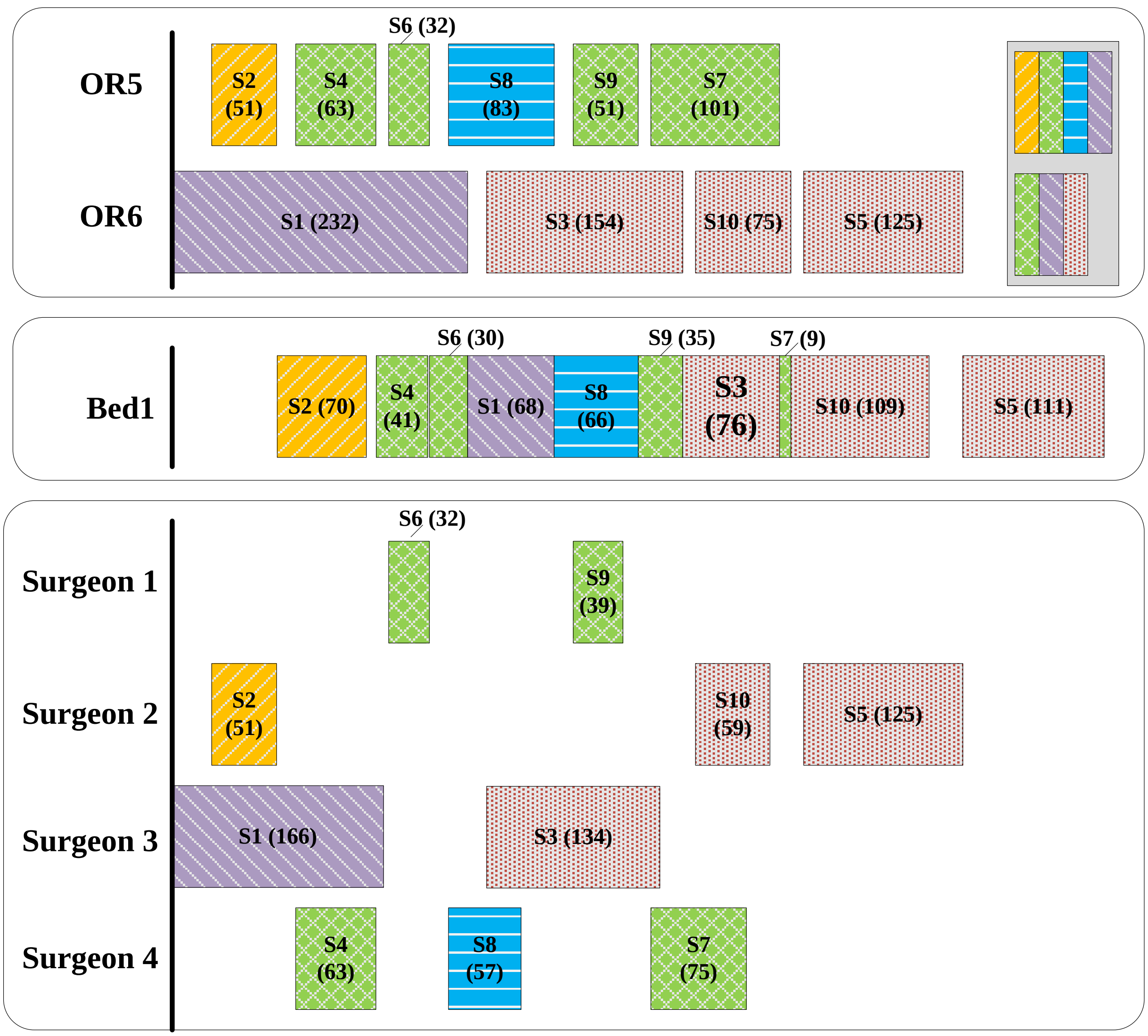


**Figure 2.4:** The Gantt chart for ORs, recovery beds, and surgeons by CP Optimizer

**Table 2.2:** The computational performance of the two models on 18 random instances

| | MIP | | | | CP | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Instance | #Cons. | #Var. | Obj. | Time(s) | #Cons. | #Var. | Obj. | Time(s) | Imp% | Best LB | MIPGap | CPGap |
| 10S-4S-2B | 2580 | 581 | 16,090 | 98* | 478 | 222 | 16,090 | 1.4* | 0 | 16000 | 1% | 1% |
| 10S-4S-1B | 2460 | 551 | 20,230 | 164* | 476 | 210 | 20,230 | 179 | 0 | 20016 | 1% | 1% |
| 14S-4S-3B | 4944 | 955 | 23,650 | 193 | 644 | 314 | 23,350 | 0.8 | 1.27% | 16000 | 32% | 31% |
| 14S-4S-2B | 4720 | 913 | 23,470 | 542 | 642 | 298 | 23,350 | 20 | 0.51% | 16000 | 32% | 31% |
| 16S-4S-3B | 6318 | 1151 | 28,140 | 596 | 726 | 354 | 28,260 | 3.3 | -0.43% | 16000 | 43% | 43% |
| 16S-4S-2B | 6030 | 1103 | 31,620 | 596 | 724 | 336 | 29,940 | 548 | 5.31% | 16205 | 49% | 46% |
| 18S-6S-3B | 8544 | 1363 | 41,090 | 89 | 964 | 434 | 38,480 | 77 | 6.35% | 16000 | 61% | 58% |
| 18S-6S-2B | 8184 | 1309 | 46,370 | 596 | 962 | 414 | 44,060 | 108 | 4.98% | 16000 | 65% | 64% |
| 20S-6S-3B | 10850 | 1651 | 45,560 | 140 | 1064 | 500 | 42,880 | 525 | 5.88% | 9315 | 80% | 78% |
| 20S-6S-4B | 10410 | 1591 | 50,000 | 217 | 1062 | 478 | 45,100 | 101 | 9.80% | 11031 | 78% | 76% |
| 24S-6S-3B | 15318 | 2167 | 56,710 | 541 | 1260 | 592 | 49,330 | 38 | 13.01% | 8000 | 86% | 84% |
| 24S-6S-4B | 14690 | 2095 | 66,150 | 595 | 1258 | 566 | 58,620 | 38 | 11.38% | 8652 | 87% | 85% |
| 20S-8S-3B | 11690 | 1651 | 43,180 | 546 | 1236 | 544 | 41,290 | 108 | 4.38% | 16118 | 63% | 61% |
| 20S-8S-4B | 11250 | 1591 | 45,940 | 359 | 1234 | 522 | 44,380 | 187 | 3.40% | 17159 | 63% | 61% |
| 24S-8S-5B | 17142 | 2239 | 50,230 | 496 | 1466 | 670 | 46,990 | 448 | 6.45% | 16000 | 68% | 66% |
| 24S-8S-4B | 16518 | 2167 | 52,470 | 597 | 1464 | 644 | 48,010 | 332 | 8.50% | 16133 | 69% | 66% |
| 30S-8S-5B | 26280 | 3151 | 76,000 | 594 | 1808 | 826 | 68,210 | 387 | 10.25% | 16000 | 79% | 77% |
| 30S-8S-4B | 25320 | 3061 | 83,740 | 239 | 1806 | 794 | 73,870 | 140 | 11.79% | 12017 | 86% | 84% |

**Sensitivity Analysis**

The solutions of the two deterministic models specify the number of ORs to open as well as the allocation and the sequence of the elective surgeries to the opened ORs, and the costs of the ambulatory surgical center are estimated based on the resulting scheduling decisions. However, the actual costs of the ambulatory surgical center could be very close to or far away from the estimated costs. The scheduling decisions are made based on the estimated surgery durations and recovery times, and the differences between the actual durations and the estimated ones determine the extent of the costs' differences. We conduct some experiment on the sensitivities of the total cost to these variations.

The actual durations of surgery and recovery with small and large variability are simulated by following the methods in Hsu et al. (2003): multiplying the estimated durations with random numbers from normal distribution $N(1, 0.15)$ and $N(1, 0.25)$, respectively. Specifically, take surgery $j$ as an example, MIP and CP models use its estimated durations $S_j$ and $R_j$; the actual durations of surgery $j$ $S'_j$ and $R'_j$ are random numbers instead of deterministic. $S'_j(R'_j)$ could be simulated as $S_j(R_j) \times r$; depending on the variation of the surgery duration is small or large, $r$ is drawn from normal distribution $N(1, 0.15)$ or $N(1, 0.25)$. For each of the 18 instances, we do the sensitivity analysis of the total costs to the small and large variations of the surgery durations, i.e., total 36 cases. For each case, we first simulate the actual durations by the methods above. Then the actual cost is calculated based on the scheduling decision obtained from the CP model in part 1 of this section, and for convenience, we call it "Cost A". In addition, we use the actual durations as inputs and re-run the CP model to get an objective value, which is the cost when one has perfect information on the durations, and for convenience, we call it "Cost B". The two costs are compared, and if the "Cost A" is not far away from the "Cost B", then we conclude that the deterministic model is a good estimation of the reality, otherwise stochastic model should be considered. The results are shown in Table 2.3.

In Table 2.3, the first column lists the instance names which have the same meanings as the ones in Table 2.2. For small and large variations, three columns are listed, including "Cost A", "Cost B" and "Diff%". Note that for instance *"24S-6S-4B"* with large variability and instance

**Table 2.3:** Sensitivity analysis of the total costs to the variation of surgery durations and recovery times

| | Small Variation | | | Large Variation | | |
|---|---|---|---|---|---|---|
| Instance | Cost A | Cost B | diff% | Cost A | Cost B | diff% |
| 10S-4S-2B | 16630 | 16600 | 0.18% | 17290 | 16000 | 8.06% |
| 10S-4S-1B | 21190 | 20920 | 1.29% | 20530 | 18940 | 8.39% |
| 14S-4S-3B | 24450 | 24000 | 1.88% | 21970 | 21910 | 0.27% |
| 14S-4S-2B | 24730 | 24000 | 3.04% | 23260 | 21970 | 5.87% |
| 16S-4S-3B | 28890 | 28590 | -1.04% | 29610 | 28290 | 4.67% |
| 16S-4S-2B | 32130 | 31530 | 1.90% | 34440 | 29430 | 17.02% |
| 18S-6S-3B | 41000 | 38750 | 5.81% | 38720 | 36710 | 5.48% |
| 18S-6S-2B | 50120 | 44800 | 11.88% | 55930 | 41780 | 33.87% |
| 20S-6S-4B | 43840 | 43390 | 1.04% | 47260 | 40040 | 18.03% |
| 20S-6S-3B | 47860 | 45970 | 4.11% | 49060 | 45670 | 7.42% |
| 24S-6S-4B | 51160 | 49960 | 2.40% | Inf | 47260 | - |
| 24S-6S-3B | 61500 | 59550 | 3.27% | 61710 | 51250 | 20.41% |
| 20S-8S-4B | 44560 | 42940 | 3.77% | 46720 | 40630 | 14.99% |
| 20S-8S-3B | 48160 | 44650 | 7.86% | 50380 | 42340 | 18.99% |
| 24S-8S-5B | 47860 | 46360 | 3.24% | 52540 | 50340 | 4.37% |
| 24S-8S-4B | 51350 | 47740 | 7.56% | 53110 | 50670 | 4.82% |
| 30S-8S-5B | Inf | 65180 | - | Inf | 67670 | - |
| 30S-8S-4B | Inf | 72970 | - | Inf | 73660 | - |

*"30S-8S-4B"* and *"30S-8S-5B"* with small and large variations, the scheduling decisions based on the estimated durations are not feasible. The schedule managers need to do lots of work to resolve the violations.

In general, the differences between the "Cost A"s and the "Cost B"s are small when the variability is small, and along with the growth of the variability of surgery durations, the costs' differences increase. Specifically, on average the "Cost A" is 4.27% and 11.53% higher than the "Cost B" when the variation of the surgery durations is small and large, respectively.

Depending on the risk the ambulatory surgical center is willing to take, different strategies should be taken. For example, if they can accept to pay 4.27% more than the estimated cost and believe the variation is small, then the current deterministic model works. Otherwise, the stochastic models should be developed to model these variations.

### Performance of the CP Model on Large instances

The performance of the CP model on 5 extremely large instances are tested in this section. As our university medical center is one of the largest hospitals in this metro area, we design instances based on the size of their business accordingly. The medical center has 30 ORs and provides a wide range of service in generally 10 different areas, including orthopedic, general, neuro, vascular, pediatric, podiatry, OMFS, ENT, ophthalmology, urology, and cardio. The average volume per day is about 70. Since the cardio surgeries are different from others and have their own 2 dedicated ORs, we consider 28 ORs, and 9 different surgery types in our computational experiment. The number of surgeries ranges from 60 to 80 given the average case number is 70. And there are 26 surgeons and 25 recovery beds for all the 5 instances. The running time limit is set to be 1800 seconds given the problem size is big.

The problem sizes as well as the solutions are reported in Table 2.4. The first 5 columns have the same meaning as Table 2.2. Note that the instances are named by the number of the surgeries. In addition, we add a new column in this table to show the number of integer solutions CP has obtained. The proposed constraint programming model can be solved within 30 minutes. And the average number of feasible solutions is 52 for these 5 instances, which means the reported solutions have been improved for 52 times on average. With these facts, we have confidence in the performance of the CP model for solving large size problems.

**Table 2.4:** Performance of CP on large instances

| Instance | #of Cons. | # of Var. | Obj. Value | Time | # of Integer Solutions |
|---|---|---|---|---|---|
| 60Sur | 10218 | 5078 | 185760 | 1093 | 53 |
| 65Sur | 11043 | 5488 | 195950 | 1468 | 75 |
| 70Sur | 11868 | 5898 | 211020 | 1417 | 52 |
| 75Sur | 12693 | 6308 | 234610 | 1553 | 49 |
| 80Sur | 13518 | 6718 | 243930 | 1693 | 33 |

## 2.5 Two Stage Stochastic Programming

In this Section, the assumption of deterministic durations is relaxed, and the problem is formulated as a stochastic programming model. The remaining of this Section is organized as follows. After a brief introduction to stochastic programming models in Section 2.5.1, the formulation of the two-stage stochastic mixed integer programming and solving methods are presented in Section 2.5.2 and Section 2.5.3, respectively. And finally, the results of the computational experiment are presented in Section 2.5.4.

### 2.5.1 Introduction on Stochastic Programming Models

Stochastic programming is used to solve optimization problems under uncertainty. One basic assumption is that the probability distributions of random parameters are known or can be estimated. The decisions must take before the realization of the observed random parameters. It is desirable that the decisions are feasible for all possible data instances. However, this cannot be achieved in some adverse situations. The chance constraints are introduced to allow a small probability violation of the constraints. The objective of stochastic programming model is to find the optimal decision to minimize the expectation of some functions.

The two-stage stochastic programming models are the most widely applied and studied stochastic programming models. In the first stage, some actions are taken; and then at the second stage, after some random events happen some recourse actions are taken; the objective value is then finalized, which is affected by both the first stage actions, the random events, and the recourse actions. When there is only one random parameter and it follows a discrete probability distribution, a common method to solve the two-stage stochastic programming model is to define a number of scenarios to represent the random outcomes and convert the models into their deterministic equivalence. Any solution approach for solving the deterministic equivalences can be applied when the number of possible scenarios is small. However, when the number of possible scenarios is large, it is necessary to exploit the structural properties of stochastic programs, and decompose them into smaller and more tractable components (Batun, 2011). For the models with many random parameters or the random parameters follow continuous probability distributions, one addition problem is to construct appropriate scenarios to approximate the outcome uncertainty. One approach is to use a finite number of random samples to represent the random outcomes, and solve the resulting deterministic programming models as one would do when the number of scenarios is finite.

Define the recourse function $Q(x,\omega) = \min_{y^\omega \in Y}\{q^T y^\omega : W y^\omega = h^\omega - T^\omega x\}$, and then the general form of the two stage stochastic linear programming model with fixed recourse can be written as:

$$\min_{x\in X}\{c^T x + \mathbb{E}_\omega[Q(x,\omega)] : Ax = b\} \tag{2.28}$$

where $X = \{x \in \mathfrak{R}^n : l \le x \le u\}$ defines the feasible set of the first stage action; $Y = \{y \in \mathfrak{R}^p : l' \le y \le u'\}$ describes the feasible set of recourse action; $q$ is the vector of recourse costs; $W$ is called recourse matrix with size of $m \times p$; and $\omega$ is the index of the scenarios If for any feasible first stage solution $x$, the second stage problem is feasible, we say that the stochastic program has relatively complete recourse.

The deterministic equivalence of the above stochastic model can be reformulated as:

$$\min \ \ c^T x + \mathbb{E}_\omega[q^T y^\omega] \tag{2.29}$$

Subject to

$$Ax = b \tag{2.30}$$

$$T^\omega x + W y^\omega = h^\omega \quad \forall \omega \in \Omega \tag{2.31}$$

$$x \in X \tag{2.32}$$

$$y^\omega \in Y \tag{2.33}$$

Note that Constraint (2.30) is considered as first stage constraints, while Constraint (2.31) is treated as second stage constraints. The deterministic equivalent is a big linear programming model with special structure, which could be solved by Bender's decomposition method.

### 2.5.2 Mathematical Formulation

By modifying the deterministic MIP model in Section 2.4.1, we develop a two stage stochastic mixed integer programming model in this section. Although some of the parameters, variables, and constraints in Section 2.4.1 are used in this section without any changes, we still redefine them here for the sake of convenience.

*Configuration Related Parameters:*

- $i$: the index of the ORs, $i = 1, 2, \cdots, I$
- $k$: the index of the beds in the recovery area, $k = 1, 2, \cdots, K$
- $j$: the index of the patients, $j = 1, 2, \cdots, J$
- $l$: the index of the surgeons, $l = 1, 2, \cdots, L$
- $F_i$: the fixed cost of OR $i$
- $O_i$: the overtime cost of OR $i$ per unit time
- $H_i$: the normal operating time of OR $i$
- $G_i$: the maximum operating time of OR $i$
- $M$: a sufficiently large number
- $T_{jj'}$: the setup time of patient $j'$ when previous patient is $j$
- $\Phi_{ji}$: binary, equal to 1 if and only if surgery $j$ could be operated in OR $i$
- $P_l$: the moment that surgeon $l$ begins available
- $Q_l$: the moment that surgeon $l$ begins unavailable
- $\Theta_{jl}$: binary, equal to 1 if and only if patient $j$ is operated by surgeon $l$

*Scenario Related Parameters*

- $\omega$: the index of the scenario
- $S_j^\omega$: the surgery duration of patient $j$
- $R_j^\omega$: the recovery time of patient $j$ after surgery

*First Stage Decision Variables*

- $y_i$: binary variable, equal to 1 if and only if OR $i$ is open
- $\alpha_{ji}$: binary variable, equal to 1 if and only if the surgery of patient $j$ is scheduled to OR $i$
- $\beta_{jj'}$: binary variable, equal to 1 if the start time of the surgery of patient $j$ is earlier than patient $j'$

- $\gamma_{jk}$: binary variable, equal to 1 if and only if patient $j$ is scheduled to bed $k$ in the recovery area
- $\delta_{jj'}$: binary variable, equal to 1 if the start time of patient $j$ in the recovery area is earlier than patient $j'$

*Second Stage Decision Variables*

- $a_j^{\omega}$: actual durations that patient $j$ stayed in an OR
- $b_{ji}^{\omega}$: start time of the surgery of patient $j$ in OR $i$, and equal to 0 if the surgery of patient $j$ is not operated in OR $i$
- $c_{ji}^{\omega}$: end time of the surgery of patient $j$ in OR $i$, and equal to 0 if the surgery of patient $j$ is not operated in OR $i$
- $d_{jk}^{\omega}$: start time of patient $j$ on bed $k$ in the recovery area, and equal to 0 if patient $j$ is not scheduled to bed $k$
- $e_{jk}^{\omega}$: end time of patient $j$ on bed $k$ in the recovery area, and equal to 0 if patient $j$ is not scheduled to bed $k$
- $x_i^{\omega}$: end time of OR $i$, and equal to 0 if OR $i$ is not open
- $z_i^{\omega}$: length of time that OR $i$ is over operated

Similar as the model (2.29)-(2.33), the two-stage stochastic programming model can be formulated as:

$$\min \sum_{i=1}^{I} y_i F_i + \mathbb{E}_{\omega}[\mathscr{Q}(y,\alpha,\beta,\gamma,\delta,\omega)] \tag{2.34}$$

subject to

$$\alpha_{ji} \leq \Phi_{ji} \quad \forall j,i \tag{2.35}$$

$$\sum_{i=1}^{I} \alpha_{ji} = 1 \quad \forall j \tag{2.36}$$

$$\sum_{j=1}^{J} \alpha_{ji} \leq |J| y_i \quad \forall i \tag{2.37}$$

$$\sum_{j=1}^{J} \alpha_{ji} \geq y_i \quad \forall i \tag{2.38}$$

$$\beta_{jj'} + \beta_{j'j} = 1 \quad \forall j, j'(j' \neq j) \tag{2.39}$$

$$\sum_{k=1}^{K} \gamma_{jk} = 1 \quad \forall j \tag{2.40}$$

$$\delta_{jj'} + \delta_{j'j} = 1 \quad \forall j, j'(j' \neq j) \tag{2.41}$$

$$y_i, \alpha_{ji}, \beta_{jj'}, \gamma_{jk}, \delta_{jj'} \in \{0,1\} \quad \forall j, j', i, k \tag{2.42}$$

where $\mathscr{Q}(y, \alpha, \beta, \gamma, \delta, \omega)$ is a recourse function, which is formulated as:

$$\mathscr{Q}(y, \alpha, \beta, \gamma, \delta, \omega) = \min \sum_{i=1}^{I} O_i z_i^{\omega} \tag{2.43}$$

subject to

$$a_j^{\omega} \geq S_j^{\omega} \quad \forall j \tag{2.44}$$

$$b_{ji}^{\omega} \leq G_i \alpha_{ji} \quad \forall j, i \tag{2.45}$$

$$c_{ji}^{\omega} \leq G_i \alpha_{ji} \quad \forall j, i \tag{2.46}$$

$$\sum_{i=1}^{I} c_{ji}^{\omega} = \sum_{i=1}^{I} b_{ji}^{\omega} + a_j^{\omega} \quad \forall j \tag{2.47}$$

$$x_i^{\omega} = \max\{c_{ji}^{\omega}, \forall j\} \quad \forall i \tag{2.48}$$

$$\sum_{i=1}^{I} b_{ji}^{\omega} - \sum_{i=1}^{I} b_{j'i}^{\omega} + M\beta_{jj'} \geq 1 \quad \forall j, j' \tag{2.49}$$

$$c_{ji}^{\omega} + T_{jj'}^{\omega} - b_{j'i}^{\omega} \leq (3 - \beta_{jj'} - \alpha_{ji} - \alpha_{j'i})M \quad \forall j, j'(j' \neq j), i \tag{2.50}$$

$$d_{jk}^{\omega} \leq M\gamma_{jk} \quad \forall j,k \tag{2.51}$$

$$e_{jk}^{\omega} \leq M\gamma_{jk} \quad \forall j,k \tag{2.52}$$

$$\sum_{k=1}^{K} e_{jk}^{\omega} = \sum_{k=1}^{K} d_{jk}^{\omega} + S_j^{\omega} + R_j^{\omega} - a_j \quad \forall j \tag{2.53}$$

$$\sum_{k=1}^{K} d_{jk}^{\omega} - \sum_{k=1}^{K} d_{j'k}^{\omega} + M\delta_{jj'} \geq 1 \quad \forall j, j' \tag{2.54}$$

$$e_{jk}^{\omega} - d_{j'k}^{\omega} \leq (3 - \delta_{jj'} - \gamma_{jk} - \gamma_{j'k})M \quad \forall j, j'(j' \neq j), k \tag{2.55}$$

$$\sum_{i=1}^{I} c_{ji}^{\omega} = \sum_{k=1}^{K} d_{jk}^{\omega} \quad \forall j \tag{2.56}$$

$$\sum_{l=1}^{L} \Theta_{jl} P_l \leq \sum_{i=1}^{I} b_{ji}^{\omega} \quad \forall j \tag{2.57}$$

$$\sum_{i=1}^{I} b_{ji}^{\omega} + S_j^{\omega} \leq Q_l \Theta_{jl} + (1 - \Theta_{jl})M \quad \forall j, l \tag{2.58}$$

$$\sum_{i=1}^{I} b_{ji}^{\omega} + S_j^{\omega} - \sum_{i=1}^{I} b_{j'i}^{\omega} \leq (3 - \beta_{jj'} - \Theta_{jl} - \Theta_{j'l})M \quad \forall j, j'(j' \neq j), l \tag{2.59}$$

$$x_i^{\omega} - H_i y_i \leq z_i^{\omega} \quad \forall i \tag{2.60}$$

$$x_i^{\omega}, z_i^{\omega}, a_j^{\omega}, b_{ji}^{\omega}, c_{ji}^{\omega}, d_{jk}^{\omega}, e_{jk}^{\omega} \geq 0 \quad \forall j, i, k \tag{2.61}$$

The explanations of Constraints (2.35-2.41), (2.44)-(2.60) are similar to those of Constraints (2.2-2.25), which are omitted here.

### 2.5.3 Solution Methods

As an extension of the Bender's decomposition method, the L-shaped method is one of the most widely applied approaches to solve the two stage stochastic linear programming models. The first stage of our model is an MIP with all binary variables, and the second stage is a linear programming model. Research has shown that the standard L-shaped algorithm is effective for solving this type of stochastic MIP model (Batun, 2011). The first stage problem is usually called master problem, and the second stage problem is called sub-problem or recourse problem. In the rest of this section, we will first present the standard L-shaped algorithm, then we will discuss about the changes we have made to improve the efficiency of the L-shaped algorithm on solving our problem.

#### L-Shaped Algorithm

The key idea of L-Shaped algorithm is to introduce an auxiliary variable $\theta$ to approximate the value of the expected recourse function $\mathbb{E}_\omega[\mathscr{Q}(y,\alpha,\beta,\gamma,\delta,\omega)]$. In each iteration of the algorithm, the master problem gets solved and returns an optimal solution, and then with respect to the first stage optimal solution, the sub-problem gets solved. If the termination condition is not satisfied, then, according to the status of the second stage solution, either a feasibility cut or an optimality cut is added to the master problem. By repeating the above process, the solutions causing infeasibility of the sub-problem get removed, and the objective value of the original problem get improved. In fact, it has been proved that the L-shaped algorithm will yield an optimal solution in a finite number of steps if the original problem is feasible (Birge and Louveaux, 2011). Next, we will discuss about the detailed implementation steps of the L-shaped algorithm.

*Step 0: Construct the initial master problem*

The auxiliary variable $\theta$ is introduced to approximate the value of the expected recourse function $\mathbb{E}_\omega[\mathscr{Q}(y,\alpha,\beta,\gamma,\delta,\omega)]$. Let $\theta_0$ be a valid lower bound of $\mathbb{E}_\omega[\mathscr{Q}(y,\alpha,\beta,\gamma,\delta,\omega)]$, then the initial master problem can be formulated as

$$\min \sum_{i=1}^{I} y_i F_i + \theta \tag{2.62}$$

Subject to

$$\theta \geq \theta_0, \tag{2.63}$$

as well as constraints (2.35) - (2.42).

*Step 1: Solve the master problem*

The master problem is solved to optimality, and the optimal solution $(\hat{y}, \hat{\alpha}, \hat{\beta}, \hat{\gamma}, \hat{\delta}, \hat{\theta})$ has been obtained.

*Step 2: solve the sub-problem*

For each scenario, given the value of scenario related parameters $S^\omega$ and $R^\omega$ and the first stage optimal solution $\hat{y}$, $\hat{\alpha}$, $\hat{\beta}$, $\hat{\gamma}$, and $\hat{\delta}$, the linear programming model (2.43)-(2.61) is solved. Note that when the sub-problem is infeasible, the objective value $\mathscr{Q}(\hat{y}, \hat{\alpha}, \hat{\beta}, \hat{\gamma}, \hat{\delta}, \omega)$ is set to be $\infty$.

After all the scenarios have been explored, the value of the expected recourse function is calculated as $\mathbb{E}_\omega[\mathscr{Q}(\hat{y}, \hat{\alpha}, \hat{\beta}, \hat{\gamma}, \hat{\delta}, \omega)] = \sum\limits_{\omega \in \Omega} p^\omega \mathscr{Q}(\hat{y}, \hat{\alpha}, \hat{\beta}, \hat{\gamma}, \hat{\delta}, \omega)$, where $p^\omega$ is the probability that scenario $\omega$ would happen and $\Omega$ is the set of possible scenarios with $\sum\limits_{\omega \in \Omega} p^\omega = 1$.

*Step 3: Check termination condition*

Depending on the value of $\mathbb{E}_\omega[\mathscr{Q}(\hat{y}, \hat{\alpha}, \hat{\beta}, \hat{\gamma}, \hat{\delta}, \omega)]$, three actions could be taken: 1) if it equals $\infty$, then go to Step 4 to add a feasibility cut; 2) if it is greater than $\hat{\theta}$, then go to step 5 to add an optimality cut based on duality; 3) otherwise the optimal solution is found, i.e., the current master problem solution results in the minimum expected second-stage cost.

*Step 4: Add feasibility cut*

According to the Farkas Lemma, for a system of $m$ linear inequalities $Ax = b, x \geq 0$ and a vector $\sigma \in \Re^m$ with $A^T\sigma \leq 0$, the following relationships must hold: 1) $\sigma^T b \leq 0$ if the system is feasible, and 2) $\sigma^T b > 0$ if the system is infeasible. This conclusion could be used to cut off solution $(\hat{y}, \hat{\alpha}, \hat{\beta}, \hat{\gamma}, \hat{\delta})$ if it causes a sub-problem infeasible. Specifically, we add the following constraint to the master problem when the sub-problem under scenario $\omega$ is infeasible:

$$\sum_{n=1}^{N} \sigma_{(n)}[(h^{\omega}_{(n)} - T^{\omega}_{(n,y)}y - T^{\omega}_{(n,\alpha)}\alpha - T^{\omega}_{(n,\beta)}\beta - T^{\omega}_{(n,\gamma)}\gamma - T^{\omega}_{(n,\delta)}\delta] \leq 0 \tag{2.64}$$

where $N$ is the total number of constraints of the sub-problem, $h^{\omega}_{(n)}$ is the right-hand-side of the constraint $n$ under scenario $\omega$, and $T^{\omega}$ are the coefficients of the first stage decision variables in scenario $\omega$, for example, $T^{\omega}_{(n,y)}$ is a vector of length $I$, representing the coefficient of $y$ in constraint $n$ under scenario $\omega$. According to the duality theorem, an infeasible linear programming has an unbounded dual problem, and the recession direction of the dual problem gives the value of vector $\sigma$. Most of the commercial linear programming solvers provide this information, for example, the "Cplex.DualFarkas" * method can be used to get the information of this vector.

*Step 5: Add optimality cut*

When $\mathbb{E}_{\omega}[\mathcal{Q}(y,\alpha,\beta,\gamma,\delta,\omega)]$ is a convex function, the following inequality holds:

$$\mathbb{E}_{\omega}[\mathcal{Q}(y,\alpha,\beta,\gamma,\delta,\omega)] \geq \mathbb{E}_{\omega}[\mathcal{Q}(\hat{y},\hat{\alpha},\hat{\beta},\hat{\gamma},\hat{\delta},\omega)] + \mu^T[(y,\alpha,\beta,\gamma,\delta) - (\hat{y},\hat{\alpha},\hat{\beta},\hat{\gamma},\hat{\delta})]$$

where $\mu$ is the sub-gradient of $\mathbb{E}_{\omega}[\mathcal{Q}(\hat{y},\hat{\alpha},\hat{\beta},\hat{\gamma},\hat{\delta},\omega)]$. By adding this constraint to the master problem, a better approximation of $\mathbb{E}_{\omega}[\mathcal{Q}(y,\alpha,\beta,\gamma,\delta,\omega)]$ is built. The sub-gradient $\mu$ can be constructed by the dual solution of the sub-problem. Specifically, let $\lambda^{\omega}$ be the dual solution of the sub-problem under scenario $\omega$, then $\mu = -\sum_{\omega\in\Omega} p^{\omega}\lambda^{\omega}T^{\omega}$. Let $\theta$ replace $\mathbb{E}_{\omega}[\mathcal{Q}(y,\alpha,\beta,\gamma,\delta,\omega)]$ in the above inequality, then newly added optimality cut is written as:

$$\begin{aligned}
\theta &\geq \mathbb{E}_{\omega}[\mathcal{Q}(\hat{y},\hat{\alpha},\hat{\beta},\hat{\gamma},\hat{\delta},\omega)] + \mu^T[(y,\alpha,\beta,\gamma,\delta) - (\hat{y},\hat{\alpha},\hat{\beta},\hat{\gamma},\hat{\delta})] \\
&= \mathbb{E}_{\omega}[\mathcal{Q}(\hat{y},\hat{\alpha},\hat{\beta},\hat{\gamma},\hat{\delta},\omega)] - \sum_{\omega\in\Omega} p^{\omega}\lambda^{\omega}T^{\omega}[(y,\alpha,\beta,\gamma,\delta) - (\hat{y},\hat{\alpha},\hat{\beta},\hat{\gamma},\hat{\delta})] \\
&= \sum_{\omega\in\Omega} p^{\omega}\lambda^{\omega}[h^{\omega} - T^{\omega}(\hat{y},\hat{\alpha},\hat{\beta},\hat{\gamma},\hat{\delta},\omega)] - \sum_{\omega\in\Omega} p^{\omega}\lambda^{\omega}T^{\omega}[(y,\alpha,\beta,\gamma,\delta) - (\hat{y},\hat{\alpha},\hat{\beta},\hat{\gamma},\hat{\delta})] \\
&= \sum_{\omega\in\Omega} p^{\omega}\lambda^{\omega}[h^{\omega} - T^{\omega}(y,\alpha,\beta,\gamma,\delta)] \\
&= \sum_{\omega\in\Omega} p^{\omega}\sum_{n=1}^{N}\lambda^{\omega}_{(n)}[(h^{\omega}_{(n)} - T^{\omega}_{(n,y)}y - T^{\omega}_{(n,\alpha)}\alpha - T^{\omega}_{(n,\beta)}\beta - T^{\omega}_{(n,\gamma)}\gamma - T^{\omega}_{(n,\delta)}\delta]
\end{aligned} \tag{2.65}$$

*Readers are referred to the blog of Dr. Paul Rubin titled "Infeasible LPs and Farkas Certificates"

**Identify the lower bound of $\mathbb{E}_{\omega}[\mathscr{Q}(y,\alpha,\beta,\gamma,\delta,\omega)]$**

We know that $\mathbb{E}_{\omega}[\mathscr{Q}(y,\alpha,\beta,\gamma,\delta,\omega)]$ represents the expected overtime cost of the ORs given a feasible first stage solution $(y,\alpha,\beta,\gamma,\delta)$. A valid lower bound of this expected recourse function would be 0. However, the solving procedure will be expedited if we can develop a valid and also tighter lower bound. Recall the Jensen's inequality that the expectation of a convex function of some argument is always greater than or equal to the function evaluated at the expected of its argument, i.e., $\mathbb{E}_{\omega}[g(\omega)] \geq g(\mathbb{E}_{\omega}[\omega])$, which could be used to develop a valid and tighter lower bound of $\mathbb{E}_{\omega}[\mathscr{Q}(y,\alpha,\beta,\gamma,\delta,\omega)]$.

**Proposition 1.** *For any feasible solution $(y,\alpha,\beta,\gamma,\delta)$, the recourse function $\mathscr{Q}(y,\alpha,\beta,\gamma,\delta,\omega)$ is a convex function of $\omega$*

*Proof.* First, we will prove the convexity a more general function $\mathscr{F}(\xi) = \min_{y\in Y}\{q^T y : Wy = \xi\}$. When the duality holds, we have $\mathscr{F}(\xi) = \max_{\xi\in\Xi}\{\xi^T z : W^T z \leq q\}$. It is known that the optimal solution to a linear programming model is one of the extreme points of the feasible region. Let $z_1, z_2, \ldots, z_K$ be the set of the extreme points of $W^T z \leq q$, then we have $\mathscr{F}(\xi) = \max_{k=1,\ldots,K}\{\xi^T z_k\}$, which is proved to be convex based on the following two facts : 1) the maximum of a set of convex function is also convex, and 2) $\xi^T z_k$ is a convex function for all $k$.

When the first stage solution $(y,\alpha,\beta,\gamma,\delta)$ makes the second stage feasible, then the duality assumption of $\mathscr{Q}(y,\alpha,\beta,\gamma,\delta,\omega)$ holds and $\mathscr{Q}(y,\alpha,\beta,\gamma,\delta,\omega) = \mathscr{F}(h^{\omega} - T^{\omega}(y,\alpha,\beta,\gamma,\delta))$. Note that $h^{\omega}$ and $T^{\omega}$ are linear expressions of $S^{\omega}$, $R^{\omega}$ and some constants, and they are the aberrations of $h(S^{\omega},R^{\omega})$ and $T(S^{\omega},R^{\omega})$, respectively.

$$
\begin{aligned}
& t\mathscr{Q}(y,\alpha,\beta,\gamma,\delta,\omega) + (1-t)\mathscr{Q}(y,\alpha,\beta,\gamma,\delta,\omega') \\
= \; & t\mathscr{F}\left(h^{\omega} - T^{\omega}(y,\alpha,\beta,\gamma,\delta)) + (1-t)\mathscr{F}(h^{\omega'} - T^{\omega'}(y,\alpha,\beta,\gamma,\delta)\right) \\
\geq \; & \mathscr{F}\left((th^{\omega} + (1-t)h^{\omega'}) - (tT^{\omega} + (1-t)T^{\omega'})(y,\alpha,\beta,\gamma,\delta)\right) \\
= \; & \mathscr{F}\left(h^{t\omega+(1-t)\omega'} - T^{t\omega+(1-t)\omega'}(y,\alpha,\beta,\gamma,\delta)\right) \\
= \; & \mathscr{Q}(y,\alpha,\beta,\gamma,\delta,t\omega + (1-t)\omega')
\end{aligned}
$$

Therefore $\mathscr{Q}(y,\alpha,\beta,\gamma,\delta,\omega)$ is a convex function of $\omega$. $\square$

**Proposition 2.** *Let $\bar{\omega}$ is the scenario that $(S_j^{\bar{\omega}}, R_j^{\bar{\omega}}) = (\mathbb{E}_\omega[S_j^\omega], \mathbb{E}_\omega[R_j^\omega])$, then $\theta_0 = \mathscr{Q}(y, \alpha, \beta, \gamma, \delta, \bar{\omega})$ is a valid lower bound of $\mathbb{E}_\omega[\mathscr{Q}(y, \alpha, \beta, \gamma, \delta, \omega)]$.*

*Proof.* According to Jensen's inequality, we have $\mathbb{E}_\omega[\mathscr{Q}(y, \alpha, \beta, \gamma, \delta, \omega)] \geq \mathscr{Q}(y, \alpha, \beta, \gamma, \delta, \mathbb{E}_\omega[\omega]) = \mathscr{Q}(y, \alpha, \beta, \gamma, \delta, \bar{\omega})$ □

### Relatively Complete Recourse Property

By solving the model (2.34) - (2.61) directly, one can always get a feasible solution. However, the L-shaped algorithm solves the master and recourse problem separately. According to the current formulation, the desirable relatively complete resource property does not hold, i.e., a feasible solution of the master problem may not be feasible in the second stage.

One type of infeasibility is caused by the completion time related constraints, including Constraints (2.45) - (2.46) and (2.58). For example, if an OR can accommodate all types of surgeries, then scheduling all patients' surgery operation and recovery to this OR is a feasible solution to the master problem; however, this solution will probably violate the constraints regarding OR maximum operating time and surgeons' availability.

Another infeasibility is related to the sequencing constraints on patients' recovery, including Constraint (2.54) and (2.55). For example, if the first stage decision $\delta_{jj'} = 1, \gamma_{jk} = 1$ and $\gamma_{j'k} = 1$, i.e., both patients $j$ and $j'$ are scheduled for recovery on bed $k$ and patient $j$ starts earlier. However, suppose that the surgery $j'$ finishes earlier than surgery $j$, so patient $j'$ has to stay in the OR for recovery. If, as shown in Figure 2.5, surgery $j$ has not finished, but patient $j'$ has completely recovered in the OR, i.e., $d_{j'k} = e_{j'k} < d_{jk}$, this leads the violations of Constraint (2.54) and (2.55).

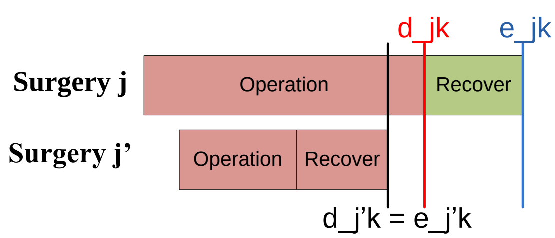


**Figure 2.5:** An example of infeasible case related to patients' sequence

As mentioned before, in each iteration, whenever the second stage problem is infeasible, the L-shaped algorithm will add an inequality constraint to the master problem, which ensures that this first stage solution will never happen again. However, this procedure is time consuming. Instead, one can modify the formulation of the sub-problem to achieve the relatively complete recourse (Birge and Louveaux, 2011; Batun, 2011).

In order to prevent the first type of infeasibility, i.e., the one causing by completion time related constraints, we introduce the three auxiliary variables, modify the objective function as well as three constraints, and add a new constraint. $U$ is a very large number, which defined as the penalty of violating the maximum OR operating time and the availability of surgeons; $v_{jl}^{\omega}$ is a non-negative variable, representing the amount of overtime surgeon $l$ need to spend to finish surgery $l$; $w_i^{\omega}$ is a non-negative variable, representing the amount of time OR $i$ exceeds the maximum operating time. The objective (2.43) has been modified as:

$$\mathscr{Q}(y,\alpha,\beta,\gamma,\delta,\omega) = \min \sum_{i=1}^{I} O_i z_i^{\omega} + U\left(\sum_{j=1}^{J}\sum_{l=1}^{L} v_{jl}^{\omega} + \sum_{i=1}^{I} w_i^{\omega}\right)$$

where the first term is the same as the original objective function, the second term represents the penalties for exceeding the maximum OR time and surgeon normal working hours. Accordingly, Constraints (2.45), (2.46), and (2.58) are modified as:

$$b_{ji}^{\omega} \leq M\alpha_{ji} \quad \forall j,i$$

$$c_{ji}^{\omega} \leq M\alpha_{ji} \quad \forall j,i$$

$$\sum_{i=1}^{I} b_{ji}^{\omega} + S_j^{\omega} - v_{jl}^{\omega} \leq Q_l\Theta_{jl} + (1-\Theta_{jl})M \quad \forall j,l$$

In addition, we add a new constraint on $z_i^{\omega}$ and $w_i^{\omega}$ as:

$$z_i^{\omega} - w_i^{\omega} \leq G - H \tag{2.66}$$

Regarding to the constraints about the sequence in the recovery area, we relax Constraints 2.67 as

$$\sum_{i=1}^{I} c_{ji}^{\omega} <= \sum_{k=1}^{K} d_{jk}^{\omega} \quad \forall j \tag{2.67}$$

and penalize the big value of $e_{jk}$ in the objective function. By doing this, the infeasibility situation shown in Figure 2.5 will be prevented. The advantages of achieving the relative complete recourse versus adding feasibility cut each iteration is compared in Section 2.5.4.

### Valid Optimality Cut

Besides the optimality cut (2.65), when the first stage variables are binary (our problem), we can derive another set of optimality cuts which are more efficient for branching based algorithms Laporte and Louveaux (1993).

$$\theta \geq (\mathbb{E}_{\omega}[\mathscr{Q}(\hat{y},\hat{\alpha},\hat{\beta},\hat{\gamma},\hat{\delta},\omega)] - \theta_0)(A - A' - |\mathscr{A}| + 1) + \theta_0 \tag{2.68}$$

where $\mathscr{A}$ is a subset of $(\hat{y},\hat{\alpha},\hat{\beta},\hat{\gamma},\hat{\delta})$ with nonzero value variables, for example, if $\hat{y}_i = 1$ then $\hat{y}_i \in \mathscr{A}$; the expressions of $A$ and $A'$ are formulated as follows: $A = \sum_{\hat{y}_i \in \mathscr{A}} y_i + \sum_{\hat{\alpha}_{ji} \in \mathscr{A}} \alpha_{ji} + \sum_{\hat{\beta}_{jj'} \in \mathscr{A}} \beta_{jj'} + \sum_{\hat{\gamma}_{jk} \in \mathscr{A}} \gamma_{jk} + \sum_{\hat{\delta}_{jj'} \in \mathscr{A}} \delta_{jj'}$ and $A' = \sum_{\hat{y}_i \notin \mathscr{A}} y_i + \sum_{\hat{\alpha}_{ji} \notin \mathscr{A}} \alpha_{ji} + \sum_{\hat{\beta}_{jj'} \notin \mathscr{A}} \beta_{jj'} + \sum_{\hat{\gamma}_{jk} \notin \mathscr{A}} \gamma_{jk} + \sum_{\hat{\delta}_{jj'} \notin \mathscr{A}} \delta_{jj'}$. The inequality $A - A' - |\mathscr{A}| \leq 0$ always holds, and it takes the value of 0 when the first stage solution is $(\hat{y},\hat{\alpha},\hat{\beta},\hat{\gamma},\hat{\delta},\omega)$. And in this case, the right-hand side of constraint (2.68) takes the value of $(\mathbb{E}_{\omega}[\mathscr{Q}(\hat{y},\hat{\alpha},\hat{\beta},\hat{\gamma},\hat{\delta},\omega)]$, otherwise, the value of the right-hand side is less than or equal to the lower bound $\theta_0$.

### Symmetry Constraints

In our problem, we assume that there is no difference between recovery beds. When a surgery ends, if more than one bed are available, then the patient can be sent to any of it. This symmetry solution structure can slow the progress of branching based algorithm for solving MIP. To deal with this symmetry, we add a term $\sum_{j=1}^{J}\sum_{k=1}^{K}\gamma_{jk}k$ to the objective function. By minimizing this term, it will ensure that if two recovery beds are available, then always assign it to the bed with a lower index.

## 2.5.4 Computational Results

In the first part of this section, we compare the different variants of the L-shaped algorithm. And the second part represents the value of stochastic solutions.

Note that the master problem is required to solve to optimality. Otherwise, the stochastic solution has no practical meaning. With respect to this fact, the instances tested in this section is not as large as the ones in Section 2.4.3. We randomly generate 32 instances with 2 different sizes. There are total 10 ORs and 2 recovery beds, 5 surgeries available for all the instances. The first set has 2 surgeons, and the other set has 3 surgeons available. For each size, we have 16 different instances. The details of the tested instances are listed in the tables of the next two sections.

The configuration related parameters listed at the beginning of Section 2.5.2 are either the same (including $F_i, O_i, H_i, G_i, M$) or generated in the same way (including $\Phi, P_l, Q_l, \Theta_{jl}$) as in Section 2.4.3. The penalty $U$ in the modified objective (2.43) is set to be 100000. While the scenario related parameters are generated as follows. For all the instances, there are total 100 different scenarios. For each scenario $\omega$, $S_j^\omega$ and $R_j^\omega$ are generated in the same way as in Section 2.4.3, i.e., depending on the type of the surgeries, $S_j^\omega$ is generated from one of the log-normal distributions with means equal to $72, 49, 103, 173, 135$ minutes and standard deviations equal to $23, 13, 25, 33, 32$ minutes, and $R_j^\omega$ is generated from one of the uniform distributions $\mathbb{U}(30,72)$, $\mathbb{U}(25,49)$, $\mathbb{U}(65,103)$, $\mathbb{U}(120,173)$, $\mathbb{U}(90,135)$. Note that when we calculate valid lower bound of the expected recourse function according to Proposition 2, $S_j^{\bar{\omega}}$ and $R_j^{\bar{\omega}}$ are equal to the mean of the 100 scenarios.

We coded the algorithms in Java and run it on a personal computer with Intel® Core™i7 2.20 GHz processor and 8.0 GB RAM. The CPLEX Optimizer is called to solve the master problem and sub-problems by using the ILOG concert technology. The time limit for all the algorithms are set to be 600 seconds and the maximum iteration is set to be 1000.

**Performance Comparison of L-shaped Algorithm Variants**

The computational performance of three different features of the L-shaped algorithm are tested in this section, including the optimality cuts, the feasibility cuts, and symmetry breaking. Three groups of experiments have been conducted and each group focuses on one feature.

As discussed previously, two sets of optimality cuts are valid for our problem: the optimality cut for general linear programming models shown in Eq. (2.65) (we call it "cut1"), and the special optimality cut for mixed integer programming models shown in Eq. (2.68) (we call it "cut2"). We have tested three different options for the optimality cuts: (1) cut1, (2) cut2, and (3) cut1 and cut2, and the objective value and computational time are reported in Table 2.5. Note that for all the three options, the other two features are the same: Modifying the original problem formulation to achieve the relatively complete recourse property and changing the objective function to break the symmetry. Overall, the objective values of the three options are almost identical, except for the instance 8 and 9 of 3 surgeons. For instance 8, the algorithms are not converged within the time limit for all three options, therefore, the best objective values are reported. The solution value of "Cut1" is slightly better than the other two options. For instance 9, "Cut1" cannot find a feasible solution for the original problem, while the other two options can find optimal solutions. By using "Cut1", most of the instances cannot converge within the time limit, while the other two options can find optimal solutions very quickly for most instances. In addition, "Cut2" takes a slightly less computational time than "Cut1&2". To sum up, "Cut1" performs the worst; "Cut2" and "Cut1&2" has compatible performance on finding optimal solutions, and "Cut2" is slightly faster.

Refer to our analysis of the relatively complete recourse property, there are two options to handle the infeasibility of the sub-problem, either adding a feasibility cut each time the problem is infeasible or reformulating the problem to achieve the relatively complete recourse property. The objective value and computational time of these two options are shown in Table 2.6. For the first set of the instances, the two options produces identical objective values, and the computational

**Table 2.5:** Computational performance of different optimality cut

| Ins | 5S-2S | | | | | | 5S-3S | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Cut1 | | Cut2 | | Cut1&2 | | Cut1 | | Cut2 | | Cut1&2 | |
| | Time | Obj | Time | Obj | Time | Obj | Time | Obj | Time | Obj | Time | Obj |
| 1 | 601 | 16000 | 5 | 16000 | 17 | 16000 | 600 | 16000 | 4 | 16000 | 2 | 16000 |
| 2 | 600 | 16000 | 4 | 16000 | 28 | 16000 | 2 | 8000 | 1 | 8000 | 1 | 8000 |
| 3 | 600 | 8933 | 600 | 8933 | 600 | 8933 | 600 | 16000 | 4 | 16000 | 14 | 16000 |
| 4 | 601 | 8918 | 47 | 8918 | 60 | 8918 | 2 | 8000 | 1 | 8000 | 1 | 8000 |
| 5 | 600 | 16000 | 3 | 16000 | 6 | 16000 | 601 | 8002 | 1 | 8002 | 1 | 8002 |
| 6 | 2 | 8000 | 1 | 8000 | 2 | 8000 | 600 | 16000 | 2 | 16000 | 3 | 16000 |
| 7 | 2 | 8000 | 1 | 8000 | 2 | 8000 | 601 | 8934 | 13 | 8934 | 26 | 8934 |
| 8 | 601 | 10988 | 2 | 10988 | 2 | 10988 | 602 | 16142 | 603 | 16166 | 603 | 16165 |
| 9 | 601 | 16000 | 16 | 16000 | 26 | 16000 | 600 | 2904874 | 124 | 16000 | 120 | 16000 |
| 10 | 600 | 16000 | 6 | 16000 | 2 | 16000 | 601 | 16000 | 25 | 16000 | 11 | 16000 |
| 11 | 601 | 16000 | 2 | 16000 | 2 | 16000 | 600 | 16000 | 13 | 16000 | 20 | 16000 |
| 12 | 603 | 16000 | 7 | 16000 | 11 | 16000 | 600 | 8003 | 1 | 8003 | 1 | 8003 |
| 13 | 601 | 16000 | 2 | 16000 | 4 | 16000 | 600 | 16000 | 2 | 16000 | 8 | 16000 |
| 14 | 600 | 16000 | 3 | 16000 | 3 | 16000 | 601 | 16000 | 2 | 16000 | 6 | 16000 |
| 15 | 600 | 16000 | 5 | 16000 | 3 | 16000 | 600 | 9933 | 49 | 9933 | 44 | 9933 |
| 16 | 600 | 16000 | 7 | 16000 | 15 | 16000 | 600 | 16000 | 3 | 16000 | 2 | 16000 |
| Mean | 526 | - | 44 | - | 49 | - | 526 | - | 53 | - | 54 | |

**Table 2.6:** Computational performance of different options to handle sub-problem infeasibility

| Ins | 5S-2S | | | | 5S-3S | | | |
|---|---|---|---|---|---|---|---|---|
| | Rel. Com. Rec. | | Add cut | | Rel. Com. Rec. | | Add cut | |
| | Time | Obj. | Time | Obj. | Time | Obj. | Time | Obj. |
| 1 | 5 | 16000 | 13 | 16000 | 4 | 16000 | 6 | 16000 |
| 2 | 4 | 16000 | 38 | 16000 | 1 | 8000 | 1 | 8000 |
| 3 | 600 | 8933 | 600 | 8933 | 4 | 16000 | 12 | 16000 |
| 4 | 47 | 8918 | 479 | 8918 | 1 | 8000 | 1 | 8000 |
| 5 | 3 | 16000 | 4 | 16000 | 1 | 8002 | 1 | 8002 |
| 6 | 1 | 8000 | 1 | 8000 | 2 | 16000 | 2 | 16000 |
| 7 | 1 | 8000 | 1 | 8000 | 13 | 8934 | 28 | 8934 |
| 8 | 2 | 10988 | 2 | 10988 | 603 | 16166 | 603 | 16156 |
| 9 | 16 | 16000 | 19 | 16000 | 124 | 16000 | 93 | - |
| 10 | 6 | 16000 | 9 | 16000 | 25 | 16000 | 6 | 16000 |
| 11 | 2 | 16000 | 2 | 16000 | 13 | 16000 | 9 | 16000 |
| 12 | 7 | 16000 | 6 | 16000 | 1 | 8003 | 1 | 8000 |
| 13 | 2 | 16000 | 2 | 16000 | 2 | 16000 | 19 | 16000 |
| 14 | 3 | 16000 | 22 | 16000 | 2 | 16000 | 11 | 16000 |
| 15 | 5 | 16000 | 4 | 16000 | 49 | 9933 | 44 | 9933 |
| 16 | 7 | 16000 | 8 | 16000 | 3 | 16000 | 2 | 16000 |
| Mean | 44 | - | 76 | - | 53 | - | 52 | - |

time of option "Add cut" is longer than option "Rel. Com. Rec.", especially for the instance 4. For the second set of the instances, option "Rel. Com. Rec." solves almost all the instances to optimality except instance 8. In contrast, besides instance 8, option "Add cut" fails to find feasible solutions for instance 9, and "-" means when the iteration limit (1000) is reached after 93 seconds, no feasible solution has been found by the algorithm. Note that for instance 12, the slight difference between the optimal objective values obtained by the two options is caused by the tolerance of the solver. The computational time of these two options on the second set of instances are compatible.

The no-difference assumption regarding the recovery beds may cause the symmetry problem and lead an unnecessary long computational time. The advantage of modifying the objective function of the sub-problem (we call it "Modify Obj.) is tested against the original objective function (we call it "No Change", and the results are reported in Table 2.7. For the first set of the instances, the two options produces identical objective values. By changing the objective function, the optimal solution of instance 14 can be found within the time limit. Besides, the computational time for optimal solutions has been reduced greatly, especially for instance 2 and 4. For the second set of the instances, the advantages of changing objectives are not as significant as the first set. On

**Table 2.7:** Computational performance of different options to handle Symmetry

| Ins | 5S-2S | | | | 5S-3S | | | |
|---|---|---|---|---|---|---|---|---|
| | Modify Obj | | No Change | | Modify Obj | | No Change | |
| | Time | Obj. | Time | Obj. | Time | Obj. | Time | Obj. |
| 1 | 5 | 16000 | 9 | 16000 | 4 | 16000 | 2 | 16000 |
| 2 | 4 | 16000 | 281 | 16000 | 1 | 8000 | 1 | 8000 |
| 3 | 600 | 8933 | 601 | 8933 | 4 | 16000 | 2 | 16000 |
| 4 | 47 | 8918 | 103 | 8918 | 1 | 8000 | 1 | 8000 |
| 5 | 3 | 16000 | 9 | 16000 | 1 | 8002 | 1 | 8002 |
| 6 | 1 | 8000 | 1 | 8000 | 2 | 16000 | 2 | 16000 |
| 7 | 1 | 8000 | 1 | 8000 | 13 | 8934 | 37 | 8934 |
| 8 | 2 | 10988 | 3 | 10988 | 603 | 16166 | 604 | 16128 |
| 9 | 16 | 16000 | 30 | 16000 | 124 | 16000 | 120 | 16000 |
| 10 | 6 | 16000 | 2 | 16000 | 25 | 16000 | 5 | 16000 |
| 11 | 2 | 16000 | 2 | 16000 | 13 | 16000 | 2 | 16000 |
| 12 | 7 | 16000 | 16 | 16000 | 1 | 8003 | 1 | 8000 |
| 13 | 2 | 16000 | 4 | 16000 | 2 | 16000 | 6 | 16000 |
| 14 | 3 | 16000 | 602 | 16000 | 2 | 16000 | 4 | 16000 |
| 15 | 5 | 16000 | 2 | 16000 | 49 | 9933 | 29 | 9933 |
| 16 | 7 | 16000 | 2 | 16000 | 3 | 16000 | 5 | 16000 |
| Mean | 44 | - | 104 | - | 53 | - | 51 | - |

average, the computational time is about 15 seconds difference, and the objective values reported by the two options are almost identical for all the instances.

Based on the above experiments, in general, using optimality cut2 only, achieving the relatively complete recourse property, modifying the objective function to breaking symmetry are better options from the perspectives of both the solution quality and computational time. Further, we compare the results based on Cplex solver with another commercial solver named Gurobi, and the results are shown in Table 2.8. In general, there's no much difference in the objective values obtained by the two solvers, and the Cplex solver computes faster on average.

**Table 2.8:** Computational performance of Cplex solver and Gurobi solver

| Ins | 5S-2S | | | | 5S-3S | | | |
|---|---|---|---|---|---|---|---|---|
| | Cplex | | Gurobi | | Cplex | | Gurobi | |
| | Time | Obj. | Time | Obj. | Time | Obj. | Time | Obj. |
| 1 | 5 | 16000 | 19 | 16000 | 4 | 16000 | 3 | 16000 |
| 2 | 4 | 16000 | 64 | 16000 | 1 | 8000 | 2 | 8000 |
| 3 | 600 | 8933 | 601 | 8933 | 4 | 16000 | 17 | 16000 |
| 4 | 47 | 8918 | 143 | 8918 | 1 | 8000 | 1 | 8000 |
| 5 | 3 | 16000 | 2 | 16000 | 1 | 8002 | 22 | 8000 |
| 6 | 1 | 8000 | 1 | 8000 | 2 | 16000 | 4 | 16000 |
| 7 | 1 | 8000 | 1 | 8000 | 13 | 8934 | 62 | 8934 |
| 8 | 2 | 10988 | 2 | 10988 | 603 | 16166 | 602 | 16164 |
| 9 | 16 | 16000 | 37 | 16000 | 124 | 16000 | 219 | 16000 |
| 10 | 6 | 16000 | 2 | 16000 | 25 | 16000 | 20 | 16000 |
| 11 | 2 | 16000 | 31 | 16000 | 13 | 16000 | 15 | 16000 |
| 12 | 7 | 16000 | 8 | 16000 | 1 | 8003 | 1 | 8000 |
| 13 | 2 | 16000 | 2 | 16000 | 2 | 16000 | 3 | 16000 |
| 14 | 3 | 16000 | 14 | 16000 | 2 | 16000 | 5 | 16000 |
| 15 | 5 | 16000 | 2 | 16000 | 49 | 9933 | 87 | 9933 |
| 16 | 7 | 16000 | 2 | 16000 | 3 | 16000 | 2 | 16000 |
| Mean | 44 | - | 58 | - | 53 | - | 67 | - |

### Value of the Stochastic Solution

For an instance, an Expected Value (EV) solution is defined as the solution of the deterministic problem when the random parameters are fixed to be equal to their average value. In our experiment, each instance has 100 different scenarios. For each scenario, we can calculate the value of the objective function by substitute the decision variables by the EV solution. The Expectation of the Expected Value problem (EEV) is defined as the expectation of the objective values over the 100 scenarios. Then the Value of the Stochastic Solution (VSS) is defined as the difference between the objective value of the Stochastic Programming model (SPV) and the EEV.

For each instance, the SPV, EEV, and VSS are reported in Table 2.9. Note that the SPV is the objective value obtained by the best performed algorithm. The EV solution is not always feasible for all scenarios. If for a scenario, one or more constraints are violated, a certain penalty is added to the objective value, and the unit penalty cost is set to be 100000. In other words, if the value of EEV is greater than 100000, then the EV solution is not valid for all scenarios of this instance. As shown in Table 2.9, 15 EV solutions are not valid for all scenarios. Implementing these EV solutions will cause the schedules of two surgeries overlap with each other or the maximum operating time of an

**Table 2.9:** Value of stochastic solution for each of the instance

| Ins | 5S-2S SPV | 5S-2S EEV | 5S-2S VSS | 5S-3S SPV | 5S-3S EEV | 5S-3S VSS |
|---|---|---|---|---|---|---|
| 1 | 16000 | 856000 | 840000 | 16000 | 16199 | 199 |
| 2 | 16000 | 16115 | 115 | 8000 | 8000 | 0 |
| 3 | 8933 | 9031 | 98 | 16000 | 16000 | 0 |
| 4 | 8918 | 9092 | 174 | 8000 | 8000 | 0 |
| 5 | 16000 | 997000 | 981000 | 8000 | 8000 | 0 |
| 6 | 8000 | 8000 | 0 | 16000 | 142112 | 126112 |
| 7 | 8000 | 8000 | 0 | 8934 | 9041 | 107 |
| 8 | 10988 | 11140 | 152 | 16128 | 566031 | 549903 |
| 9 | 16000 | 986000 | 970000 | 16000 | 2904874 | 2888874 |
| 10 | 16000 | 158102 | 142102 | 16000 | 16537 | 537 |
| 11 | 16000 | 1016000 | 1000000 | 16000 | 996000 | 980000 |
| 12 | 16000 | 1052087 | 1036087 | 8000 | 8000 | 0 |
| 13 | 16000 | 876000 | 860000 | 16000 | 396071 | 380071 |
| 14 | 16000 | 219253 | 203253 | 16000 | 266010 | 250010 |
| 15 | 16000 | 16113 | 113 | 9933 | 10162 | 230 |
| 16 | 16000 | 866000 | 850000 | 16000 | 16229 | 229 |

OR is exceeded. It requires lots of work of the OR manager to fix these problems. The solution of the stochastic programming model is quite robust, and it will not cause any violations in the surgery sequence or maximum operating time. For the rest of the 17 instances where the EV solutions do work, 10 of them have positive VSS, ranging from 98 (1.1%) to 537 (3.36%); All these analyses show that the solution of the stochastic programming model is never worse than the EV solution; and the advantages of the stochastic programming solution are more obvious for the instances with scenarios that are prone to be infeasible.

## 2.6 Summary

This chapter studies an elective surgery scheduling problem in a surgical center with multiple operations rooms and recovery beds. The problem is solved in one step, and three scheduling decisions are considered, including the number of ORs to open, surgery-OR assignment, and surgery sequencing. These decisions are subject to the availability of different resources, including

ORs, surgeons, recovery beds, surgery type. In addition, we have studied the impact of sequence-dependent setup time on the schedule decision. Both the deterministic and stochastic versions of this problem have been investigated.

For the deterministic problem, we have built an MIP model and a CP model and tested them on randomly generated instances. Although both the MIP and CP models fail to find the optimal solutions for most of the cases, the CP model is superior in the following aspects: (1) it has a smaller number of constraints and variables; (2) it is solved faster; (3) it yields better solutions, and (4) it provides Gantt chart for each scheduling solution. And along with the increase of the instance size, these advantages become more significant. Besides, the sensitivity analysis is conducted to examine the influence of the variations in surgery durations and recovery times on the implementation of the scheduling decisions obtained by the CP model. It is concluded that when the variations are small, i.e., the estimated surgery durations and recovery times are very close to the actual values, on average the hospital needs to pay 4.27% more than the total cost indicated by the deterministic solution versus 11.53% for large variations. Another issue about the deterministic solutions is that the maximum operating time constraints might be violated.

Instead of finding the solution that is the best with respect to the mean surgery durations and recovery times, we proposed a stochastic programming model to achieve the minimum expected cost. Specifically speaking, the stochastic problem relaxes the deterministic assumption about the surgery duration and recovery time; instead, it is assumed that their variation can be represented by a finite number of scenarios. The L-shaped algorithm is proposed to solve the two-stage stochastic mixed integer programing model. Different features have been tested by randomly generated instance, including, the optimality cut, the feasibility cut, and symmetry breaking. The stochastic solutions are always better than the EV solutions, and the advantages of the stochastic solutions are more significant when the EV solutions prone to be infeasible for some of the scenarios.

# Chapter 3

# Appointment Scheduling Problem in an Outpatient Clinic

## 3.1 Introduction

In outpatient clinics, the patients usually consult with their own primary care physician, and each physician's schedule can be treated independently. Therefore, we consider a single physician in this study. One session (usually an 8-hour shift) is divided into several equal-length slots. When patients call for making appointments for a certain day, they are either accepted and scheduled to a particular slot or rejected based on the current schedule of the physician, punctuality of the patients, and other factors. A certain percentage of the rejected appointments will be resubmitted for another day within the planning horizon by either the patient or the schedule manager. In general, appointment requests arrive individually and dynamically. The decision process at each call-in stage is illustrated in Figure 3.1. The medical resource is tight, and usually patients will accept any arrangement as long as their appointment requests are accepted. In this sense, patient's preference is not that urgent to model. Therefore, this study does not consider the preferences of the patients.

The major problem of scheduling appointments in outpatient clinics is the waiting time, which is also the major cause of dissatisfaction of the service. The waiting time of the patients

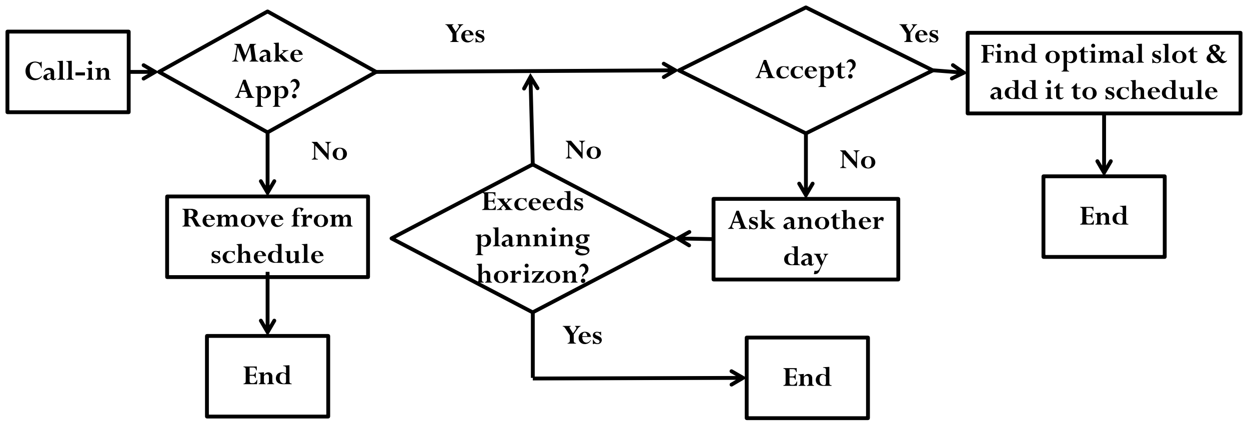


**Figure 3.1:** The decision process at each call-in stage of the appointment scheduling

includes two parts. One is direct waiting time, which is the time between the moment when the patient arrives at the clinic and the moment the service begins. The other is the indirect waiting time, which is the time between the day when the appointment request arrives and the day the appointment is scheduled. Since patients' physical conditions change over time, if the indirect waiting time is long patients might get recovered, see other doctors, or forget, which lead a significant rate of cancellations and no-shows. No-shows are patients who do not appear to their scheduled appointment, which waste the available capacity of valuable resources (physician, staff, and equipment), decrease the quality of care, increase costs, and limit clinic access to the patient population. In some clinics, up to 42% of scheduled patients fails to show up for prearranged appointments (Muthuraman and Lawley, 2008). Appointment delay, patient demographics and medical conditions, physician characteristics, as well as patient-physician interactions are the main factors causing no-shows. Although several methods are proposed, including sending telephone/mail reminders and patient education, to alleviate the impact of the no-shows, some of the clinics using the reminder system still report a high percentage of no shows (Hixon et al., 1999). Walk-in patients, cancellations, and no-shows disturb the well-arranged schedule.

Most outpatient clinics have developed scheduling systems to arrange patients' appointments according to clinic managers' concerns on system performance. The variety of criteria used to evaluate the appointment scheduling system can be divided into four main classes, including cost-based measures, time-based measures, congestion measures, fairness measures. Among them, the patients' waiting time, the physicians' overtime working, and the clinic's revenue are the most popular performance measures found in literature. Figure 3.2 presents a realization of a schedule of a certain day and illustrates several popular performance measures. There are four equal-length time slots in this session, and the notation $3(4)$ in Slot 1 means 4 patients arrive at Slot 1 and only 3 of them get served. As a result, one patient waits for one slot and overflows to Slot 2. In Slot 3, the time needed to serve the arrived 3 patients are shorter than the slot length, which leads idleness of the physician. In the last slot, 4 patients arrive, but only 3 of them get served, and the physician has to work overtime to serve the remaining 1 patients.

This work studies an on-line appointment scheduling problem in an outpatient clinic over a long planning horizon. The scheduling decisions are made upon receiving the calls, including whether to accept the appointment request, and if the request gets accepted, which day and slot to put this request. We are the first few who consider the cost of patients' indirect waiting time and model the probability of patient's cancellation rate as well as no-show rate as functions of their indirect waiting. An MDP model is developed and an easy-implemented scheduling policy is proposed.

The rest of this chapter is organized as follows. Section 3.2 summarizes the related work on appointment scheduling problems. Section 3.3 defines the scope of the problem, including the assumptions, objective, decisions, and methodologies. Section 3.4 presents the proposed MDP model and analyzes the properties of the optimal policies. Section 3.5 describes the backward induction algorithm for the optimal policy as well as a heuristic policy. Section 3.6 evaluates the optimal polices on randomly generated numerical instances and compared it with the heuristic

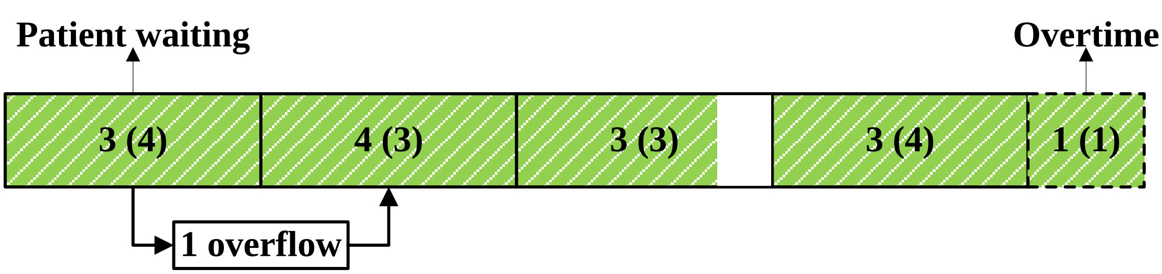


**Figure 3.2:** An illustration of patient waiting time and physician's overtime

policy and another commonly used scheduling rule. Section 3.7 concludes this chapter and summarizes the findings.

## 3.2 Literature Review

In the delivery of healthcare services, the uncertainty of patient arrival and service processes causes a huge waste of facility's resources and unnecessary patient waiting times Robinson and Chen (2009). The appointment scheduling of outpatient services has been widely studied with various tools and techniques since 1950s. Cayirli and Veral (2003) conducted a comprehensive survey of previous studies on the appointment scheduling in outpatient clinics. They discussed the general problem formulations and modeling considerations, including number of service/physicians, patients' arrival patterns, service processes, punctuality of physicians and their interruption, and queuing discipline. Further, they classified the performance measures into four major categories, and proposed general guidelines to design appointment systems. Gupta and Denton (2008) reviewed the literatures related to the appointment scheduling system of primary care, specialty care, and elective surgeries. They analyzed the factors that affect performance of appointment system and pointed out the open challenges to either improve the efficiency of the appointing systems or model the process more precisely.

The rest of this section reviews the related literatures from three aspects, including two aspects. Specifically, commonly used scheduling policies or scheduling rules are summarized in the first part of this review. And then the complicating factors as well as variations of problem settings are discussed in the second part of this review.

### 3.2.1 Appointment Scheduling Policies and Rules

The scheduling policy can be divided into two types. One is the sequential scheduling policy, and the other is the open-access scheduling policy. The sequential scheduling policy is widely used in the outpatient clinic, which is also the scheduling policy studied in this study. The sequential scheduling policy requires patients to call the clinic in advance for making appointments at a certain time of a certain day (Muthuraman and Lawley, 2008). Open access policy is introduced recently as an alternative to traditional sequential scheduling policy. It requires patients to come right away or call in the morning to make an appointment for the same day (Liu et al., 2010). This policy virtually eliminates no-shows; however, it introduces considerably imbalances in the day-to-day workload because the number of patients who want to be seen each day varies randomly. When the demand and supply are in balance, there are reported improvement in service quality by applying this policy. Some researchers combined these two policies by allocating a certain portion of the capacity for future appointments.

According to Cayirli and Veral (2003), the scheduling rules are combinations of the following three variables: the block-size, the begin-block, and the appointment interval. The clinic session is divided into several blocks. The appointment interval is the length of the block, constant or variable; the block-size is the number of patients assigned to each block, it could be one, other constant or variable; and the begin-block is the number of patients scheduled at the start of each session.

The single-block rule requires all patients to arrive at the beginning of each clinic session, and it was common in most clinics in the 1950s. Therefore, most of the early studies on the appointment scheduling problem are about this rule. Another prevalent rule is Bailey rule. In this rule, more than one patient is assigned to arrive at the beginning of the session, the block size is one, and the appointment interval is fixed. This rule minimizes the risk of the idle time of physicians if the first patient does not arrive punctually. Recently, researchers developed an individual-block/variable-interval rule, in which the patients are scheduled individually at varying appointment intervals. Some studies indicate that the optimal appointment intervals exhibit a ”dome pattern” when the service time is independent identically distributed and the waiting cost of the patients are uniformly distributed. Specifically, the dome pattern means that the intervals increase towards the middle of the session and then decrease.

Cayirli et al. (2009) employed simulation model to examine 18 different appointment scheduling rules under different patient arrival patterns. They pointed out that by adjusting interval length patients' waiting time, doctors' idle time and overtime could be improved. LaGanga and Lawrence (2007a) studied a single server system with deterministic service time and punctual arrivals. By compressing the appointment intervals, the excess appointment is accommodated. They used a simulation model to analyze the effects of the placement of the extra appointments in an overbooked appointment schedule. Kaandorp and Koole (2007) examined the optimal scheduling rule for a single server clinic with exponential distributed service time. The proposed optimal schedules are compared with the single-block rule and Bailey's rule. Wijewickrama and Takakuwa (2008) developed a simulation model to evaluate the appointment scheduling rules in a multi-facility system.

### 3.2.2 Complicating Factors

A common assumptions for studies on scheduling rules is all appointment requests are known at the beginning of the clinic session, and the service time is the single source of uncertainty. However, in reality, both the day and the time of the visits are determined upon receiving the calls. The demand uncertainty increases the complexity of the problem.

The complicating factors discussed in this subsection include the planning horizon, the number of physicians, service time distribution, patient characteristics, patient arrival process, no-shows and cancellations, and the walk-ins.

#### Planning Horizon

The planning horizon of the appointment scheduling in our study is a long time instead of one-day. In literature, only a few papers have addressed the appointment scheduling problem in a long term. Liu et al. (2010) developed a model for dynamic appointment scheduling decision over a planning horizon of $N$ days. However, the dynamic heuristic policies they proposed only specify on which day to schedule an appointment, but not to which slot of that day. Green and Savin (2008) studied the appointment system over a long period as a single server queuing system in which customers who are about to enter service have a state-dependent probability of not being served and may rejoin the queue. Their model aims to find a proper patient panel size. The performance measures used in this paper are expected patient backlog and the probability of getting a same-day appointment. Gupta and Wang (2008) modeled the patients' choice explicitly, and they divided the booking horizon (time between opening for reservation and the beginning of the workday) into $\tau$ discrete time periods such that there is at most one appointment request in each period. They developed an MDP model to determine which appointment to accept in order to maximize revenue. However, the unfavored patients' behaviors, including no-shows, cancellations, and walk-ins, are not considered in their study.

### Number of Physicians

A single server assumption is frequently used in previous studies (LaGanga and Lawrence, 2007a; Kaandorp and Koole, 2007; Liu et al., 2010; Green and Savin, 2008). It not only makes the problem simpler, but also reflects the reality at some degree. It is common that the doctors usually have their own lists of patients, especially in the primary health care clinic. On the one hand, it improves the service quality and reduces the service time since the doctor is familiar with the patient's situation. On the other hand, this practice improves the loyalty of the patient to the physician.

### Service Time

The service time distribution is another concern when modeling the appointment scheduling process. To simplify the model, some literatures assume deterministic service time (Kim and Giachetti, 2006; LaGanga and Lawrence, 2007a; Patrick et al., 2008; Robinson and Chen, 2009). Due to different situations of the patients, the service time should be stochastic. The empirical data indicates that the service time distribution is unimodal and right-skewed. Some studies used Erlang or exponential service times to keep their model tractable Cayirli and Veral (2003). The duration and the variety of the service time influence the system's performance, for example, the patients' waiting time and doctors' idle time. Therefore, the service time distribution should be carefully modeled.

### Patient Priority and Preference

Most of the previous study ignored the difference of the patients' situation, and few of them explicitly considered the patients priority when scheduling. Patrick et al. (2008) is the first one who proposed a method to dynamically schedule patients with different priorities to access the diagnostic facility. The priority of the patient is assigned based on medically acceptable waiting time. The scheduling process was modeled as an MDP, transformed into an equivalent linear program, and solved through approximate dynamic programming. They proposed a scheduling policy to guide the decisions as to when to book each priority class. Cayirli et al. (2009) relaxed the homogeneous patients assumption and treated patients as new and return when building appointment scheduling system.

Another aspect about patients that has been relatively overlooked is patient choice. Due to the medical resource constraints, most of the previous studies assumed that the patients will accept the arrangements. However, considering patients' preferences will improve their satisfaction about the service. The patient choices include the acuity of their need, time preference (which day and which time of a day), and loyalty to the doctors. Gupta and Wang (2008) is the first one (probably the only one) who modeled patient choice when scheduling appointments. When a call for the future appointment with a particular doctor arrived, based on the patient's and the doctor's identities, the appointment was either accepted by showing the patients all available time slots or declined. The workload of the physicians and the loyalty of the patients to their physicians were considered in their model. They pointed out that the positive dependence among the same-day demands and its variability deteriorated the system's performance, and patient loyalty benefited the system only when the workload between the physicians was balanced.

**Patient Arrival Process**

In most of the previous studies, the patients are assumed arriving punctually. However, most of the time, patients either arrive early (which cause congestion of the waiting room), late (which cause the idle time and/or overtime of the physician), or even don't show up (which wastes the resources). The scheduling system needs to make adjustments to reduce the disruptive effects caused by non-punctuality. Another common assumption is about the independence of the patients. There are a few studies on demand dependency. Abdus-Salaam et al. (2010) built a discrete-time, discrete space, stationary infinite-horizon MDP model to study the batch arrivals in a pediatric dental clinic with an open access policy. They concluded that companions need to be taken into consideration, especially when the majority of the patients are children or aged. Cayirli and Veral (2003) pointed out that the waiting room might get congestion because of the companions.

### No-shows and Cancellations

The no-shows and late cancellations waste the resources and decrease service quality, and it is necessary to take the no-shows into account when scheduling the patients in outpatient clinics. In some of the studies, all patients are treated as homogeneous and have the same no-show probabilities. For example, in the queue model of Green and Savin (2008), the impact of a constant no-show rate on the daily patient panel size was investigated. Similarly, in the simulation study of LaGanga and Lawrence (2007b), a fixed fraction of patients is treated as no-shows. In some of the studies, the no-show rates differentiated between patients. For example, Qu et al. (2007) considered both pre-scheduled appointment and open-access appointment. In their study, the no-show rates for the two kinds of appointment were different and both of them are constant. Muthuraman and Lawley (2008), Chakraborty et al. (2010), Zeng et al. (2010) estimated the no-show probabilities based on the patients' attributes. Specifically speaking, all the patients are divided into several groups or types based on some attribute analysis. And the patients of the same group/type are assumed to have the same no-show probability which is estimated from historical no-show data. Kim and Giachetti (2006) considered the no-show probability based on the current status of the system. Specifically, the probability of a patient show up for his/her appointment depends on the number of accepted appointments when the call made. In our problem, the no-show rates are modeled as a function of the appointment delay, and this method for modeling the no-show rates is also used in Liu et al. (2010).

### Walk-ins

Besides no-shows and cancellation, the unexpected walk-ins (either emergencies or regular patients), which fill in some of the empty slots caused by no-shows, make the situation more complicated. To handle the walk-in demands, extra capacities might be set-aside. The 'extra capacity' can be achieved via two ways (Cayirli and Veral, 2003). One method is during the appointment scheduling stage, instead of booking all the available capacity, several time slots do not accept appointments. The other method is to let the staff work overtime to provide health care service for the walk-in emergencies. Previous studies on the impact of walk-ins on the appointment scheduling can be found in Kim and Giachetti (2006) and Cayirli et al. (2009).

## 3.3 Problem Definition

The decision of this problem is the assignments of the appointment requests to the equal-length time slots in a session of a physician. The performance measure of the scheduling decision/policy is the net reward, i.e., the revenue serving patients minus the patient indirect and direct waiting time as well as physician's overtime. Note that the costs of the idleness of the physicians are not considered in this study since we believe those costs are included in the salaries of the physicians. All the cost functions are linear in this study.

The scheduling horizon of the clinic is $H$ days, i.e., the maximum indirect waiting time is $H$ days. The patients could call on day $t-1,\cdots,t-H$ to make an appointment for day $t$. We assumed that the calls are answered in the first-come-first-served matter. When a patient calls for making an appointment, he/she is either accepted and scheduled to a particular slot or rejected. If the appointment on day $t$ is rejected, the patient or the schedule manager will submit the appointment request for another day within the planning horizon. If no appropriate day is found to schedule this appointment to, then the patient will be asked to call in tomorrow or several days later. For a patient who has already been scheduled, he/she may call for canceling the appointment. The call-in period is partitioned into $N$ time intervals, and each interval is sufficiently small such that the schedule manager can handle no more than one call during any given interval. Note that the call may be either to make an appointment or to cancel an existing appointment.

Besides cancellation, another unexpected behavior of the patients is not showing up. The appointment cancellation and no-show rate is a function of the indirect waiting (the time between the day they call for the appointment and the day they are actually served by the physician).

If the patients show up, they are assumed to be punctual, i.e., arriving at the beginning of the slot they are scheduled to. Further, it is assumed that the physician must see all scheduled patients before they leave, so sometimes they may need to work overtime. The preferences of the patients on slots are not considered in this study. Beside the scheduled patients, we also consider walk-ins, which are unscheduled arrivals of patients and often require urgent care. Patients' appointment request and walk-ins are assumed independent of each other. Empirical studies showed that the service time follows unimodal and right-skewed distribution, and are usually modeled as Erlang or Exponential distribution (Cayirli and Veral, 2003). In this work, the service duration of each patient is independent from each other and assumed to follow an exponential distribution.

We build an MDP model to solve this problem and proposed a backward induction method to solve this MDP model to optimality. Figure 3.3 presents the definition of the appointment scheduling problem, including the objectives, decisions, assumptions, models and solving techniques.

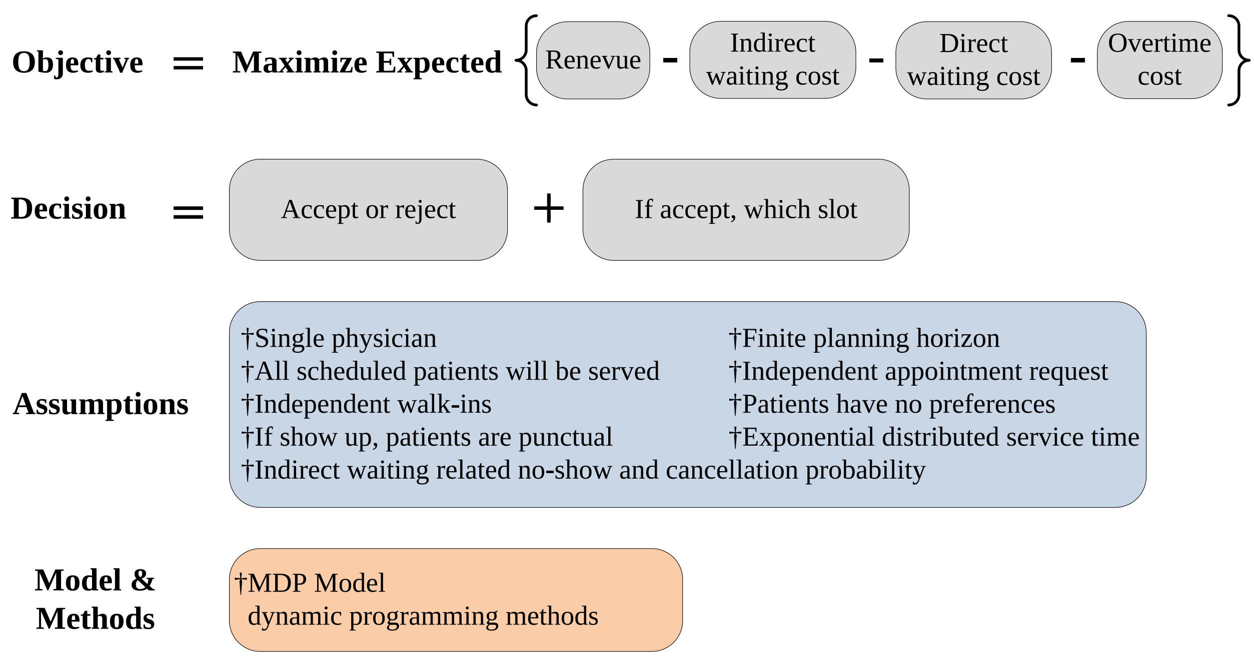


**Figure 3.3:** The definition of the elective surgery scheduling problem.

## 3.4 Markov Decision Process Model

This section presents the MDP model. We first give a brief introduction to general MDP models. After that, we develop the MDP model step by step for our appointment scheduling problem. And finally, we discuss the properties of the value functions and the form of the optimal policies.

### 3.4.1 Introduction

As generalizations from two stages to many stages, the MDP model provides a mathematical framework for modeling decision making process in situations where outcomes are not only determined by the decision maker but also random in nature. More precisely, an MDP is a stochastic control process. The basic assumption of the MDP is that the next state depends only on the previous state and action and the history does not count. At each decision stage, the status of the system is represented by a random variable $s$; the decision maker can choose any action $a$ from the action set $A$ in state $s$; with the probability $P_a(s,s')$ the action $a$ in state $s$ leads the system to state $s'$ and gives the decision maker a corresponding reward $R_a(s,s')$. Usually, an MDP can be represented by a 4-tuple $(S,A,P,R)$, where $S$ is a set of states, $A$ is a set of actions, $P$ is the set of transition probabilities, and $R$ is the immediate reward. The goal of solving an MDP model is to find a policy to maximize the future expected rewards. A policy specifies the action in each decision stage that the decision makers should take.

To find the best policy, we first define the value function for a policy, which quantifies the goodness of a policy. The value function is usually the expected total reward through the planning horizon, which is equal to the expected one step reward for the first action plus the expected reward for following the policy for the rest of the steps.

### 3.4.2 Model Development

Before presenting the MDP model, we first define the following notations.

**Notation of the Bellman's equation**

- $N$ The total number of call-in time intervals, also the total number of the stage in the MDP
- $n$ The index of the call-in time intervals, also the index of the stage in the MDP, $n = 1, 2, \cdots, N$
- $H$ The scheduling horizon
- $h$ The indirect waiting time of a patient, i.e., the days between the patient is scheduled and he/she is actually served by the physician, $h = 1, \cdots, H$.
- $k$ The type of patients, which depends on the indirect waiting time; type $k$ patient has indirect waiting of $h_k \sim h_{k+1} - 1$ days, where $h_1$ = 1and $h_{K+1} - 1 = H$
- $J$ The total number of slots in one day
- $j$ The index of the slot, $j = 1, 2, \cdots, J$
- $Y$ The state of the schedule, which is a $J \times K$ matrix; the element $Y_{jk}$ denoting the number of type $k$ patients scheduled to slot $j$
- $e_{jk}$ A matrix of size $J \times K$ with one at the $j,k$th position and zeros elsewhere
- $Y + e_{jk}$ The state of the schedule $Y$ is changed by adding the appointment of a type $k$ patient to slot $j$
- $Y - e_{jk}$ The state of the schedule $Y$ is changed by deleting the appointment of a type $k$ patient from slot $j$
- $\beta^n$ The probability that no calls at state $n$
- $\gamma_k^n$ The probability that an appointment request is received at stage $n$ from a type $k$ patient

- $\delta^n_{jk}(Y_{jk})$ The probability that a cancellation request is received at stage $n$ from a type $k$ patient who has been scheduled to slot $j$; it is a function of $Y_{jk}$. Note that if $Y_{jk} = 0$, then $\delta^n_{jk}(Y_{jk}) = 0$

- $\mathbb{V}^n(Y)$ The value function of state of the schedule at stage $n$

- $r$ The revenue earned by serving one patient

- $c_k$ The indirect waiting cost of type $k$ patient, $c_k < r$

**Notation for the value function of the final schedule**

- $\alpha_k$ The probability that a type $k$ patient shows up for his/her appointment

- $\pi_{jk}$ Discrete random variables, denoting the number of type $k$ patients showing up at the beginning of slot $j$, $\pi_{jk} \leq Y_{jk}$

- $X_j$ Discrete random variables, denoting the number of scheduled patients arriving at the beginning of slot $j$, $X_j = \sum_{k=1}^{K} \pi_{jk}$

- $Z_j$ Discrete random variables, denoting the number of walk-ins in each slot

- $U_j$ Discrete random variables, denoting the number of patients arrived in slot $j$, including pre-scheduled patients and walk-ins, $U_j = X_j + Z_j$

- $\mu$ The mean service rate of the physician

- $V$ Discrete random variables, denoting the number of patients served in each slot given an infinite number of patients, random variable following Poisson distribution with mean equal to $\mu$

- $W_j$ Discrete random variables, denoting the number of patients waiting for completion of service at the end of slot $j$

- $O$ Continuous random variables, denoting the overtime of the physician

- $c'$ The waiting cost per slot per patient

- $c''$ The overtime cost per slot per physician

According to the discretization of the time intervals, i.e., each interval is sufficiently small such that the scheduler will not receive more than one calls during any given interval, each time interval is a stage in the MDP. There are total $N$ stages in this model, and stage 1 is considered as the opening of the physician's schedule for reservation and stage $N$ is the beginning of the schedule execution. In each stage $n < N$, we assume exactly one of the following events will happen: (1) an arrival of appointment request from a type $k$ patient, $k = 1, \cdots, K$, (2) a cancellation request from a patient who has been scheduled, and (3) no calls. This assumption is satisfied naturally when the appointment requests and cancellations arrive according to a Poisson process (Subramanian et al., 1999). Under the assumptions above, the equation $\sum_{k=1}^{K} \gamma_k^n + \sum_{k=1}^{K} \sum_{j=1}^{J} \delta_{jk}^n(Y_{jk}) + \beta^n = 1$ holds in every stage.

It is assumed that the decision is made upon receiving the call and the state changes within each stage. When a type $k$ patient calls for making appointments, the possible actions are either scheduling the patient to a slot $j$ ($\forall j = 1, \cdots, J$), i.e., $Y \leftarrow Y + e_{jk}$, or rejecting the request (transferring the patient request to another day), i.e., $Y$ stays the same; when a cancellation request from a type $k$ patient who has already scheduled to slot $j$ arrives at a stage, the only possible action is to delete the patient from the schedule, i.e., $Y \leftarrow Y - e_{jk}$; when no calls, the schedule state $Y$ remains the same. Note that for cancellations and no calls, no decision needs to make since there is only one action that can be taken in both cases. The state transition diagram at stage $n$ is shown in Figure 3.4.

Whenever an appointment request from type $k$ patient is accepted, the net reward earned by the clinic is the revenue seeing one patient minus the indirect waiting cost of this patient, i.e., $r - c_k$; Similarly, whenever a cancellation occurs, the clinic loses the net reward of accepting this patient, i.e, $-r + c_k$. In this study, we treat all patients equally and don't differentiate the revenues generated by each of them. This assumption could be relaxed in future research work.

Many dynamic programming methods are used to solve the MDP model, which breaks the decision problem into smaller sub-problems. The optimal policy can be constituted based on the Bellman's Principle of Optimality: whatever the initial state is, remaining decisions must be

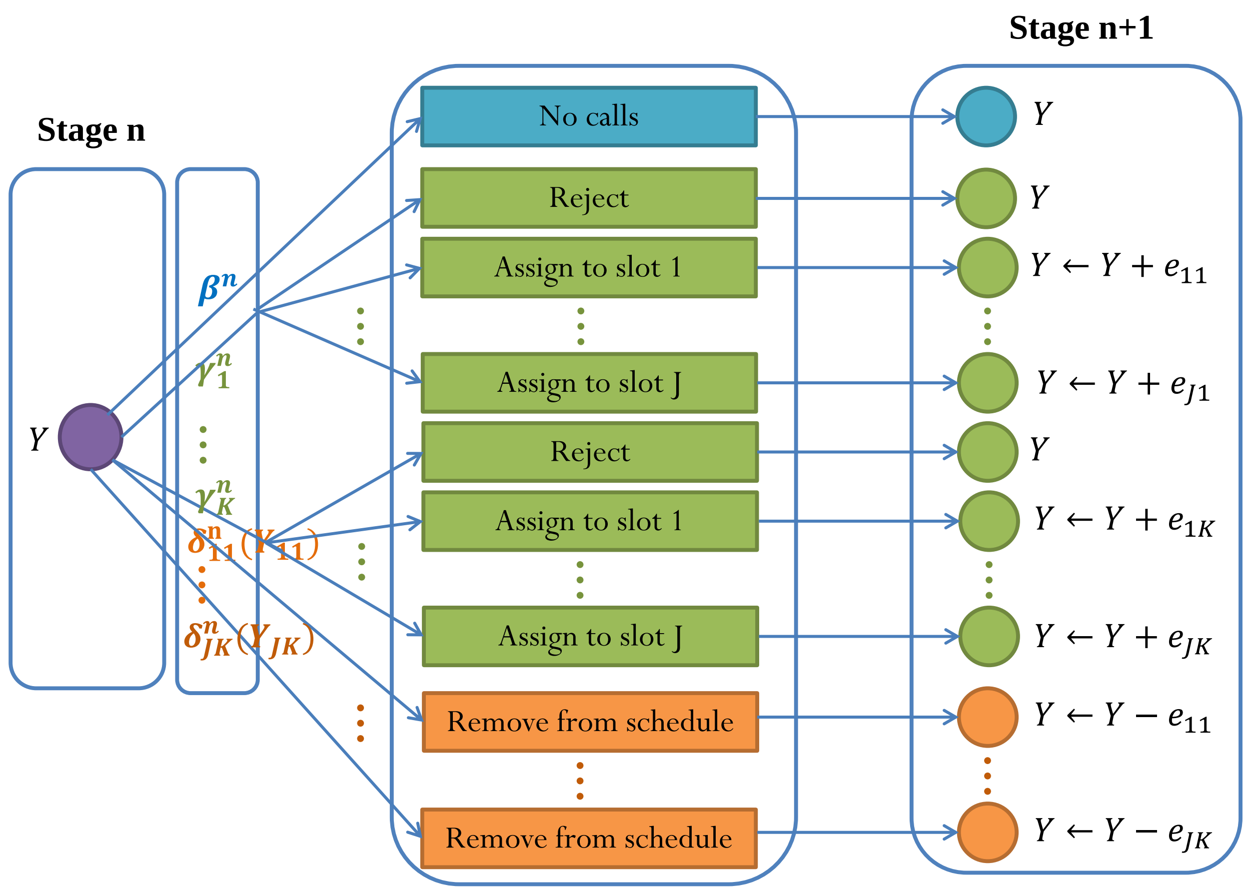


**Figure 3.4:** State transition diagram at stage $n$.

optimal with regard the state following from the first decision. According to the Bellman Equation, which is derived from the Principle of Optimality, all the values of possible consequent states at stage $n+1$ are used to provide an expected value for the current state $Y$ at stage $n$ ($n < N$):

$$\begin{aligned}\mathbb{V}^n(Y) = \; & \sum_{k=1}^{K} \gamma_k^n \left( \max\left\{ \mathbb{V}^{n+1}(Y), \max_{j=1,\dots,J} \{ r - c_k + \mathbb{V}^{n+1}(Y + e_{jk}) \} \right\} \right) \\ & + \sum_{k=1}^{K} \sum_{j=1}^{J} \delta_{jk}^n (Y_{jk}) \left( -r + c_k + \mathbb{V}^{n+1}(Y - e_{jk}) \right) + \beta^n \mathbb{V}^{n+1}(Y)\end{aligned} \tag{3.1}$$

where the first part of the equation is the expected value when an appointment request arrives at stage $n$, the second part is the expected value when a cancellation occurs, and the third part is corresponding to the situation of no calls. At stage $N$, the schedule is finalized, and value function of $\mathbb{V}^N(Y)$ is evaluated as the expected net reward of the final schedule $Y$, which equals the revenue gained by the walk-ins minus the loss of revenue due to no-shows, the cost of the patient direct waiting time as well as the physician's overtime. Before presenting the formulation of $\mathbb{V}^N(Y)$, we first explore the probability distribution as well as the expect value of 1) number of patients arriving in each slot, 2) number of patients waiting at the beginning of each slot, and 3) the overtime of the physician.

Not all of the schedule patients will show up at their scheduled time. We assume all the patients are independent of each other, and each of them has a no-show probability depending on their indirect waiting time. Let $\pi_{jk}$ denote the number of type $k$ patients showing up at the beginning of slot $j$, where $\pi_{jk} \le Y_{jk}$ for $j = 1, 2, \cdots, J$ and $k = 1, 2, \cdots, K$. Since the patient arrivals are independent of each other, $\pi_{jk}$ follows binomial distribution $B(Y_{jk}, \alpha_k)$, i.e.,

$$\Pr\{\pi_{jk} = m\} = b(Y_{jk}, m, \alpha_k) = \binom{Y_{jk}}{m} \alpha_k^m (1 - \alpha_k)^{Y_{jk} - m} \tag{3.2}$$

The number of scheduled patients showing up at the beginning of time slot $j$ is the sum of all types of patients who have shown up, i.e., $X_j = \sum_{k=1}^{K} \pi_{jk}$. For $x = 0, 1, 2, \cdots, \sum_{k=1}^{K} Y_{jk}$, the probability distribution of $X_j$ is:

$$\begin{aligned} P_{X_j}(x) &= \Pr\{X_j = x\} \\ &= \Pr\{\textstyle\sum_{k=1}^{K} \pi_{jk} = x\} \\ &= \sum_{m_{jk} \le Y_{jk}, \sum_{k=1}^{K} m_{jk} = x} \prod_{k=1}^{K} \Pr\{\pi_{jk} = m_{jk}\} \end{aligned} \tag{3.3}$$

The patients arrived in slot $j$ ($\forall j = 1, \cdots, J$) consists the pre-scheduled ones and walk-ins, i.e., $U_j = X_j + Z_j$. For $u = 0, 1, 2, \cdots$, the probability distribution of $U_j$ is:

$$\begin{aligned} P_{U_j}(u) &= \Pr\{U_j = u\} \\ &= \Pr\{X_j + Z_j = u\} \\ &= \sum_{x=0}^{u} \Pr\{X_j = x\}\Pr\{Z_j = u - x\} \\ &= \sum_{x=0}^{u} P_{X_j}(x)P_{Z_j}(u - x) \end{aligned} \tag{3.4}$$

The service time of each patient follows an exponential distribution with the mean equal to $1/\mu$. The number of patients waiting for completion of service at the end of slot $j$ ($\forall j = 1, \cdots, J$), $W_j$, is related to the number of patients overflowed from slot $j-1$, $W_{j-1}$, the number of patients arrived at slot $j$, $U_j$, and the number of patients served in slot $j$, $V$. If $V < W_{j-1} + U_j$, $W_j = W_{j-1} + U_j - V$, otherwise 0. Therefore, for $w = 1, 2, \cdots$, the probability distribution of $W_j$ is:

$$\begin{aligned} P_{W_j}(w) &= \Pr\{W_j = w\} \\ &= \Pr\{W_{j-1} + U_j - V = w\} \\ &= \Pr\{V = W_{j-1} + U_j - w\} \\ &= \sum_{v=w}^{\infty} \Pr\{V = v - w\}\Pr\{W_{j-1} + U_j = v\} \\ &= \sum_{v=w}^{\infty} P_V(v - w) \sum_{u=0}^{v} P_{W_{j-1}}(v - u)P_{U_j}(u) \end{aligned} \tag{3.5}$$

Note that we assume that there is no patient waiting at the beginning of the day, i.e., $p_{W_0}(0) = 1$. The probability of $W_j$ equal to 0 is:

$$\begin{aligned} P_{W_j}(0) &= \Pr\{W_j = 0\} \\ &= \Pr\{W_{j-1} + U_j - V \leq 0\} \\ &= \Pr\{V \geq W_{j-1} + U_j\} \\ &= \sum_{v=0}^{\infty} \Pr\{V \geq v\}\Pr\{W_{j-1} + U_j = v\} \\ &= \sum_{v=0}^{\infty} (1 - \sum_{i=0}^{v-1} P_V(i)) \sum_{u=0}^{v} P_{W_{j-1}}(v - u)P_{U_j}(u) \end{aligned} \tag{3.6}$$

The expected number of patients waiting at the end of the time slot $j$ can be calculated as:

$$E(W_j) = \sum_{w=1}^{\infty} wP_{W_j}(w) \tag{3.7}$$

The physician's overtime is the time needed to serve the patients who are still waiting at the end of the last time slot $J$, i.e., $W_J$. Since the service time of one patient follows an exponential distribution with mean equal to $1/\mu$, so the service time of $w$ patients follows an Erlang distribution with shape parameter equal to $w$ and scale parameter equal to $1/\mu$. So for $t > 0$, the probability of the physician overtime smaller than or equal to $t$ is:

$$F_O(t) = \sum_{w=1}^{\infty} P_{W_J}(w)\text{Erlang}(t,w,\mu) = \sum_{w=1}^{\infty} P_{W_J}(w)(1 - \sum_{i=0}^{w-1} \frac{e^{-(t\mu)}(t\mu)^i}{(i)!}). \tag{3.8}$$

By taking the first derivative of the Cumulative Density Function$F_O(t)$, we got the Probability Density Function of $O$ as:

$$f_O(t) = \mathrm{d}F_O(t) = \sum_{w=1}^{\infty} P_{W_J}(w)\frac{\mu^w t^{w-1} e^{-t\mu}}{(w-1)!} \tag{3.9}$$

The expected overtime of the physician $E(O)$ can be calculated by definition as:

$$\begin{aligned} E(O) &= \int_{t=0}^{\infty} t f_O(t) \\ &= \int_{t=0}^{\infty} t \sum_{w=1}^{\infty} P_{W_J}(w)\frac{\mu^w t^{w-1} e^{-t\mu}}{(w-1)!} \\ &= \sum_{w=1}^{\infty} P_{W_J}(w)\frac{w}{\mu} \\ &= \frac{E(W_J)}{\mu} \end{aligned} \tag{3.10}$$

Based on the above analysis, the expected net reward of schedule $Y$ at the final stage $N$ can be written as:

$$\begin{aligned} \mathbb{V}^N(Y) &= E\left[r\sum_{j=1}^{J} Z_j - \sum_{j=1}^{J}\sum_{k=1}^{K}(r-c_k)(Y_{jk}-\pi_{jk}) - c'\sum_{j=1}^{J} W_j - c''O\right] \\ &= r\sum_{j=1}^{J} E(Z_j) - \sum_{j=1}^{J}\sum_{k=1}^{K}(r-c_k)Y_{jk}(1-\alpha_k) - E\left[c'\sum_{j=1}^{J} W_j + c''O\right] \\ &= r\sum_{j=1}^{J} E(Z_j) - \sum_{j=1}^{J}\sum_{k=1}^{K}(r-c_k)Y_{jk}(1-\alpha_k) - c'\sum_{j=1}^{J} E(W_j) - \frac{c''E(W_J)}{\mu} \end{aligned} \tag{3.11}$$

As shown in Eq. (3.11), the net reward consists four parts: the revenue of walk-ins, the revenue loss caused by no-shows, the direct waiting cost of patients, and the overtime cost of the physician.

Similar as Subramanian et al. (1999), we define $\mathbb{H}^n(Y)$* be the total expected loss of revenue over periods $n$ to $N$ caused by cancellations and no-shows, and $\mathbb{H}^n(Y)$ is given by the following

*$\mathbb{H}^n(Y)$ can be interpreted as "the negative of the value function of the policy that rejects all arrivals"(Subramanian et al., 1999)

recursive equations:

$$\mathbb{H}^n(Y) = \sum_{k=1}^{K} \gamma_k^n \mathbb{H}^{n+1}(Y) + \sum_{k=1}^{K}\sum_{j=1}^{J} \delta_{jk}^n(Y_{jk})\left(r - c_k + \mathbb{H}^{n+1}(Y - e_{jk})\right) + \beta^n \mathbb{H}^{n+1}(Y) \tag{3.12}$$

$$\mathbb{H}^N(Y) = \sum_{j=1}^{J}\sum_{k=1}^{K} (r - c_k) Y_{jk}(1 - \alpha_k) \tag{3.13}$$

Further, we define $\mathbb{U}^n(Y)^{\dagger} = \mathbb{V}^n(Y) + \mathbb{H}^n(Y)$, then based on Eq. (3.1) and Eq. (3.11- 3.13), the following recursive equations are established:

$$\begin{aligned} \mathbb{U}^n(Y) &= \textstyle\sum_{k=1}^{K} \gamma_k^n \left( \max\left\{ \mathbb{U}^{n+1}(Y), \max_{j=1,\ldots,J} \{ r - c_k - \mathbb{G}_{jk}^n + \mathbb{U}^{n+1}(Y + e_{jk}) \} \right\} \right) \\ &+ \textstyle\sum_{k=1}^{K}\sum_{j=1}^{J} \delta_{jk}^n(Y_{jk}) \mathbb{U}^{n+1}(Y - e_{jk}) + \beta^n \mathbb{U}^{n+1}(Y) \end{aligned} \tag{3.14}$$

$$\mathbb{U}^N(Y) = r\sum_{j=1}^{J} E(Z_j) - c' \sum_{j=1}^{J} E(W_j) - \frac{c'' E(W_J)}{\mu} \tag{3.15}$$

where $\mathbb{G}_{jk}^n = \mathbb{H}^{n+1}(Y + e_{jk}) - \mathbb{H}^{n+1}(Y)$. The Eq. (3.14 - 3.15) generate the same optimal booking policy as the original value function (3.1) and (3.11), and thus they are equivalent in this sense.

In our problem, we assume that $\delta_{jk}^n$ is a function of $Y_{jk}$ and independent of reservations of other slot by other types of patients. Under this assumption, the cancellations and no-shows of different type of patients in different slots are independent of each other. Therefore, $\mathbb{H}^n(Y)$ can be written as $\sum_{j=1}^{J}\sum_{k=1}^{K} \mathbb{H}_{jk}^n(Y_{jk})$ and the following recursive equations about $\mathbb{H}_{jk}^n(Y_{jk})$ are established:

$$\mathbb{H}_{jk}^n(Y_{jk}) = \left(1 - \delta_{jk}^n(Y_{jk})\right) \mathbb{H}_{jk}^{n+1}(Y_{jk}) + \delta_{jk}^n(Y_{jk}) \left(r - c_k + \mathbb{H}_{jk}^{n+1}(Y - e_{jk})\right) \tag{3.16}$$

$$\mathbb{H}_{jk}^N(Y_{jk}) = (r - c_k) Y_{jk}(1 - \alpha_k) \tag{3.17}$$

Furthermore, from the above two equations, it follows that:

$$\begin{aligned} \mathbb{G}_{jk}^n &= \mathbb{H}^{n+1}(Y + e_{jk}) - \mathbb{H}^{n+1}(Y) = \mathbb{H}_{jk}^{n+1}(Y_{jk} + 1) - \mathbb{H}_{jk}^{n+1}(Y_{jk}) \\ &= (\delta_{jk}^{n+1}(Y_{jk} + 1) - \delta_{jk}^{n+1}(Y_{jk}))(r - c_k) + (1 - \delta_{jk}^{n+1}(Y_{jk} + 1))\mathbb{G}_{jk}^{n+1}(Y_{jk} + 1) \\ &+ \delta_{jk}^{n+1}(Y_{jk})\mathbb{G}_{jk}^{n+1}(Y_{jk}) \end{aligned} \tag{3.18}$$

$$\mathbb{G}_{jk}^{N-1} = \mathbb{H}^N(Y + e_{jk}) - \mathbb{H}^N(Y) = (r - c_k)(1 - \alpha_k) \tag{3.19}$$

If the cancellation rate $\delta_{jk}^n$ is proportional to $Y_{jk}$ and independent of slot, i.e., $\delta_{jk}^n(Y_{jk}) = \delta_k^n Y_{jk}$, then $\mathbb{G}_{jk}^n = \mathbb{G}_k^n$, which only depends on the type of patients and the decision stage, and the following

[†] $\mathbb{U}^n(Y)$ can be interpreted as "the maximal expected controllable net benefit of operating the system over period $n$ to $N$" (Subramanian et al., 1999)

recursive equations are satisfied:

$$\mathbb{G}_k^n = \delta_k^{n+1}(r - c_k) + (1 - \delta_k^{n+1})\mathbb{G}_k^{n+1} \tag{3.20}$$

$$\mathbb{G}_k^{N-1} = (r - c_k)(1 - \alpha_k) \tag{3.21}$$

### 3.4.3 Properties of Optimal Policy

In this section, we analyze the properties of the value function and the form of the optimal policy.

**Proposition 3.** $\mathbb{U}^N(Y)$ *is a non-increasing concave function of* $Y$

*Proof.* Since $E(Z_j)$ is a constant and independent of $Y$, $\mathbb{U}^N(Y)$ is a linear combination of $E(W_j)$ with negative coefficients. To prove that $\mathbb{U}^N(Y)$ is a non-increasing concave function of $Y$, it is equivalent to prove $E(W_j)$ is a non-decreasing convex function of $Y$.

Define $\Pi$ as a $J \times K$ matrix to represent the number of scheduled patients who show up for their scheduled appointment, $\pi_j$ be the sum of the row $j$, representing the total number of patients showing up at slot $j$, and $\pi_{jk}$ be the $(j,k)^{th}$ element of this matrix representing the number of type $k$ patients showing up at slot $j$. Now we will show that $E(W_j|\Pi)$ is a non-decreasing convex function of $\Pi$. According to Eq. (3.5), we can easily get the conditional probability of $P_{W_j|M}(w)$ as:

$$\begin{aligned} P_{W_j|\Pi}(w) &= \Pr\{W_j = w|\Pi\} \\ &= \Pr\{W_{j-1} + Z_j + \pi_j - V = w\} \\ &= \Pr\{V = W_{j-1} + Z_j + \pi_j - w\} \\ &= \begin{cases} \displaystyle\sum_{v=w-\pi_j}^{\infty} P_V(v + \pi_j - w) \sum_{u=0}^{v} P_{W_{j-1}}(v-u) P_{Z_j}(u) & \text{if} w \geq \pi_j \\ \displaystyle\sum_{v=0}^{\infty} P_V(v + \pi_j - w) \sum_{u=0}^{v} P_{W_{j-1}}(v-u) P_{Z_j}(u) & \text{if} w < \pi_j \end{cases} \end{aligned} \tag{3.22}$$

Then the conditional expectation $E(W_j|\Pi)$ can be calculated as:

$$
\begin{aligned}
E(W_j|\Pi) &= \sum_{w=1}^{\infty} w P_{W_j|\Pi}(w) \\
&= \sum_{w=1}^{\pi_j-1}\left(w\sum_{v=0}^{\infty}P_V(v+\pi_j-w)\sum_{u=0}^{v}P_{W_{j-1}}(v-u)P_{Z_j}(u)\right) \\
&+ \sum_{w=\pi_j}^{\infty}\left(w\sum_{v=w-\pi_j}^{\infty}P_V(v+\pi_j-w)\sum_{u=0}^{v}P_{W_{j-1}}(v-u)P_{Z_j}(u)\right) \\
&= \sum_{w=1}^{\pi_j-1}\left(w\sum_{v=0}^{\infty}P_V(v+\pi_j-w)\sum_{u=0}^{v}P_{W_{j-1}}(v-u)P_{Z_j}(u)\right) \\
&+ \sum_{w=0}^{\infty}\left((w+\pi_j)\sum_{v=w}^{\infty}P_V(v-w)\sum_{u=0}^{v}P_{W_{j-1}}(v-u)P_{Z_j}(u)\right) \\
&= \sum_{w=1}^{\pi_j-1}\left(w\sum_{v=0}^{\infty}P_V(v+\pi_j-w)\mathbb{A}\right)+\sum_{w=0}^{\infty}w\mathbb{B}+\sum_{w=0}^{\infty}\pi_j\mathbb{B}
\end{aligned}
\tag{3.23}
$$

where $\mathbb{A}=\sum_{u=0}^{v}P_{W_{j-1}}(v-u)P_{Z_j}(u)$ and $\mathbb{B}=\sum_{v=w}^{\infty}P_V(v-w)\sum_{u=0}^{v}P_{W_{j-1}}(v-u)P_{Z_j}(u)$. Let $\mathbb{C}(\pi_j)=E(W_j|\Pi)$, since Eq. (3.23) indicates that $E(W_j|\Pi)$ is a function of $\pi_j$. Next we will prove that $\mathbb{C}(\pi_j)$ is a non-decreasing convex function of $\pi_j$.

Case 1: $\pi_j \geq 2$

$$
\begin{aligned}
& \mathbb{C}(\pi_j+1)-\mathbb{C}(\pi_j) \\
= & \sum_{w=1}^{\pi_j} w\sum_{v=0}^{\infty}P_V(v+\pi_j+1-w)\mathbb{A}-\sum_{w=1}^{\pi_j-1}w\sum_{v=0}^{\infty}P_V(v+\pi_j-w)\mathbb{A}+\sum_{w=0}^{\infty}\mathbb{B} \\
= & \sum_{w=0}^{\pi_j-1}(w+1)\sum_{v=0}^{\infty}P_V(v+\pi_j-w)\mathbb{A}-\sum_{w=1}^{\pi_j-1}w\sum_{v=0}^{\infty}P_V(v+\pi_j-w)\mathbb{A}+\sum_{w=0}^{\infty}\mathbb{B} \\
= & \sum_{w=0}^{\pi_j-1}\sum_{v=0}^{\infty}P_V(v+\pi_j-w)\mathbb{A}+\sum_{w=0}^{\infty}\mathbb{B}\geq 0
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{C}(\pi_j+2)-\mathbb{C}(\pi_j+1) &= \sum_{w=0}^{\pi_j}\sum_{v=0}^{\infty}P_V(v+\pi_j+1-w)\mathbb{A}+\sum_{w=0}^{\infty}\mathbb{B} \\
&= \sum_{w=-1}^{\pi_j-1}\sum_{v=0}^{\infty}P_V(v+\pi_j-w)\mathbb{A}+\sum_{w=0}^{\infty}\mathbb{B}\geq 0
\end{aligned}
$$

$$
\mathbb{C}(\pi_j+2)-2\mathbb{C}(\pi_j+1)+\mathbb{C}(\pi_j)=\sum_{v=0}^{\infty}P_V(v+\pi_j+1)\mathbb{A}+\sum_{w=0}^{\infty}\mathbb{B}\geq 0
$$

Case 2: $\pi_j = 1$

$$
\mathbb{C}(\pi_j+1)-\mathbb{C}(\pi_j)=\sum_{v=0}^{\infty}P_V(v+1)\mathbb{A}+\sum_{w=0}^{\infty}\mathbb{B}\geq 0
$$

$$\mathbb{C}(\pi_j+2)-\mathbb{C}(\pi_j+1)=\sum_{v=0}^{\infty}P_V(v+2)\mathbb{A}+\sum_{v=0}^{\infty}P_V(v+1)\mathbb{A}+\sum_{w=0}^{\infty}\mathbb{B}\geq 0$$

$$\mathbb{C}(\pi_j+2)-2\mathbb{C}(\pi_j+1)+\mathbb{C}(\pi_j)=\sum_{v=0}^{\infty}P_V(v+2)\mathbb{A}\geq 0$$

Case 3: $\pi_j=0$

$$\mathbb{C}(\pi_j+1)-\mathbb{C}(\pi_j)=\sum_{w=0}^{\infty}\mathbb{B}\geq 0$$
$$\mathbb{C}(\pi_j+2)-\mathbb{C}(\pi_j+1)=\sum_{v=0}^{\infty}P_V(v+1)\mathbb{A}+\sum_{w=0}^{\infty}\mathbb{B}\geq 0$$

$$\mathbb{C}(\pi_j+2)-2\mathbb{C}(\pi_j+1)+\mathbb{C}(\pi_j)=\sum_{v=0}^{\infty}P_V(v+1)\mathbb{A}\geq 0$$

To sum up the above three cases, $\mathbb{C}(\pi_j+1)-\mathbb{C}(\pi_j)\geq 0$ and $\mathbb{C}(\pi_j+2)-2\mathbb{C}(\pi_j+1)+\mathbb{C}(\pi_j)\geq 0$ for all $\pi_j$. Therefore, we have proved that $C(\pi_j)$ is a non-decreasing convex function of $\pi_j$. Since $\pi_j=\sum_{k=1}^{K}\pi_{jk}$, we have

$$\begin{aligned}
& \mathbb{C}\left(\lambda(\pi_{j1},\cdots,\pi_{jK})+(1-\lambda)(\pi'_{j1},\cdots,\pi'_{jK})\right)\\
= & \mathbb{C}\left(\lambda\pi_j+(1-\lambda)\pi'_j\right)\\
\leq & \lambda\mathbb{C}(\pi_j)+(1-\lambda)\mathbb{C}(\pi'_j)\\
= & \lambda\mathbb{C}(\pi_{j1},\cdots,\pi_{jK})+(1-\lambda)\mathbb{C}(\pi_{j1},\cdots,\pi_{jK})
\end{aligned}$$

Therefore, function $\mathbb{C}$, i.e., $E(W_j|\Pi)$, is a convex function of $\pi_{jk}$.

Since $\pi_{jk}$ is a random variable from binomial distribution $B(Y_{jk},\alpha_k)$, according to the Lemma 1 in Subramanian et al. (1999) ‡, $E(W_j)=E[E(W_j|\Pi)]$ is a non-decreasing convex function of $Y$. And thus we have proved that $\mathbb{U}^N(Y)=r\sum_{j=1}^{J}E(Z_j)-c'\sum_{j=1}^{J}E(W_j)-\dfrac{c''E(W_J)}{\mu}$ is a non-increasing concave function of $Y$. □

**Proposition 4.** $\mathbb{D}^n(Y)=\max\left\{\mathbb{U}^{n+1}(Y),\ \max_{j=1,\ldots,J}\{r_k^n+\mathbb{U}^{n+1}(Y+e_{jk})\}\right\}$ *is a non-increasing concave function if* $\mathbb{U}^{n+1}(Y)$ *is a non-increasing concave function.*

*Proof.* $\mathbb{D}^n(Y)$ is a non-increasing concave function is equivalent to $\mathbb{D}^n(Y)-\mathbb{D}^n(Y+e_{j'k})\geq 0$ and $\mathbb{D}^n(Y)-2\mathbb{D}^n(Y+e_{j'k})+\mathbb{D}^n(Y+2e_{j'k})\leq 0$ for all $j'k$.

‡Let $f(y)$ $(y\geq 0)$ be a nondecreasing convex function. For each non-negative integer $x$, let $Y(x)$ be a binomial-$(x,\gamma)$ random variable $(0<\gamma<1)$ and let $h(x):=E[f(Y(x))]$. Then $h(x)$ is nondecreasing convex in $x\in\{0,1,\cdots\}$

Since $\mathbb{U}^{n+1}(Y)$ is non-increasing, we have $\mathbb{U}^{n+1}(Y) \geq \mathbb{U}^{n+1}(Y+e_{j'k})$, $\mathbb{U}^{n+1}(Y+e_{jk}) \geq \mathbb{U}^{n+1}(Y+e_{j'k}+e_{jk})$ for $j=1,\cdots,J$. Therefore, $\max\left\{\mathbb{U}^{n+1}(Y), \max_{j=1,\ldots,J}\{r_k^n+\mathbb{U}^{n+1}(Y+e_{jk})\}\right\} \geq \max\left\{\mathbb{U}^{n+1}(Y+e_{j'k}), \max_{j=1,\ldots,J}\{r_k^n+\mathbb{U}^{n+1}(Y+e_{j'k}+e_{jk})\}\right\}$, i.e., $\mathbb{D}^n(Y) \geq \mathbb{D}^n(Y+e_{j'k})$. And thus we have proved that $\mathbb{D}^n(Y)$ is a non-increasing function. Next, we will prove that $\mathbb{D}^n(Y)$ is a concave function.

Case 1: $\mathbb{D}^n(Y) = \mathbb{U}^{n+1}(Y)$

Since $\mathbb{U}^{n+1}(Y)$ is a non-increasing concave function, we have $\mathbb{U}^{n+1}(Y+e_{j'k}) - \mathbb{U}^{n+1}(Y) \geq \mathbb{U}^{n+1}(Y+e_{j'k}+e_{jk}) - \mathbb{U}^{n+1}(Y+e_{jk})$ for $j=1,\ldots,J$. In addition, from $\mathbb{D}^n(Y) = \mathbb{U}^{n+1}(Y)$, we can conclude that $\mathbb{U}^{n+1}(Y) \geq r_k^n + \mathbb{U}^{n+1}(Y+e_{jk})$ for $j=1,\ldots,J$. Add the above two equations together, we get $\mathbb{U}^{n+1}(Y+e_{j'k}) \geq \mathbb{U}^{n+1}(Y+e_{j'k}+e_{jk})$, i.e., $\mathbb{D}^n(Y+e_{j'k}) = \mathbb{U}^{n+1}(Y+e_{j'k})$. Similarly, we have $\mathbb{D}^n(Y+2e_{j'k}) = \mathbb{U}^{n+1}(Y+2e_{j'k})$. Therefore, the following relationship is satisfied:

$$
\begin{aligned}
& \mathbb{D}^n(Y) - 2\mathbb{D}^n(Y+e_{j'k}) + \mathbb{D}^n(Y+2e_{j'k}) \\
= \quad & \mathbb{U}^{n+1}(Y) - 2\mathbb{U}^{n+1}(Y+e_{j'k}) + \mathbb{U}^{n+1}(Y+2e_{j'k}) \leq 0
\end{aligned}
$$

Case 2: $\mathbb{D}^n(Y) = r_k^n + \mathbb{U}^{n+1}(Y+e_{j_1k}), \mathbb{D}^n(Y+e_{j'k}) = \mathbb{U}^{n+1}(Y+e_{j'k})$

Based on the analysis in Case 1, equation $\mathbb{D}^n(Y+2e_{j'k}) = \mathbb{U}^{n+1}(Y+2e_{j'k})$ is satisfied. In addition, according to the definition of $\mathbb{D}^n(Y+e_{j'k})$, we have $\mathbb{U}^{n+1}(Y+e_{j'k}) \geq r_k^n + \mathbb{U}^{n+1}(Y+e_{j'k}+e_{j_1k})$. Therefore, the following relationship is satisfied:

$$
\begin{aligned}
& \mathbb{D}^n(Y) - 2\mathbb{D}^n(Y+e_{j'k}) + \mathbb{D}^n(Y+2e_{j'k}) \\
= \quad & r_k^n + \mathbb{U}^{n+1}(Y+e_{j_1k}) - 2\mathbb{U}^{n+1}(Y+e_{j'k}) + \mathbb{U}^{n+1}(Y+2e_{j'k}) \\
\leq \quad & r_k^n + \mathbb{U}^{n+1}(Y+e_{j_1k}) - \mathbb{U}^{n+1}(Y+e_{j'k}) - r_k^n - \mathbb{U}^{n+1}(Y+e_{j'k}+e_{j_1k}) + \mathbb{U}^{n+1}(Y+2e_{j'k}) \\
= \quad & \mathbb{U}^{n+1}(Y+e_{j_1k}) - \mathbb{U}^{n+1}(Y+e_{j'k}) - \mathbb{U}^{n+1}(Y+e_{j'k}+e_{j_1k}) + \mathbb{U}^{n+1}(Y+2e_{j'k}) \\
\leq \quad & 0
\end{aligned}
$$

Case 3: $\mathbb{D}^n(Y) = r_k^n + \mathbb{U}^{n+1}(Y+e_{j_1k}), \mathbb{D}^n(Y+e_{j'k}) = r_k^n + \mathbb{U}^{n+1}(Y+e_{j'k}+e_{j_2k}), \mathbb{D}^n(Y+2e_{j'k}) = \mathbb{U}^{n+1}(Y+2e_{j'k})$

According to the definition of $\mathbb{D}^n(Y+e_{j'k})$, we have $\mathbb{U}^{n+1}(Y+e_{j'k}+e_{j_2k}) \geq \mathbb{U}^{n+1}(Y+e_{j'k}+e_{j_1k})$ and $r_k^n + \mathbb{U}^{n+1}(Y+e_{j'k}+e_{j_2k}) \geq \mathbb{U}^{n+1}(Y+e_{j'k})$. Therefore, the following relationship is satisfied:

$$
\begin{aligned}
&\mathbb{D}^n(Y) - 2\mathbb{D}^n(Y+e_{j'k}) + \mathbb{D}^n(Y+2e_{j'k}) \\
= \;& r_k^n + \mathbb{U}^{n+1}(Y+e_{j_1k}) - 2(r_k^n + \mathbb{U}^{n+1}(Y+e_{j'k}+e_{j_2k})) + \mathbb{U}^{n+1}(Y+2e_{j'k}) \\
= \;& \mathbb{U}^{n+1}(Y+e_{j_1k}) - \mathbb{U}^{n+1}(Y+e_{j'k}+e_{j_2k}) - (r_k^n + \mathbb{U}^{n+1}(Y+e_{j'k}+e_{j_2k})) + \mathbb{U}^{n+1}(Y+2e_{j'k}) \\
\leq \;& \mathbb{U}^{n+1}(Y+e_{j_1k}) - \mathbb{U}^{n+1}(Y+e_{j'k}+e_{j_1k}) - \mathbb{U}^{n+1}(Y+e_{j'k}) + \mathbb{U}^{n+1}(Y+2e_{j'k}) \\
\leq \;& 0
\end{aligned}
$$

Case 4: $\mathbb{D}^n(Y) = r_k^n + \mathbb{U}^{n+1}(Y+e_{j_1k})$, $\mathbb{D}^n(Y+e_{j'k}) = r_k^n + \mathbb{U}^{n+1}(Y+e_{j'k}+e_{j_2k})$, $\mathbb{D}^n(Y+2e_{j'k}) = r_k^n + \mathbb{U}^{n+1}(Y+2e_{j'k}+e_{j_3k})$

According to the definition of $\mathbb{D}^n(Y+e_{j'k})$, we have $\mathbb{U}^{n+1}(Y+e_{j'k}+e_{j_2k}) \geq \mathbb{U}^{n+1}(Y+e_{j'k}+e_{j_1k})$ and $\mathbb{U}^{n+1}(Y+e_{j'k}+e_{j_2k}) \geq \mathbb{U}^{n+1}(Y+e_{j'k}+e_{j_3k})$. In addition, based on the conclusion of Case 2, we have $\mathbb{U}^{n+1}(Y+e_{j_1k}) - \mathbb{U}^{n+1}(Y+e_{j'k}+e_{j_1k}) \leq \mathbb{U}^{n+1}(Y+e_{j'k}) - \mathbb{U}^{n+1}(Y+2e_{j'k})$ Therefore, the following relationship is satisfied:

$$
\begin{aligned}
&\mathbb{D}^n(Y) - 2\mathbb{D}^n(Y+e_{j'k}) + \mathbb{D}^n(Y+2e_{j'k}) \\
= \;& \mathbb{U}^{n+1}(Y+e_{j_1k}) - \mathbb{U}^{n+1}(Y+e_{j'k}+e_{j_2k}) - \mathbb{U}^{n+1}(Y+e_{j'k}+e_{j_2k}) + \mathbb{U}^{n+1}(Y+2e_{j'k}+e_{j_3k}) \\
\leq \;& \mathbb{U}^{n+1}(Y+e_{j_1k}) - \mathbb{U}^{n+1}(Y+e_{j'k}+e_{j_1k}) - \mathbb{U}^{n+1}(Y+e_{j'k}+e_{j_3k}) + \mathbb{U}^{n+1}(Y+2e_{j'k}+e_{j_3k}) \\
\leq \;& \mathbb{U}^{n+1}(Y+e_{j'k}) - \mathbb{U}^{n+1}(Y+2e_{j'k}) - \mathbb{U}^{n+1}(Y+e_{j'k}+e_{j_3k}) + \mathbb{U}^{n+1}(Y+2e_{j'k}+e_{j_3k}) \\
= \;& \mathbb{U}^{n+1}(Y') - \mathbb{U}^{n+1}(Y'+e_{j'k}) - \mathbb{U}^{n+1}(Y'+e_{j_3k}) + \mathbb{U}^{n+1}(Y'+e_{j'k}+e_{j_3k}) \\
\leq \;& 0
\end{aligned}
$$

where $Y' = (Y+e_{j'k})$.

To sum up, we have proved that $\mathbb{D}^n(Y) - \mathbb{D}^n(Y+e_{j'k}) \geq 0$ and $\mathbb{D}^n(Y) - 2\mathbb{D}^n(Y+e_{j'k}) + \mathbb{D}^n(Y+2e_{j'k}) \leq 0$, i.e., $\mathbb{D}^n(Y)$ is a non-increasing concave function.

□

**Proposition 5.** *For $1 \leq n \leq N$, $U^n(Y)$ is a non-increasing concave function of $Y$.*

*Proof.* We will prove it by induction on $n$.

Step 1: Hold for $N$

$\mathbb{U}^N(Y)$ is a non-increasing concave function of $Y$ based on Proposition 3.

Step 2: Hold for $n$ if hold for $n+1$, for $1 \leq n \leq N-1$

$\mathbb{U}^n(Y) = \sum_{k=1}^{K} \gamma_k^n \mathbb{D}^n(Y) + \sum_{k=1}^{K} \sum_{j=1}^{J} \delta_{jk}^n(Y_{jk}) \mathbb{U}^{n+1}(Y-e_{jk}) + \beta^n \mathbb{U}^{n+1}$, where $\mathbb{D}^n(Y)$ is defined in

Proposition 4 and proved to be a non-increasing concave function. As a linear combination of a set of non-increasing function with positive coefficients, $\mathbb{U}^n(Y)$ is also a non-increasing concave function.

$\square$

It has been proved by (Lippman and Stidham Jr, 1977) that when the value function is non-increasing and concave the optimal policy is monotonic in the state. In the context of our problem, the optimal policy is a book limit policy.

## 3.5 Solution Methods

In this section, we first present the backward induction method for solving the MDP and obtaining the optimal policy. And in the second part, we proposed an alternative heuristic policy.

### 3.5.1 Backward Induction Methods

The state of a stage $Y$, which is a $J \times K$ matrix. The element $Y_{jk}$ represents the number of type $k$ patients scheduled to slot $j$ ($\forall j = 1, \cdots, J$). We define $M$ as the largest number of type $k$ patients could be scheduled to one slot, which is reasonable since the optimal policy is a book-limit policy. The state space, i.e., total number of possible schedules, is $(M+1)^{JK}$, which grows rapidly along with the increase of $J$ and $K$ and could be intractable when these values are too large.

For a small size problem, the state space is tractable and could be enumerated. All the possible schedules could be the final schedule, and we could evaluate the value function of each possible final schedule based on Equation (3.11). The backward induction method is used to solve the Equation (3.1) from the final stage to the first and create the decision tables for each stage. Specifically, based on the value function of stage $n+1$, Equation (3.1) is used to generate the value function of stage $n$ for $n < N$ as well as the optimal action $a^*$ to take. As mentioned previously, three events will possibly happen in each stage: call for making appointments, call for cancellations, and no calls. When a type $k$ patient calls for making an appointment, the optimal action $a^*(Y,n,k)$ can be found as follows: (1) if $\text{Max}_{j \in \{1,2,\cdots,J\}}\{r - c_k - \mathbb{G}^n_{jk} + \mathbb{U}^{n+1}(Y + e_{jk})\} \geq$

$\mathbb{U}^{n+1}(Y)$, then assign this patient to time slot $j = arg\mathrm{M}ax_{j\in\{1,2,\cdots,J\}}\{r - c_k - \mathbb{G}^n_{jk} + \mathbb{U}^{n+1}(Y + e_{jk})\}$; (2) otherwise, transfer this patient to another day, i.e., reject it. When a patient calls for cancellation or there is no call for appointment, the optimal policy is relatively straightforward since there is only one possible action for each of the two cases.

The optimal scheduling policy, i.e., the decision tables at each stage, is calculated in advance. Therefore, whenever a patient calls for making an appointment, based on the current schedule and patient type, the schedule manager retrieves these decision tables to find the optimal action: either scheduling the patient to the time slot which results a new schedule with the greatest value function or transferring the patient to another day.

### 3.5.2 Alternative Heuristic Policy

As we mentioned when deriving Eq. (3.20), the cancellation rate $\delta^n_{jk}$ is proportional to $Y_{jk}$ and independent of slot $j$, therefore, $\delta^n_k$ can be treated as the cancellation rate of a type $k$ patient at time period $n$. Following that, if the current stage is $n_0$, the probability that a type $k$ patient will never cancel his/her appointment is calculated as $\Pi^{N-1}_{n=n_0}(1-\delta^n_k)$. Given that the appointment is not canceled, the probability of a type $k$ patient show up is $\alpha_k$. When we say a patient actually presents for his/her appointment, we mean the patient never calls for cancellation and shows up on time. Depending the stage we are currently at, the probability a type $k$ patient actually presents for the appointment is predicted differently. For example, if we are at the stage $n_0$, this probability is calculated as $\alpha_k\Pi^{N-1}_{n=n_0}(1-\delta^n_k)$.

Based on the above analysis, give a schedule $Y$ at stage $n_0$, we can estimate the expected number of patients who will show up in each slot. Taking slot $j$ as an example, the expected number would be $\sum^K_{k=1} Y_{jk}(\alpha_k\Pi^{N-1}_{n=n_0}(1-\delta^n_k)) + E(Z_j)$, denoted by $\overline{Y}_j$. The proposed heuristic policy is described as follows: when a type $k$ patient calls for making an appointment, the slot which satisfies the condition of $Y_{jk} \leq M$ is added to the candidate set $\mathbb{C}$; the action would be: (1) if set $\mathbb{C}$ is not empty, then assign this patient to slot $j = arg\mathrm{M}in_{j\in\mathbb{C}}\{\overline{Y}_j\}$; (2) if $\mathbb{C}$ is empty, then reject this appointment.

## 3.6 Numerical Examples

In this section, we first test the backward induction method on a randomly generated instance, report the optimal action taken in each decision period, and show the evolution of the value function. And next, we compare the performance of the heuristic policy with the optimal policy.

### 3.6.1 Optimal Policy by Backward Induction

As we analyzed before, the state space grows exponentially along with the increase of the total patient type $K$ and total slots $J$, which makes the backward induction method only applicable for relatively small instances. In this section, we have used the backward induction method to solve an instance with total 4 time slots and 2 type of patients. We set $M$=2 in this instance, and the total number of possible schedule is $3^8 = 6,561$.

The reward of serving one patient $R$ is set to be greater than the unit direct waiting cost, but less than the unit overtime cost. We set the revenue seeing one patient $r = 100$, the indirect waiting cost of type 1 patient $c_1 = 10$, the indirect waiting cost of type 2 patient $c_2 = 10$, the direct waiting cost per slot per patient $c' = 30$, and the overtime cost per slot per physician $c'' = 240$.

The patients with shorter appointment delay have higher show-up probability. We set the show-up probabilities $\alpha$ for type 1 and type 2 patients as 0.8 and 0.5, respectively. During each time slot, at most one walk-in could happen. We set the probabilities of exactly one walk-in in the four time slots are set as $0.2, 0.6, 0.5, 0.3$, respectively. The service time follows the exponential distribution, and the mean service rate per slot $\mu = 2$.

In this instance, there are total 15 call-in periods, i.e., 15 decision stages. At each decision stage, one of the three types of events will happen, including making appointments, canceling appointments, no calls. The value of the parameters $\gamma_k^n$ and $\delta_k^n$ are shown in Table 3.1. Note that the probability of a type $k$ patient cancel his/her appointment of slot $j$ not only depends on $\delta_k^n$ but also on the state of the system, and is calculated as $\delta_k^n Y_{jk}$. The probability of no calls at stage $n$ is calculated as $1 - \gamma_k^n - \sum_{j=1}^{J} \delta_k^n Y_{jk}$.

**Table 3.1:** State-dependent event probability

| Parameter | Period n | | | | |
|---|---|---|---|---|---|
| | 1-5 | 6-9 | 10-12 | 13-14 | 15 |
| $\gamma_1^n$ | 0 | 0.4 | 0.6 | 0.7 | 0.8 |
| $\gamma_2^n$ | 0.4 | 0 | 0 | 0 | 0 |
| $\delta_1^n$ | 0 | 0.06 | 0.04 | 0.03 | 0.02 |
| $\delta_2^n$ | 0.1 | 0.07 | 0.05 | 0.04 | 0.03 |

Based on the above information, the decision table for each decision stage and system state is calculated in advance. For this instance, it takes about 421 seconds to run, and the calculation of the final $\mathbb{U}^N$ takes about 415 seconds.

According to the definition of $\mathbb{V}^n(Y)$, it consists of four parts, revenue of walk-ins, revenue loss of no-shows, directing waiting cost, and overtime cost. To get the total net reward of a schedule $Y$ at time period $n$, we need to include the revenue of seeing scheduled patients as well as indirect waiting costs on top of $\mathbb{V}^n(Y)$. Therefore, we define one more function $\mathbb{W}^n(Y)$ to represent the net reward of $Y$ at stage $n$, and it calculated as $\mathbb{W}^n(Y) = \mathbb{V}^n(Y) + \sum_{j=1}^{j} \sum_{k=1}^{K} (r - c_k) Y_{jk}$.

Next a series of random events for each stage are generated, and the optimal action for each appointment request is found by retrieving the pre-calculated decision table. Table 3.2 compares the optimal policy and the heuristic policy on this random instance from 2 different aspects The first two columns list the index of the stage and the events happen in each stage. For example, “App (type $k$)” means a type $k$ patient calls for making an appointment, “Cancel (stage $n$)” means the patient who has called in stage $n$ asks to cancel his/her appointment, and “No calls” means no patients has been called during this period. The next 2 columns show the details of the optimal policy. If it is an appointment request, the optimal action, $A_o$, is assigning this patient to a slot, denoted by “+ Slot $j$”; if it is a cancellation request, the action would be removing this patient from the schedule, denoted by “- Slot $j$”. The updated schedule, $Y_o$, after taking the optimal action is reported in the next column. The schedule of each slot are separated by semi-colons, and for each slot, the $k$th element shows the number of type $k$ patients scheduled to that slot. And similarly, the last 2 columns present action and resulting scheduling of the heuristic policy, denoted by the subscript $h$.

**Table 3.2:** Comparison of optimal policy and the heuristic policy on a random instance

| | | Optimal Policy | | Heuristic Policy | |
|---|---|---|---|---|---|
| Stage $n$ | Event | $A_o$ | Schedule $Y_o$ | $A_h$ | Schedule $Y_h$ |
| 1 | App (type 2) | + Slot 4 | [0 0; 0 0; 0 0; 0 1] | + Slot 1 | [0 1; 0 0; 0 0; 0 0] |
| 2 | App (type 2) | + Slot 4 | [0 0; 0 0; 0 0; 0 2] | + Slot 4 | [0 1; 0 0; 0 0; 0 1] |
| 3 | App (type 2) | + Slot 3 | [0 0; 0 0; 0 1; 0 2] | + Slot 1 | [0 2; 0 0; 0 0; 0 1] |
| 4 | App (type 2) | + Slot 3 | [0 0; 0 0; 0 2; 0 2] | + Slot 3 | [0 2; 0 0; 0 1; 0 1] |
| 5 | App (type 2) | + Slot 2 | [0 0; 0 1; 0 2; 0 2] | + Slot 4 | [0 2; 0 0; 0 1; 0 2] |
| 6 | App (type 1) | + Slot 1 | [1 0; 0 1; 0 2; 0 2] | + Slot 2 | [0 2; 1 0; 0 1; 0 2] |
| 7 | App (type 1) | + Slot 1 | [2 0; 0 1; 0 2; 0 2] | + Slot 3 | [0 2; 1 0; 1 1; 0 2] |
| 8 | Cancel (stage 7) | - Slot 1 | [1 0; 0 1; 0 2; 0 2] | - Slot 3 | [0 2; 1 0; 0 1; 0 2] |
| 9 | App (type 1) | + Slot 1 | [2 0; 0 1; 0 2; 0 2] | + Slot 3 | [0 2; 1 0; 1 1; 0 2] |
| 10 | App (type 1) | + Slot 4 | [2 0; 0 1; 0 2; 1 2] | + Slot 1 | [1 2; 1 0; 1 1; 0 2] |
| 11 | Cancel (stage 6) | - Slot 1 | [1 0; 0 1; 0 2; 1 2] | - Slot 2 | [1 2; 0 0; 1 1; 0 2] |
| 12 | Cancel (stage 4) | - Slot 3 | [1 0; 0 1; 0 1; 1 2] | - Slot 3 | [1 2; 0 0; 1 0; 0 2] |
| 13 | No calls | N/A | [1 0; 0 1; 0 1; 1 2] | N/A | [1 2; 0 0; 1 0; 0 2] |
| 14 | App (type 1) | + Slot 1 | [2 0; 0 1; 0 1; 1 2] | + Slot 2 | [1 2; 1 0; 1 0; 0 2] |
| 15 | App (type 1) | + Slot 3 | [2 0; 0 1; 1 1; 1 2] | + Slot 3 | [1 2; 1 0; 2 0; 0 2] |

In this instance, total 11 appointments and 3 of them get canceled. At the end of the decision stage, there are total 5 type 1 patients and 4 type 2 patients, which are around half of the booking capacity of each type patient $M * J = 8$. Therefore, in this instance, no appointment has been rejected. In general, the optimal policy tends to assign type 2 patients to later of the day, starting to fill slot 4 up to the capacity and then slot 3 and so on; while the heuristic policy is different from the optimal one, and doesn't have a significant pattern. The decisions highly depend on the parameter setting, and parameter changes may lead to a significantly different action sequence. This experiment is to show how the optimal policy and heuristic policy work.

Figure 3.5 plots the value of $W^{n+1}(Y_o)$ ( net reward o $Y$ at stage $n$, shown as a blue dash line) and $W^N(Y_o)$ (the net reward if $Y$ is the final schedule, shown as a black solid line) and labels each decision stage by the event. For example, 'T2A" is short for "App (type 2)", "S6C" is short for "Cancel (stage 6)", and so on.

In general, the net reward increases whenever accepting an appointment. In addition, the gap between these two values has a decreasing trend along with the increase of the decision stage, and they converge to the same value at the last stage. The marginal reward of accepting the first type 1 patient is the largest. In addition, the value of $W^N(Y_h)$ is also plotted in Figure 3.5, shown as a red

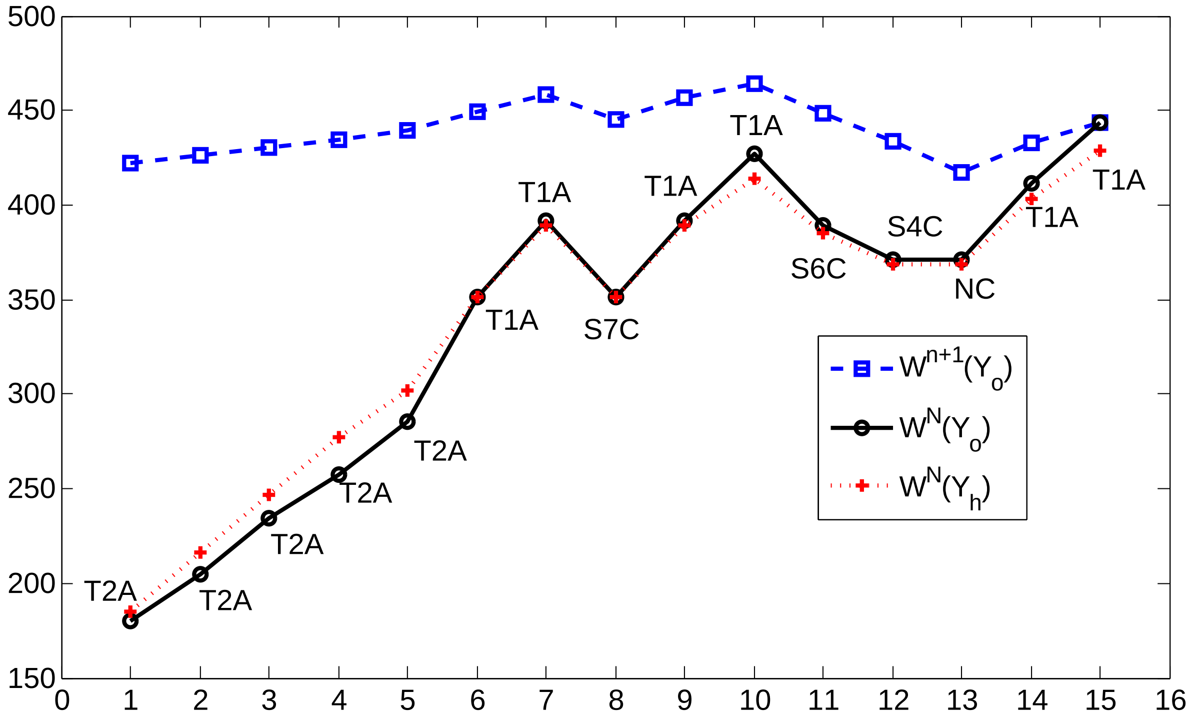


**Figure 3.5:** The evolution of the net rewards

dotted line. Comparing with the optimal policy, the net reward of the heuristic policy is slightly higher or the same at the first 9 stages. From stage 10 to the end, the optimal policy shows a slight advantages. The net reward of the final schedule of the optimal policy is 443.7, being 15 units (3.5%) higher than the heuristic policy.

To have a rough idea how the computational time would change when the size of the problem increases, we have tested two more instances with 5 and 6 time slots while keeping the other parameters same. For the instance with 5 time slots, the state space is $3^{10} = 59,046$. It takes 6,191 seconds to evaluate the final $\mathbb{U}^N$, and 3.6 seconds to calculate the value function of each of the other stages. For the instances with 6 time slots, the stage space is $3^{12} = 531,441$. It takes 83,866 seconds, about 23.3 hours, to evaluate the final $\mathbb{U}^N$, and 33.7 seconds to calculate the value function of each of the other stages. It will take a couple of days to obtain the decision table for larger size instances

### 3.6.2 Policy comparison

We introduce another heuristic policy as a benchmark of the performance of the optimal policy and the proposed heuristic policy. The round robin policy is similar to the proposed heuristic policy, except that it uses the scheduled number of patients $\sum_{k=1}^{K} Y_{jk}$ instead of calculating $\overline{Y}_j$, and it is easy to implement and leads to a schedule that is evenly spread out if all patients show up (Muthuraman and Lawley, 2008). The three policies are compared on 8 different instances. All the instances have 2 types of patients and the maximum capacity of each slot for each type of patient is 2. The instance size ranges from 4 time slots and 15 decision stages to 5 time slots and 60 stages. For each size of the instance, we run 10000 replications, and each replication has different event sequence.

Three pairs of policies are compared, including optimal versus heuristic, optimal versus round robin, heuristic versus round robin. The decision table retrieving as well as the other two scheduling policy requires trivial calculation, therefore the details of the computational time are not discussed here. Instead, we focus on the net reward earned by the three policies. Specifically, for each policy pair, the absolute differences and percent differences of the final schedule net reward are calculated. Figure 3.6 and 3.7 present these differences for all 10000 replications of instance with 4 slot and 15 stages. Note that the final schedule of each policy is distinguished by their subscripts, i.e., $o$ stands for optimal policy, $h$ stands for heuristic policy, and $r$ is round robin policy.

In general, the optimal policy is better than the heuristic policy, and the round robin policy performs the worst: Referring the solid black line, the heuristic policy performs better than the optimal policy for about 100 instances; the optimal policy yields higher net reward for 5500 instances; and for the rest of the instances the two policies have equivalent performance. Referring to the blue dash line, the advantage of the optimal policy over the round robin policy is more significant, and it yields higher net reward for about 8000 instances. Referring to the red dot line, the round robin policy is better than the heuristic policy on about 1500 instances, while heuristic policy has a higher reward on 4500 instances.

Table 3.3 listed the average absolute (denoted by "abs") and percent (denoted by "per") net reward differences of each pair of the policies. Take the first instance as an example, on average,

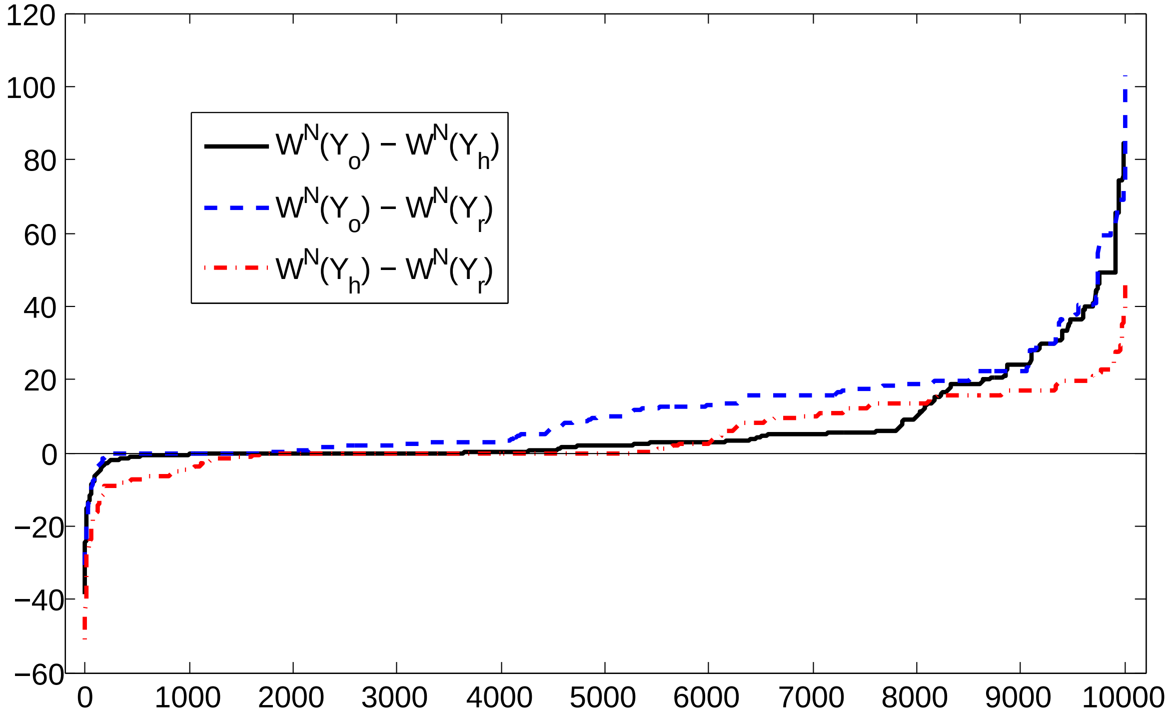


**Figure 3.6:** Net reward comparison between three scheduling policy (in dollars)

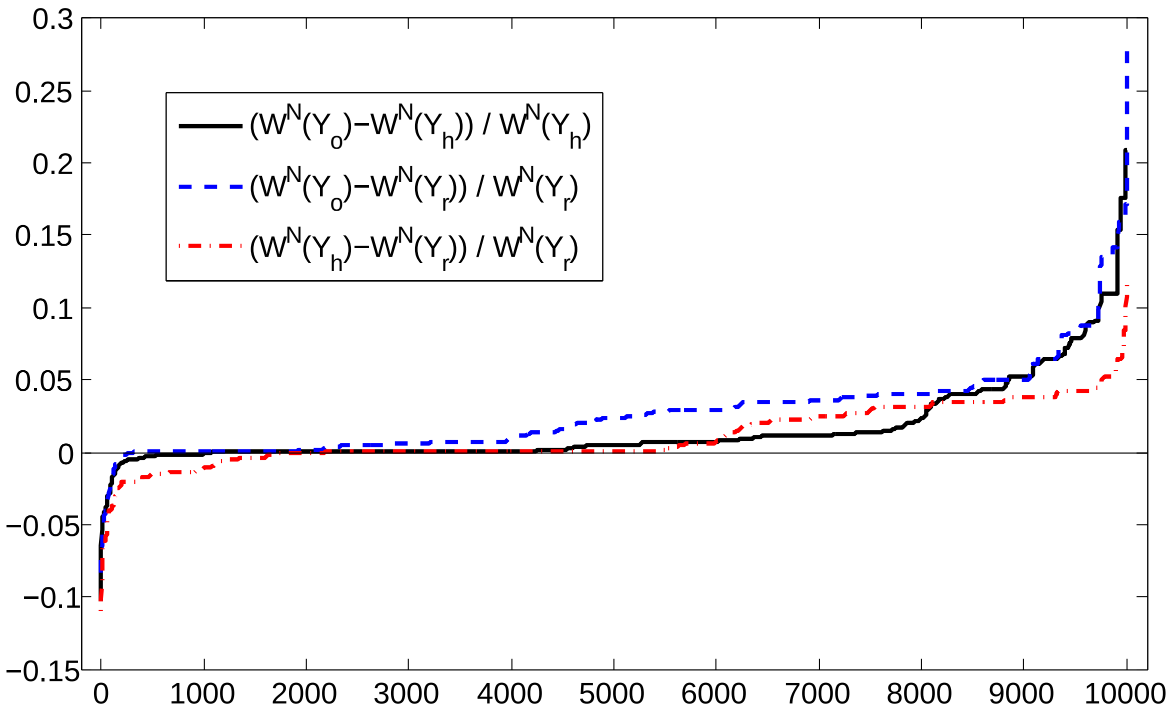


**Figure 3.7:** Net reward comparison between three scheduling policy (in percent)

**Table 3.3:** Net reward comparison between three scheduling policies

| | $\mathbb{W}^N(Y_o) - \mathbb{W}^N(Y_h)$ | | $\mathbb{W}^N(Y_o) - \mathbb{W}^N(Y_r)$ | | $\mathbb{W}^N(Y_h) - \mathbb{W}_N(Y_r)$ | |
|---|---|---|---|---|---|---|
| ins | abs | per | abs | per | abs | per |
| 4 slots, 15 stages | 6.83 | 1.55% | 11.47 | 2.62% | 4.64 | 1.07% |
| 4 slots, 30 stages | 9.37 | 2.14% | 13.57 | 3.11% | 4.19 | 1.00% |
| 4 slots, 45 stages | 6.26 | 1.42% | 8.49 | 1.93% | 2.23 | 0.53% |
| 4 slots, 60 stages | 3.64 | 1.21% | 4.85 | 0.82% | 1.21 | 0.29% |
| 5 slots, 15 stages | 2.66 | 0.56% | 9.15 | 1.94% | 6.49 | 1.38% |
| 5 slots, 30 stages | 10.78 | 1.89% | 26.07 | 4.75% | 15.29 | 2.83% |
| 5 slots, 45 stages | 7.35 | 1.29% | 16.99 | 3.05% | 9.64 | 1.77% |
| 5 slots, 60 stages | 4.02 | 0.71% | 9.2 | 1.63% | 5.17 | 0.95% |

the optimal policy is 6.83 (1.55%) and 11.47 (2.62%) higher than the heuristic policy and the round robin policy, respectively; and heuristic policy is 4.64 (1.07%) higher than the round robin policy. Under the current problem setting, the differences between the optimal policy and the heuristic policy is smaller for instances with 5 slots, and the advantages of the proposed heuristic policy over the round robin policy is more significant.

To sum up, the proposed heuristic policy is slightly worse than the optimal policy and better than the round robin policy. Considering the time of computing the decision table is quite long for large instances, the proposed heuristic policy would be a good scheduling policy for real-size problem.

## 3.7 Summary

This chapter studies an online optimization problem of scheduling appointment for one physician in an outpatient clinic. It is required the schedule manager makes decisions upon receiving the call to maximize the expected net reward of the system, which is calculated as the revenue of seeing patients minus the patient indirect and direct waiting cost as well as the physician's overtime cost. The decisions are either accepting the appointment and assigning the patient to a slot of a day or rejecting it. A good scheduling decision not only considers the current state of the system, but also takes the future demand arrival into account. Once the decisions have been made, the schedule manager cannot change them only if the patients call for canceling the appointment. The cancellations as well as no-shows are another two important factors need to be considered when making the schedule decisions. Previous empirical research shows that the cancellation rate and no-show rate are related to the length of the indirect waiting time. The uncertainties in demand arrivals, cancellation, no-shows, and service times make this problem is very hard to solve.

To obtain the optimal policy, an MDP model is built to solve the problem. We further developed an equivalent value function that generates the same optimal policy as the original one. Then this equivalent value function is proved to be non-increasing concave function of the schedule for all decision stages. Based on this analysis, we have proved that the optimal booking policy is a book limit policy. The backward induction method is used to solve the MDP model and find the optimal policy. In addition, we have proposed a heuristic policy, i.e., assigning a patient to a slot with the lowest expected number of patients given not exceeding the booking capacity. The optimal policy, heuristic policy, and a round robin policy are compared on randomly generated instances. Although the decision table of the optimal policy takes long time to compute, it can be calculated off-line and the time for retrieving the decision table is negligible. However, along with the increase of the problem size, it could take days to obtain the optimal decision table. The heuristic policy is easy to implement and does not require off-line computing. Although it yields less net reward than the optimal policy, it does perform better than the commonly used round robin policy. Therefore, when the computational time of the optimal decision table is too long to be affordable, the proposed heuristic policy is a good solution.

# Chapter 4

# Conclusions and Future Research

This work studies two problems in the healthcare delivery system, including an elective surgery scheduling problem in an ambulatory surgical center and an appointment scheduling problem in an outpatient clinic, by applying a variety of operations research techniques, especially the techniques addressing uncertainties. The remainder of this chapter summarizes the findings of these two problems and points out the direction for future research.

## 4.1 Summary of Findings

Chapter 2 studies the elective surgery scheduling problem in a surgical center with multiple operating rooms and recovery beds. The set of surgeries are known in advance, including surgery type, surgeons, estimated surgery durations, and estimated surgery recovery time. We proposed an MIP model and a CP model for solving the deterministic version of the problem, which is applicable when these estimations are accurate. The CP model has advantages in modeling logic constraints such as no overlap of two surgeries. The numerical experiments show that the CP model also performs better in the terms of the computational time and quality of solutions. When the duration estimations are not accurate, it is necessary to take the uncertainties into consideration when building mathematical models. We propose a two-stage stochastic mixed integer programming model to address this problem, and solve it by the L-shaped algorithm. The commercial solver Cplex is used to solve the decompose (mixed integer) linear programming models, therefore, the capability of this stochastic model and algorithm is subject to the computational limitation of the Cplex solver.

Chapter 3 investigates the appointment scheduling problem of a single physician in an outpatient clinic. When a patient calls for making an appointment of a particular day, the scheduling manager determines whether to accept this appointment and assign it to a selected slot or reject it and ask the patient if another day is acceptable. The decision maker cannot change the schedule, however, the patients may change their mind and cancel their appointments or simply do not show up. A good scheduling policy should take the uncertainties in demand, patient arrival, cancellation, no-shows, service time, and walk-ins into consideration. An MDP model is proposed to analyze this dynamic process. The scheduling horizon is divided into a finite number of decision stage, and at most one call will be received at each stage. The value functions of the MDP model are proved to be non-increasing concave functions for all stages, which leads to the conclusion that the optimal policy takes the form of book limit policy. The optimal policy can be obtained by the backward induction method. This method requires long computational time, which limits the application of the optimal policy. Therefore, we further proposed a heuristic policy, which has compatible performance as the optimal policy and is also easy to implement.

## 4.2 Future Research Direction

In the stochastic version of the elective surgery scheduling problem, we have assumed that the scenarios of the surgery durations are finite. This assumption can be relaxed in future research, and one could apply sampling techniques to construct appropriate scenarios to approximate the uncertainty outcome space. This study focuses on the scheduling problem before the day of the surgery. In the future research, we could expand our study to the schedule adjustment problem on the day of the surgery when a surgery takes an unexpectedly long time or a surgery finishes too early. In addition, we can relax the assumption that the number of the recovery beds is fixed, and allow the operations research tool to figure out the right number of recovery beds.

For the appointment scheduling problem, we make a reasonable assumption that patients usually consult with their own primary care physicians. However, sometimes, the patients show no preferences on physicians in the clinic. By pooling the scheduling of multiple physicians, we can assign these patients to the physicians with less workload. It is a worthwhile research topic and might lead to higher profit of the clinic. Moreover, this study considers both the pre-scheduled patients and walk-ins but does not differentiate them. The emergency walk-ins should have the highest priority to get served, while the regular walk-ins should be assigned the lowest priority. In addition, we could estimate the cancellation and no-show probability of the return patients more precisely by using the historical records on their punctuality if available.

# References

Abdennadher, S. and Schlenker, H. (1999). Nurse scheduling using constraint logic programming. In *Proc. of the Elecenth Conference on Innovative application of Artificial Intelligence*, pages 838–843. AAAI Press. 26

Abdus-Salaam, H., Davis, L., and de Oliveira Mota, D. (2010). Modeling dependent demand arrivals within an open-access scheduling system. In *Science and Technology for Humanity (TIC-STH), 2009 IEEE Toronto International Conference*, pages 256–261. IEEE. 67

Adan, I. and Vissers, J. (2002). Patient mix optimisation in hospital admission planning: a case study. *International journal of operations & production management*, 22(4):445–461. 18

Arnaout, J. (2010). Heuristics for the maximization of operating rooms utilization using simulation. *Simulation*, 86(8-9):573–583. 15

Arnaout, J. and Kulbashian, S. (2008). Maximizing the utilization of operating rooms with stochastic times using simulation. In *Proceedings of the 40th conference on winter simulation*, pages 1617–1623. Winter Simulation Conference. 15

Bartak, R. (2005). Constraint propagation and backtracking-based search. In *Lecture notes. First international summer school on CP*. 25

Batun, S. (2011). *Scheduling multiple operating rooms under uncertainty*. PhD thesis, University of Pittsburgh. 13, 40, 46, 51

Beliën, J. and Demeulemeester, E. (2007). Building cyclic master surgery schedules with leveled resulting bed occupancy. *European Journal of Operational Research*, 176(2):1185–1204. 19

Birge, J. R. and Louveaux, F. (2011). *Introduction to stochastic programming*. Springer. 46, 51

Blake, J., Carter, M., et al. (1997). Surgical process scheduling: a structured review. *Journal of the Society for Health Systems*, 5(3):17. 14

Blake, J., Dexter, F., and Donald, J. (2002). Operating room managers use of integer programming for assigning block time to surgical groups: A case study. *Anesthesia & Analgesia*, 94(1):143–148. 11, 18

Bourdais, S., Galinier, P., and Pesant, G. (2003). Hibiscus: A constraint programming application to staff scheduling in health care. In *Proc. of International Conference on Principles and Practice of Constraint Programming, Lecture Notes in Computer Science 2833*, pages 153–167. 26

Cardoen, B., Demeulemeester, E., and Beliën, J. (2009). Optimizing a multiple objective surgical case sequencing problem. *International Journal of Production Economics*, 119(2):354–366. 18

Cardoen, B., Demeulemeester, E., and Beliën, J. (2010). Operating room planning and scheduling: A literature review. *European Journal of Operational Research*, 201(3):921–932. 10

Cayirli, T. and Veral, E. (2003). Outpatient scheduling in health care: a review of literature. *Production and Operations Management*, 12(4):519–549. 63, 64, 66, 68, 69, 70

Cayirli, T., Veral, E., and Rosen, H. (2009). Assessment of patient classification in appointment system design. *Production and Operations Management*, 17(3):338–353. 64, 67, 69

Chaabane, S., Meskens, N., Guinet, A., and Laurent, M. (2006). Comparison of two methods of operating theatre planning: application in belgian hospital. In *Service Systems and Service Management, 2006 International Conference on*, volume 1, pages 386–392. IEEE. 18

Chakraborty, S., Muthuraman, K., and Lawley, M. (2010). Sequential clinical scheduling with patient no-shows and general service time distributions. *IIE Transactions*, 42(5):354–366. 68

Denton, B., Miller, A., Balasubramanian, H., and Huschka, T. (2010). Optimal allocation of surgery blocks to operating rooms under uncertainty. *Operations research*, 58(4-Part-1):802–816. 2

Denton, B., Viapiano, J., and Vogl, A. (2007). Optimization of surgery sequencing and scheduling decisions under uncertainty. *Health Care Management Science*, 10(1):13–24. 7, 13

Dexter, F. and Traub, R. (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. 32

Erdogan, S. and Denton, B. (2011). Surgery planning and scheduling. In Cochran, J., Cox, L., Keskinocak, P., Kharoufeh, J., and Smith, J., editors, *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, Hoboken, NJ,. 2, 7, 8, 10, 12

Fei, H., Chu, C., Meskens, N., and Artiba, A. (2008). Solving surgical cases assignment problem by a branch-and-price approach. *International journal of production economics*, 112(1):96–108. 12, 18

Fei, H., Meskens, N., and Chu, C. (2010). A planning and scheduling problem for an operating theatre using an open scheduling strategy. *Computers & Industrial Engineering*, 58(2):221–230. 11

Gerchak, Y., Gupta, D., and Henig, M. (1996). Reservation planning for elective surgery under uncertain demand for emergency surgery. *Management Science*, pages 321–334. 12

Green, L. and Savin, S. (2008). Reducing delays for medican appointments: a queueing model. *Operations Research*, 56(6):1526–1538. 65, 66, 68

Guerriero, F. and Guido, R. (2011). Operational research in the management of the operating theatre: a survey. *Health Care Managment Science*, 14(1):89–114. 10

Guinet, A. and Chaabane, S. (2003). Operating theatre planning. *International Journal of Production Economics*, 85(1):69–81. 12, 14, 19

Gul, S. (2010). *Optimization of surgery delivery systems*. PhD thesis, Arizona State Unviersity. 3

Gupta, D. and Denton, B. (2008). Appointment scheduling in health care: challenges and opportunities. *IIE Transactions*, 40(9):800–819. 11, 63

Gupta, D. and Wang, L. (2008). Revenue management for a primary-care clinic in the presence of patient choice. *Operations Research*, 56(3):576–592. 65, 67

Hanset, A., Meskens, N., and Duvivier, D. (2010). Using constraint programming to schedule an operating theatre. In *Proceedings of the IEEE workshop on Healthcare Management*, pages 1–6. 26

Hixon, A. L., Chapman, R. W., and Nuovo, J. (1999). Failure to keep clinic appointments: implications for residency education and productivity. *FAMILY MEDICINE-KANSAS CITY-*, 31:627–630. 61

Hsu, V., de Matta, R., and Lee, C. (2003). Scheduling patients in an ambulatory surgical center. *Naval Research Logistics (NRL)*, 50(3):218–238. 14, 37

Jebali, A., Hadj Alouane, A., and Ladet, P. (2006). Operating rooms scheduling. *International Journal of Production Economics*, 99(1):52–62. 13, 14, 18, 32

Kaandorp, G. and Koole, G. (2007). Optimal outpatient appointment scheduling. *Health Care Management Science*, 10(3):217–229. 65, 66

Keehan, S., Sisko, A., Truffer, C., Smith, S., Cowan, C., Poisal, J., Clemens, M., et al. (2008). Health spending projections through 2017: the baby-boom generation is coming to medicare. *Health Affairs*, 27(2):w145–w155. 2

Kharraja, S., Albert, P., and Chaabane, S. (2006). Block scheduling: Toward a master surgical schedule. In *Service Systems and Service Management, 2006 International Conference on*, volume 1, pages 429–435. IEEE. 18

Kharraja, S., Chaabane, S., and Marcon, E. (2002). Evaluation de performances pour deux stratégies de programmation opératoire de bloc. In *Conférence Internationale Francophone d'Automatique CIFA*. 14

Kim, S. and Giachetti, R. (2006). A stochastic mathematical appointment overbooking model for healthcare providers to improve profits. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 36(6):1211–1219. 66, 68, 69

LaGanga, L. and Lawrence, S. (2007a). Appointment scheduling with overbooking to mitigate productivity loss from no-shows. In *Conference Proceedings of Decision Sciences Institute Annual Conference*, pages 17–20. 64, 66

LaGanga, L. and Lawrence, S. (2007b). Clinic overbooking to improve patient access and increase provider productivity*. *Decision Sciences*, 38(2):251–276. 68

Lamiri, M., Xie, X., Dolgui, A., and Grimaud, F. (2008a). A stochastic model for operating room planning with elective and emergency demand for surgery. *European Journal of Operational Research*, 185(3):1026–1037. 12, 13

Lamiri, M., Xie, X., and Zhang, S. (2008b). Column generation approach to operating theater planning with elective and emergency patients. *IIE Transactions*, 40(9):838–852. 12, 13

Laporte, G. and Louveaux, F. V. (1993). The integer l-shaped method for stochastic integer programs with complete recourse. *Operations research letters*, 13(3):133–142. 52

Lippman, S. A. and Stidham Jr, S. (1977). Individual versus social optimization in exponential congestion systems. *Operations Research*, 25(2):233–247. 85

Liu, N., Ziya, S., and Kulkarni, V. (2010). Dynamic scheduling of outpatient appointments under patient no-shows and cancellations. *Manufacturing & Service Operations Management*, 12(2):347–364. 64, 65, 66, 68

Magerlein, J. and Martin, J. (1978). Surgical demand scheduling: a review. *Health services research*, 13(4):418. 10

Marcon, E. and Dexter, F. (2006). Impact of surgical sequencing on post anesthesia care unit staffing. *Health Care Management Science*, 9:87–98. 14

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. 11

Muthuraman, K. and Lawley, M. (2008). A stochastic overbooking model for outpatient clinical scheduling with no-shows. *Iie Transactions*, 40(9):820–837. 61, 64, 68, 91

Pandit, J. and Carey, A. (2006). Estimating the duration of common elective operations: implications for operating list management. *Anaesthesia*, 61(8):768–776. 32

Papel, I. (2009). *Facial Plastic and Reconstructive Surgery*. Thieme Publishers Series. Thieme New York. 6

Patrick, J., Puterman, M., and Queyranne, M. (2008). Dynamic multi-priority patient scheduling for a diagnostic resource. *Operations research*, 56(6):1507–1525. 66

Persson, M. and Persson, J. (2007). Optimization modelling of hospital operating room planning: analyzing strategies and problem settings. In *Operational Research for Health Policy: Making Better Decisions: Proceedings of the 31st Annual Conference of the European Working Group on Operational Research Applied to Health Services*, page 137. Peter Lang. 18

Persson, M. and Persson, J. (2009). Health economic modeling to support surgery management at a swedish hospital. *Omega*, 37(4):853–863. 18

Pham, D. and Klinkert, A. (2008). Surgical case scheduling as a generalized job shop scheduling problem. *European Journal of Operational Research*, 185(3):1011–1025. 13, 18

Qu, X., Rardin, R., Williams, J., and Willis, D. (2007). Matching daily healthcare provider capacity to demand in advanced access scheduling systems. *European journal of operational research*, 183(2):812–826. 68

Robinson, L. and Chen, R. (2009). *The Effects of Patient No-Shows on Traditional and Open-Access Appointment Scheduling Policies*. PhD thesis, University of California at Davis. 63, 66

Roland, B., Di Martinelly, C., and Riane, F. (2006). Operating theatre optimization: a resource-constrained based solving approach. In *Service Systems and Service Management, 2006 International Conference on*, volume 1, pages 443–448. IEEE. 19

Santibáñez, P., Begen, M., and Atkins, D. (2007). Surgical block scheduling in a system of hospitals: an application to resource and wait list management in a british columbia health authority. *Health care management science*, 10(3):269–282. 18

Schmitz, H. and Kwak, N. (1972). Monte carlo simulation of operating-room and recovery-room usage. *Operations Research*, pages 1171–1180. 14

Sier, D., Tobin, P., and McGurk, C. (1997). Scheduling surgical procedures. *Journal of the Operational Research Society*, pages 884–891. 19

Spangler, W., Strum, D., Vargas, L., and May, J. (2004). Estimating procedure times for surgeries by determining location parameters for the lognormal model. *Health care management science*, 7(2):97–104. 32

Subramanian, J., Stidham, S., and Lautenbacher, C. J. (1999). Airline yield management with overbooking, cancellations, and no-shows. *Transportation Science*, 33(2):147–167. 74, 78, 79, 82

Sufahani, S. F., Razali, M., Siti, N., and Ismail, Z. (2011). A scheduling problem for hospital operating theatre. *Journal of Applied Science and Mathematics*, pages 1–6. 11

Sun, Y. and Li, X. (2011). Optimizing surgery start times for a single operating room via simulation. In *Proceedings of the 2011 Winter Simulation Conference*, pages 1330–1337. Winter Simulation Conference. 9, 13

Testi, A., Tanfani, E., and Torre, G. (2007). A three-phase approach for operating theatre schedules. *Health Care Management Science*, 10(2):163–172. 12

van Oostrum, J., Van Houdenhoven, M., Hurink, J., Hans, E., Wullink, G., and Kazemier, G. (2008). A master surgical scheduling approach for cyclic scheduling in operating room departments. *OR spectrum*, 30(2):355–374. 18

Vissers, J., Adan, I., and Bekkers, J. (2005). Patient mix optimization in tactical cardiothoracic surgery planning: a case study. *IMA Journal of Management Mathematics*, 16(3):281–304. 18

Wang, P. (1993). Static and dynamic scheduling of customer arrivals to a single-server system. *Naval Research Logistics*, 40(3):345–360. 13

Weil, G., Heus, K., Francois, P., and Poujade, M. (1995). Constraint programming for nurse scheduling. *IEEE Engineering in Medicine and Biology Magazine*, 14(4):417–422. 26

Wijewickrama, A. and Takakuwa, S. (2008). Outpatient appointment scheduling in a multi facility system. In *Simulation Conference, 2008. WSC 2008. Winter*, pages 1563–1571. IEEE. 65

Zeng, B., Turkcan, A., Lin, J., and Lawley, M. (2010). Clinic scheduling models with overbooking for patients with heterogeneous no-show probabilities. *Annals of Operations Research*, 178(1):121–144. 68

Zhang, B., Murali, P., Dessouky, M., and Belson, D. (2008). A mixed integer programming approach for allocating operating room capacity. *Journal of the Operational Research Society*, 60(5):663–673. 18

# Vita

Zhaoxia Zhao received the B.S. degree in Logistics Engineering from Huazhong University of Science and Technology, China, in June 2009. After that, she joined the Department of Industrial and Systems Engineering at the University of Tennessee, Knoxville. She is expected to complete her Doctor of Philosophy degree in Industrial Engineering in August 2014. Her research interests include multi-objective optimization, facility location, scheduling, and stochastic optimization in Operations Research. She has published several peer-review journal and conference papers. She took an intern in the Department of Revenue Management and Analytics at Walt Disney Parks and Resorts, Orlando, FL in 2013, and will join the Operations Research group at Revenue Analytics in Atlanta, GA on June 2014.