Optimization of Surgery Delivery Systems

by

Serhat Gul

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

ARIZONA STATE UNIVERSITY

December 2010

Optimization of Surgery Delivery Systems

by

Serhat Gul

has been approved

September 2010

Graduate Supervisory Committee:

John W. Fowler, Co-Chair
Brian T. Denton, Co-Chair
Teresa Wu
Muhong Zhang

ACCEPTED BY THE GRADUATE COLLEGE

ABSTRACT

Optimization of surgical operations is a challenging managerial problem for surgical suite directors. This dissertation presents modeling and solution techniques for operating room (OR) planning and scheduling problems. First, several sequencing and patient appointment time setting heuristics are proposed for scheduling an Outpatient Procedure Center. A discrete event simulation model is used to evaluate how scheduling heuristics perform with respect to the competing criteria of expected patient waiting time and expected surgical suite overtime for a single day compared to current practice. Next, a bi-criteria Genetic Algorithm is used to determine if better solutions can be obtained for this single day scheduling problem. The efficacy of the bi-criteria Genetic Algorithm, when surgeries are allowed to be moved to other days, is investigated. Numerical experiments based on real data from a large health care provider are presented. The analysis provides insight into the best scheduling heuristics, and the tradeoff between patient and health care provider based criteria. Second, a multi-stage stochastic mixed integer programming formulation for the allocation of surgeries to ORs over a finite planning horizon is studied. The demand for surgery and surgical duration are random variables. The objective is to minimize two competing criteria: expected surgery cancellations and OR overtime. A decomposition method, Progressive Hedging, is implemented to find near optimal surgery plans. Finally, properties of the model are discussed and methods are proposed to improve the performance of the algorithm based on the special structure of the model.

It is found simple rules can improve schedules used in practice. Sequencing surgeries from the longest to shortest mean duration causes high expected overtime, and should be avoided, while sequencing from the shortest to longest mean duration performed quite well in our experiments. Expending greater computational effort with more sophisticated optimization methods does not lead to substantial improvements. However, controlling daily procedure mix may achieve substantial improvements in performance. A novel stochastic programming model for a dynamic surgery planning problem is proposed in the dissertation. The efficacy of the progressive hedging algorithm is investigated. It is found there is a significant correlation between the performance of the algorithm and type and number of scenario bundles in a problem instance. The computational time spent to solve scenario subproblems is among the most significant factors that impact the performance of the algorithm. The quality of the solutions can be improved by detecting and preventing cyclical behaviors.

ACKNOWLEDGMENTS

This research was supported by National Science Foundation (DMI-0620504) of the United States. Research in Chapter 2 was also partially supported by Mayo Clinic, Minnesota. I am, especially, grateful to Todd Huschka for his feedback related to the application of the work in industry. I owe many thanks to my co-advisors John W. Fowler and Brian T. Denton for their continuous guidance through the four years of my PhD studies. I also would like to thank John W. Fowler particularly for his understanding and support at times I was trying to handle my personal health problems in addition to the health care systems engineering problems. I want to thank Brian T. Denton particularly for his thorough research feedback, general advices and extremely quick responses.

My friends and colleagues at Arizona State University also deserve thanks. In alphabetical order, I want to thank Adrian, Baris, Baykal, Mustafa, Ozgur, and Shrikant for their support. In addition, Sakine from the University of Pittsburgh had valuable contributions into my studies. Most of all, I want to thank my parents for everything.

TABLE OF CONTENTS

Page

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1. Introduction

Health care expenditures in the United States are currently estimated to be 17% of the Gross Domestic Product (GDP) and expected to grow (Gruber 2010). The rising costs have caused health care managers to take operational actions to control and reduce expenditures. A reduction in cost can be achieved, in part, through increasing the efficiency. However, the other important attributes of patient care related to the quality should also be taken into account while improving the efficiency.

Operations Research methods can help improve health care delivery systems in terms of quality, efficiency, effectiveness, safety and patient access. Such methods have been applied to improve the health care operations for decades (e.g. in appointment scheduling, nurse scheduling, medical decision making). As health care organizations continue to make investments in information technology infrastructure, the availability of data is creating the potential for greater applicability of the operations research methods. Therefore, it is likely that the amount of interdisciplinary work done by the health care and operations research communities will significantly rise in the future.

1.2. Motivation

Improvement of the surgery delivery systems is particularly important due to the fact that operating room (OR) revenues constitute as much as 40% of the revenues generated by the hospitals, and a large portion of the total hospital costs (Erdogan and Denton 2010). Furthermore, many patients visit other departments in the hospital before and after surgery. Thus, patient flow in the OR suite would also have considerable impact on the flow through other departments. The interested reader is referred to Cardoen et al. (2010), Gupta (2007), Blake and Carter (1997), Przasnyski (1986), Magerlein and Martin (1978) for further motivation of the practical importance and benefits of the optimization of surgery delivery systems.

Efficient planning and scheduling of surgeries is one way of improving the patient flow through the surgical suite without making a change in the level of resources available in the suite. However, designing efficient surgical schedules is a challenging problem due to dependencies between different components in the system, uncertainty in demand for surgery and duration of surgeries, and the competing criteria of patients and providers. In this dissertation, new models are formulated to consider these challenges. The models are used to provide new insights into ways to improve efficiency of surgery planning and scheduling.

Surgery planning and scheduling decisions and their potential outcomes can be analyzed at several different levels. At the strategic level, various planning strategies are used by the health care organizations. The three most common strategies include open scheduling, block scheduling and modified block scheduling.

In the open scheduling strategy, all surgery time blocks are pooled together and any surgeon can use any one of the available blocks. This strategy provides a significant level of flexibility from a capacity management perspective; however it also requires that surgeons be willing to share OR capacity and be flexible about the daily surgery schedule.

Under the block scheduling strategy, certain blocks are first assigned to surgeons/surgical groups in advance. The surgeons/surgical groups utilize these allocated blocks to schedule their surgeries. It is critical that block allocation be done efficiently, because allocating more than necessary blocks to a surgeon/surgical group decreases the overall utilization of the surgical suite resources. One reason for poor utilization is that blocks that are reserved for a surgeon/surgical group are typically not transferred to another surgeon or group even if there is insufficient demand to utilize the block.

The modified block scheduling combines the advantages of open scheduling and block scheduling. Under this strategy, some portion of the blocks may be left as non-dedicated and thus be available for any surgical group. Alternatively, all blocks may be reserved up to a certain point before the surgery day, at which point any unused blocks are released to other surgeons/surgical groups.

In this dissertation, problems at the operational level of surgery scheduling are studied. The models, methods and analysis are valid for any of the scheduling strategies described above. Given that the information about which slots are available for the assignment of a surgery is known, the techniques and analysis provide significant insights into the planning and scheduling of surgeries.

1.3. Organization and Contributions of the Chapters

In Chapter 2, several methods are used to find optimal sequences and patient appointment times for outpatient surgeries. The challenges imposed by the uncertainty in the surgery durations, and dependencies between different steps in the surgery process are taken into consideration. First, a discrete event simulation model is used to evaluate how 12 different sequencing and patient appointment time setting heuristics perform with respect to the competing criteria of expected patient waiting time and expected surgical suite overtime for a single day compared to current practice. Second, a bi-criteria genetic algorithm (GA) is used to determine if better solutions can be obtained for this single day scheduling problem. Third, the efficacy of the bi-criteria GA when surgeries are allowed to be moved to other days is investigated. Numerical experiments are presented based on real data from a large health care provider. The analysis provides insight into the best scheduling heuristics, and the tradeoff between patient and health care provider based criteria. Finally, several important managerial insights based on the findings are summarized.

Several studies from the literature are reviewed. In comparison to these studies, the unique contributions of Chapter 2 include the following. First, a hybrid solution technique is proposed by mixing a bi-criteria GA with appointment time setting heuristics to find the (near) Pareto optimal set of schedules and reveal the tradeoff between factors affecting both the patient and the provider. Second, several commonly used scheduling heuristics are tested against the GA to estimate the potential benefits of optimization based methods for scheduling system improvements. Finally, the GA is used to estimate the potential benefits of optimizing daily procedure mix.

In chapter 3, a multi-stage stochastic mixed integer programming formulation is proposed for the allocation of surgeries to ORs over a finite planning horizon. The demand for surgery and surgical duration are considered to be random variables. The objective of the study is to minimize two competing criteria: expected surgery cancellations and OR overtime. The literature related to the multi-period OR planning problem is categorized into three levels based on the complexity of the variants of the problem. The first is deterministic models. The second is models with stochastic surgery durations; however the demand for elective surgeries is still deterministic in these models. Due to the latter assumption, the models have a static nature (i.e. the scheduling decisions are given at the beginning of the planning period and the decisions can not be revised over the course of the period). The third category breaks this assumption of certain demand and formulate dynamic models to study multi-period planning problems. Chapter 3 also belongs to this category which represents the most realistic case.

The study in Chapter 3 is the among the first that proposes a multi-stage stochastic programming formulation to solve the dynamic OR planning problem. The model formulates the following decision process. At each stage (e.g. day), new surgery requests are scheduled into future. Surgeries that are previously scheduled to the current day may be canceled to decrease OR overtime at the expense of cancellation cost. Canceled surgeries are then rescheduled into future days. Note that the OR assignments are also done at the time the scheduling decisions are given.

Distinct feature of the model in Chapter 3 is that it relaxes assumptions common in the existing literature, such as a Poisson arrival process for surgery requests, and independently and identically distributed surgery durations.

The model of Chapter 3 was solved using the progressive hedging algorithm (PHA), proposed by Rockafellar and Wets (1991). The PHA proceeds by applying scenario decomposition to the overall problem, iteratively solving the resulting individual scenario subproblems, and aggregating individual scenario solutions. The PHA hedges against uncertainties iteratively using its procedures until it converges to the solution for the overall problem. Although the PHA is guaranteed to converge asymptotically to a global optimal solution in the convex case, there is no guarantee for this model, because the problem studied has a non-convex nature since there are integer variables at all stages of the formulation. Due to the integer variables in the model, solving even the individual scenario subproblems can require significant computational time. The efficacy of the PHA and several related research questions are investigated in Chapter 3.

In Chapter 4, several methods are proposed and evaluated to improve the convergence speed of the PHA and quality of the PHA solutions. The methods proposed addresses the following questions: (a) What criteria should be used while updating the PHA penalty parameters at a particular iteration? (b) What criteria should be used while updating the Lagrangian multipliers at a particular iteration?

(c) What other techniques can be utilized to improve the performance of the PHA? It is found that the efficiency level of the scenario subproblem solution method is among the most significant factors that impact the performance of the PHA. The quality of the solutions are negatively affected by the cyclical behaviors and can be considerably improved by detecting and preventing cycles along the iterations.

In Chapter 5, the most significant conclusions are summarized and discussed. Chapter 5 concludes with a discussion of future research opportunities.

CHAPTER 2

BI-CRITERIA SCHEDULING OF SURGICAL SERVICES FOR AN OUTPATIENT PROCEDURE CENTER

2.1. Introduction

Surgical services require the coordination of many activities including patient intake and preparation, the surgical procedure, and patient recovery. Designing schedules that achieve smooth patient flow is a complicated task due to the dependencies between these activities. Scheduling is further complicated by considerable uncertainty in the duration of activities. These problems are amplified for Outpatient Procedure Centers (OPCs) which typically perform a variety of elective procedures on an outpatient basis. A high volume of surgical procedures combined with significant uncertainty in the duration of activities and a fixed length of time that the surgical suite is open (typically 8-10 hours) give rise to difficult stochastic scheduling problems involving multiple, competing criteria.

The physical resources in a surgical suite include operating rooms (ORs), intake rooms, and recovery rooms, as well as equipment resources such as diagnostic devices and surgical instrument kits. There are also several human resources including surgeons, nurses and nurse anesthetists.

Surgical services occur in three major steps. The first, *intake*, starts when the patient arrives at the surgical suite to initiate his/her check-in process, and ends when the patient reaches an OR bed. The *intra-operative care* period starts when the patient is admitted to the OR area and ends when the patient is taken to a recovery bed. The surgical procedure itself is performed during this period. The last step, *recovery*, starts when the patient is admitted to a recovery area and ends when the patient is discharged. Even for very routine surgeries the duration of each of these activities exhibits considerable variation (Berg et al. 2010).

In this chapter, *expected patient waiting time* and *expected surgical suite overtime* are the focused performance measures. These are among the most important performance measures that a manager (e.g. charge nurse) must consider on a daily basis. These criteria are in conflict because a schedule with small time intervals between procedures tends to have low surgical suite overtime and high patient waiting times, and vice versa. A bi-criteria analysis is performed to estimate the impact of three types of scheduling improvements and answer the following three questions:

1. What are the potential benefits of using easy-to-implement heuristics for daily appointment scheduling?

2. What are the potential benefits of optimization methods over commonly used and easy-to-implement heuristics for daily appointment scheduling?

3. What are the potential benefits of controlling daily procedure mix from day to day?

An OPC at Mayo Clinic, in Rochester, Minnesota, forms the testbed for this study. First, a discrete event simulation model (DES) is constructed and used to evaluate easy-to-implement scheduling heuristics based on expected patient waiting time and expected surgical suite overtime. The DES is a comprehensive model that includes all three major surgical service steps. Next, the simulation model is embedded within a hybrid solution method that contains both a bi-criteria Genetic

Algorithm (GA) and appointment time setting heuristics to construct a (near) Pareto optimal set of schedules. Furthermore, the GA is used to examine the potential benefits of controlling the daily surgical mix.

The remainder of the chapter is organized as follows. In the next section, some background on OPCs is provided. In Section 2.3., a brief literature review of relevant studies is presented. In Section 2.4., the simulation model is described. In Section 2.5., the methodologies I have applied including the scheduling heuristics and the GA are discussed. In Section 2.6., the experimental results are presented. Finally, the most significant managerial insights are summarized in Section 2.7..

2.2. Background on Outpatient Procedure Centers

OPCs are complex systems, often with several surgical groups (e.g. departments or subgroups within departments) sharing resources on a given day. The layout of a typical suite is illustrated in Figure 1. The physical space used for patient care can be broken into three sections. The first is the patient waiting area, the second is the pre/post room area (used for patient intake and recovery), and the third is the OR area.

Typically there is some dedication of intake, operating and recovery rooms to surgical groups. For example, in the OPC I studied, ORs are dedicated as follows: *Pain Medicine* has one OR, each of *Urology* and *Ophthalmology* has two ORs, and *Oral Maxillofacial* (OMS) has three ORs. Thus there are 8 ORs in total which are shared by the three surgical groups. There are 20 pre/post rooms, four of which are dedicated to Pain Medicine. Oral Maxillofacial also has four dedicated pre/post rooms, but the remaining 12 pre/post rooms can be utilized by any one of the surgical cases of the other groups.

The OPC depicted in Figure 1 combines resources by using the same set of rooms for intake and recovery. This increasingly common layout is motivated by the desire to balance resources and reduce congestion (since intake areas tend to be heavily utilized early in the day while recovery areas are empty and vice versa at the end of the day). Patients first go to the check-in desk, and then to the patient waiting area, where they wait for an intake room to become available. After the intake process, they wait for their surgeon and OR to become available. Once the procedure is complete, they reenter the pre/post room area to recover, and exit the OPC when their recovery is complete.

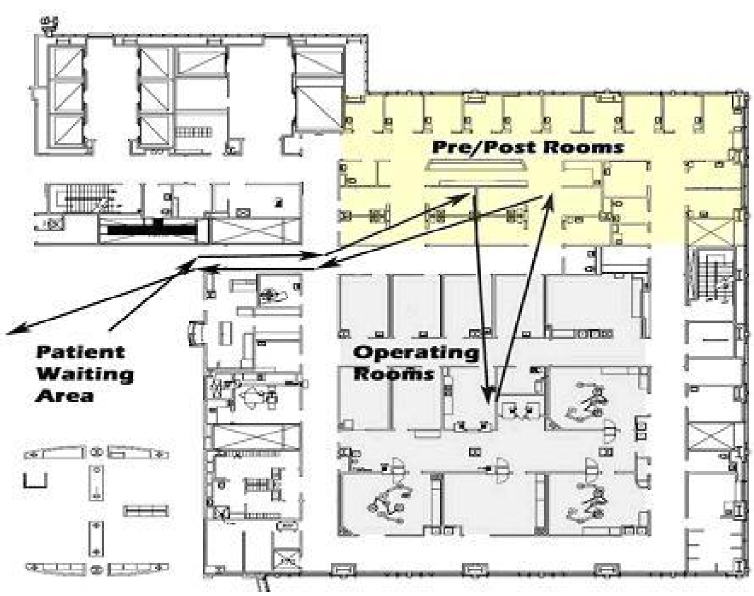

Figure 1. Layout and Patient Flow Through an OPC Including the Patient Waiting Area, Pre/Post rooms, and ORs.

There is significant uncertainty in the time necessary for completing activities in the OPC. In Figure 2, empirical estimates of probability density functions are plotted for intake, surgical procedure, and recovery, for procedures from the same surgical group. Surgical procedure durations can differ considerably among procedures even within the same surgical group and they tend to have a long tail which represents unpredictable low probability complications that may occur during the procedure. Intake and recovery distributions are generally quite similar within a surgical group. Intake distributions are similar, because patients are going through similar intake processes. Recovery distributions also do not differ, since procedures within a surgical group tend to use similar levels of anesthetic.

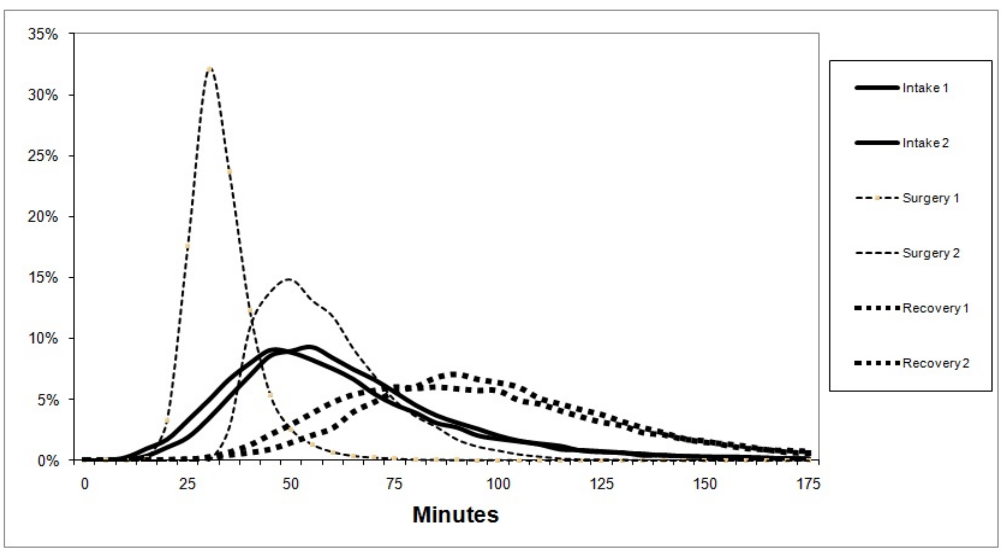

Figure 2. Probability Density Functions for Intake, Procedure, and Recovery Times for Two Different Types of Surgical Procedures within a Surgical Group

The particular OPC considered in this study opens at 8am, which is the scheduled time of the first patient's arrival. The planned closure time is 5pm. Overtime results in additional costs for those staff that stay beyond 5pm. There is also a loss of goodwill on the part of staff because most staff members prefer not to work overtime. Furthermore, there is anecdotal evidence that long patient wait times, which lead to unhappy patients, reduce staff morale and can lead to turnover, particularly among nurses.

The process flow defined above, and the probability density functions for intake, surgical procedure, recovery and other activity times are used to construct the DES model, which I describe in detail in Section 2.4..

2.3. Literature Review

Following is a brief literature review that covers several examples from the literature that are related to my work. The focus is on studies that either (a) evaluate scheduling heuristics for multiple ORs using a DES model or (b) consider resources in addition to ORs (e.g. recovery area resources) or (c) analyze multi-criteria problems related to planning and scheduling. For a more extensive review of the literature on surgery planning and scheduling, the reader is referred to Magerlein and Martin (1978), Blake and Carter (1997), Gupta (2007), Gupta and Denton (2008).

Dexter et al. (1999b) use simulation to test heuristics for allocating block time to surgeons, and schedule elective cases to maximize OR utilization. They evaluate four on-line bin packing algorithms to schedule elective cases: next fit, first fit, best fit, and worst fit. Dexter et al. (1999a) evaluate 10 different algorithms (online, off-line, and hybrid algorithms) for scheduling add-on cases into the open

OR time available to evaluate their effectiveness in increasing OR suite utilization. Testi et al. (2007) use simulation to evaluate different surgery sequences with regard to the longest waiting time of the surgeries in the waiting list, longest processing time, and shortest processing time after building the Master Surgery Schedule (MSS).

Dexter and Marcon (2006) studied the impact of several different surgery sequencing heuristics on workload of a post anesthesia care unit (PACU) including: random sequence, longest cases first (LCF), shortest cases first (SCF), Johnson's rule, and several others. The authors analyzed how sequencing affects OR overutilization, PACU completion time, delays in discharging from the OR into PACU, and the maximum number of patients in the PACU throughout the day. They found that even though LCF is the most popular rule used in practice, it is one of the worst rules with regard to the performance measures of the study. Random sequencing is suggested if it is difficult to implement rules that performed better, due to the constraints (such as medical and equipment) that are not considered in the study, because, implementation of random sequencing is trivial and it yields medium level results.

Berg et al. (2010) use a DES model to analyze an endoscopy suite with respect to surgeon-to-OR allocation scenarios. Competing performance measures such as overtime for the endoscopy suite and patient waiting time were analyzed in the model and a simulated annealing heuristic was used to improve the scheduled start time of cases with respect to expected overtime and patient waiting time. An endoscopy suite is a simplification of a general OPC since the case mix is limited to only upper and lower endoscopies. The suite considered in Berg et al. (2010) consists of three independent process areas (i.e. intake, procedure, recovery) and the authors assume that the capacities of intake and recovery areas are unlimited. In contrast I assume intake and recovery have fixed capacity and potentially limit patient flow through the suite. Finally, the authors use only a very simple simulated annealing approach to design schedules, whereas I provide a detailed comparison of standard heuristics as well as a more advanced bi-criteria genetic algorithm. Lehtonen et al. (2007) build a simulation model to analyze the effect of six process interventions on open-heart surgery with respect to OR productivity and overtime amount.

Marcon et al. (2003) simulate a surgical suite to estimate the number of PACU beds required. They also investigated the effect of a decrease in the number of porters (patient escorts) in the OR on the number of PACU beds needed. Lowery and Davis (1999) use a simulation tool to study the effect of decreasing the number of ORs in a hospital. They analyzed the effects of changes in the surgery schedule and in case times on the number of rooms required. Tyler et al. (2003) simulate an OR to determine the optimum OR utilization and analyze the important factors such as average patient waiting time and variability of case durations which impact OR utilization. Lowery (1992) uses a simulation model to simulate the patient flow through critical care units to determine the number of beds required.

Multi-criteria studies related to surgery planning and scheduling include the following. Jebali et al. (2006) developed a two-phase approach to solve the surgery assignment and sequencing problem formulated as an integer program. In their approach, operations are first assigned to ORs with the objective of minimizing hospitalization, undertime and overtime costs. Second, optimal sequences are sought for minimizing the total overtime cost for ORs. Guinet and Chaabane (2003) solved the weekly patient-to-OR assignment problem using a primal-dual heuristic. Patient satisfaction and resource efficiency are considered in this study where the objective includes the minimization of the number of days patients wait in the hospital and the overtime. Lamiri et al. (2008b) proposed a stochastic programming model for the assignment of elective surgeries to ORs over a planning horizon. Uncertainty comes from the demand for emergent cases in this formulation. The study aims to minimize both OR utilization costs and patient related costs. They solve the problem using a column generation method.

In the context of ambulatory care services, Cayirli et al. (2006) tested several sequencing and appointment rules for clinic visits using simulation with regards to patient waiting time, doctor idle time and overtime. The most significant finding of this study is that the impact of sequencing on the criteria is more important than that of the appointment rule. Lovejoy and Li (2002) consider an OR capacity expansion problem. They focus on the tradeoff between waiting time, procedure start time reliability, and hospital revenues.

This work differs from the aforementioned papers in the following ways. First, a hybrid solution technique is proposed by mixing a bi-criteria GA with appointment time setting heuristics to find the (near) Pareto optimal set of schedules and reveal the tradeoff between factors affecting both the patient and the provider. Second, several commonly used scheduling heuristics are tested against the GA to estimate the potential benefits of optimization based methods for scheduling system improvements. Finally, the GA is used to estimate the potential benefits of optimizing daily procedure mix.

2.4. Simulation Model

The DES model was developed based on an OPC in Rochester, MN (Huschka et al. 2007). It is a *terminating simulation* (Banks et al. 2005), in the sense that a finite number of procedures are scheduled each day within a pre-determined time in which the OPC is open each day. Patients arrive into the check-in area according to a deterministic schedule (constructed using one of the heuristics I discuss in Section 2.5.). It is assumed arrivals are on-time and all patients show up for their scheduled procedure (extensions such as tardiness and no-shows are straightforward with my model, however, they are uncommon in the OPC studied, and for simplicity they are not included in the analysis). Subject to pre/post room and surgeon/OR availability, patients proceed through the OPC with activity start and completion times based on samples from the continuous probability density functions of Tables 1 and 2.

The number of surgeons per surgical group on a given day is equal to the number of ORs allocated to the group and surgeons may operate in any OR assigned to their group. While these policies are not necessarily in place in all OPCs, they are reasonably common, and representative of scheduling problems faced in practice.

Table 1. Mean, Standard Deviations and Distributions of the Intake, Procedure and Recovery Times for Various Procedure Groups of the Surgical Groups with the Number of Patients Data Used to Calculate Them.

Surgical Group	Procedure Group	Process	Mean	Standard Deviation	Number of Operations	Distribution Fit
OMS	1	Intake	42.02	21.92	1472	Weibull
		Procedure	33	19.11	1472	Lognormal
		Recovery	53.02	33.88	1472	Gamma
	2	Intake	0	0	0	-
		Procedure	36	33.88	1919	Lognormal
		Recovery	0	0	0	-
Pain Medicine	1	Intake	38.4	20.22	58	Erlang
		Procedure	19.78	12.12	58	Lognormal
		Recovery	21.09	9.74	58	Weibull
	2	Intake	38.72	24.37	244	Gamma
		Procedure	20.49	10.86	244	Lognormal
		Recovery	23.64	16.65	244	Erlang
	3	Intake	34.7	21.11	1551	Gamma
		Procedure	20.93	15.08	1551	Lognormal
		Recovery	19.94	14.17	1551	Erlang
	4	Intake	32.79	16.79	24	Triangular
		Procedure	40.5	26.12	24	Lognormal
		Recovery	52.58	29.93	24	Weibull
	5	Intake	36.46	21.47	970	Gamma
		Procedure	34.01	17.42	970	Lognormal
		Recovery	23.26	15.84	970	Beta
Ophthalmology	1	Intake	65.58	26.32	1696	Gamma
		Procedure	41.63	16.43	1696	Lognormal
		Recovery	29.84	14.56	1696	Weibull
	2	Intake	65.65	28.57	589	Triangular
		Procedure	77.66	44.03	589	Lognormal
		Recovery	42.75	26.9	589	Erlang
Urology	1	Intake	64.92	27.59	329	Weibull
		Procedure	53.3	27.7	329	Lognormal
		Recovery	89.33	39.18	329	Gamma
	2	Intake	58.14	26.56	640	Gamma
		Procedure	31.3	16.37	640	Lognormal
		Recovery	94.23	36	640	Erlang
	3	Intake	64.15	22.78	153	Beta
		Procedure	138.16	56.77	153	Lognormal
		Recovery	126.95	49.55	153	Weibull
	4	Intake	61.37	25.18	345	Erlang
		Procedure	55.78	22.89	345	Lognormal
		Recovery	99.91	33.13	345	Beta
	5	Intake	58.18	26.68	496	Gamma
		Procedure	80.33	43.76	496	Lognormal
		Recovery	96.56	44.97	496	Weibull

Data from the year 2006 for 4034 patients at Mayo Clinic (corresponding to the operations of the first 21 weeks of the year) was used. Probability density functions were fit for all stages of a patient's movement through the surgical suite including intake, surgical procedure, and recovery (see Table 1 for a summary of data). The procedure times are partitioned into three parts (*pre-incision*, *incision*, and *post-incision times*) and fit distributions for each independently. This was nec-

Table 2. Distributions and Their Parameters Set Subjectively by the Experts for the Transfer Times Between Units as well as the Turnover Times for Different Rooms

Transfer Times	
Patient Flow (From - To)	Distribution
Check-in desk - Waiting area	Triangular (5,6,7)
Waiting area - Pre/post room	Triangular (2,3,4)
Pre/post room - OR	Constant (2)
OR - Pre/post room	Triangular (1,2,2)
Turnover Times	
Room Type	Distribution
Pain Medicine OR	Triangular (2,3,8)
Other ORs	Triangular (5,6.5,8)
Pre/post rooms	Triangular (5,6.5,8)

essary because these activities require different resources. For instance, the OR is utilized the entire time, but surgeons do not need to take part in the pre-incision and post-incision activities.

Distributions were fit separately for each surgical procedure type. The log-normal distribution is used for procedure times because it yielded a best fit based on maximum likelihood estimation and because it is commonly used in the literature (see, for example, Zhou and Dexter (1998)). For intake and recovery it is found Erlang, gamma, beta, Weibull, and exponential distributions were the most common best fit. OR turnover and transfer times were estimated by triangular distributions based on expert estimates of the minimum, mean, and maximum times (see Table 2).

My validation is based on a comparison of model outputs such as the number of surgeries completed per day and expected daily overtime estimates with similar values from the particular outpatient procedure practice at Mayo Clinic in Rochester, MN (i.e. the baseline schedule). The results based on the model were also presented to experts at Mayo Clinic familiar with the system including an operations research analyst specializing in surgery in the Division of Health Care Policy and Research, an Administrator for the surgical practice, and the group of nurses that work within the unit.

2.5. Methodology

The DES model is used to compare easy-to-implement heuristics used in practice with a GA-based heuristic on the basis of total expected patient waiting time and expected surgical suite overtime. Overtime is the difference between the time the last patient completes recovery and 5pm (if it is non-negative). Total patient waiting time is the sum of the times a patient spends waiting for a pre/post room to initiate intake and waiting for an OR to begin the surgical procedure. As an aggregate measure, the average of the expected patient waiting times over all patients served across all days is calculated.

In Section 2.5.1., several combinations of sequencing and appointment time heuristics are discussed for selecting the schedule of patient arrivals to the check-in area of the OPC. In Section 2.5.2., the GA-based approach is discussed.

2.5.1. Heuristics

To answer question 1 of Section 1, several combinations of patient sequencing and appointment time heuristics are tested. The cases of each OR and day combination according to four different sequencing rules are tested: increasing mean of procedure time (SPT), decreasing mean of procedure time (LPT), increasing variance of procedure time (VAR), and increasing coefficient of variation of procedure time (COV).

Given a specified sequence of patients, the first appointment is set to the beginning of the day, and subsequent appointments are set to the prior appointment time plus the estimated time for the previous patients' procedure. The estimate of the procedure time influences the patient waiting time and overtime. If the estimate is too large, it may lead to unnecessary overtime; if it is too low it may result in unnecessary patient waiting time. To explore this trade-off, the time is estimated using various percentiles of the distribution. Appointment times are determined by the following recursion:

$$A_{i+1} = A_i + h_i, i = 2, ..., n$$

where $A_1 = 0$ and h_i is the percentile of procedure i duration. This is known in the literature as *job hedging* (Yellig and Mackulak 1997) and it has been investigated extensively in the context of OR and single server appointment scheduling (for example, see Charnetski (1984), Ho and Lau (1992), Weiss (1990)).

2.5.2. A Bi-Criteria Genetic Algorithm

To answer questions 2 and 3 from Section 1 two different models are solved using a GA. The first (model A), assumes the daily procedure mix each day is fixed based on a pre-defined schedule. The second (model B) assumes the daily procedure mix may be modified by rescheduling procedures among days within a time window of n days ($n = 1$ and $n > 1$ for models A and B, respectively). The remainder of this section provides a brief summary of my GA (more complete details are presented in the appendix).

A GA is a local search algorithm based on the biological evolution paradigm (Holland 1975). An initial population is created and genetic operators are used to search the neighborhood of the initial population through successive improving iterations. At each iteration a selection is made based on the *survival of the fittest* rule to determine the members of the next generation. This mechanism continues until a stopping a criterion is met (e.g. after a fixed number of iterations, or if the solution is not sufficiently improved after a certain number of iterations).

Members of the population are called *chromosomes* and each chromosome represents a solution (in my context a solution is a surgery schedule). The chromosome stores the job hedging level, day, and known attributes of a procedure i.e. type and the surgeon for each procedure.

The algorithm starts with an initial set of solutions (note that the term solution and chromosome are used interchangeably) which are generated as follows. One of the solutions in the initial population is the actual schedule used at the OPC in the year 2006. The rest of the solutions are created using a combination of the following techniques: (i) scheduling based on the heuristics described in Section 2.5.1. and (ii) randomly assigning procedures to time slots available within the n days of time window at the actual schedule.

At each iteration solutions are evaluated using the DES model and the expected patient waiting time and expected surgical suite overtime are stored. The solutions are ranked based on these two criteria. The approach is based on the Non-Dominated Sorting Genetic Algorithm II (NSGA II) proposed by Deb et al. (2000) and is illustrated in Figure 3. The non-dominated solutions, i.e., the (near) Pareto optimal set, are assigned to the *first front*. Then, the remaining solutions are compared and the non-dominated ones are assigned to the *second front*. Using this approach the fronts of all the solutions in the population are determined and solutions are ranked based on their associated front. Solutions on the same front are further prioritized using a crowding distance operator (described in the appendix) to diversify the solution set along a given front.

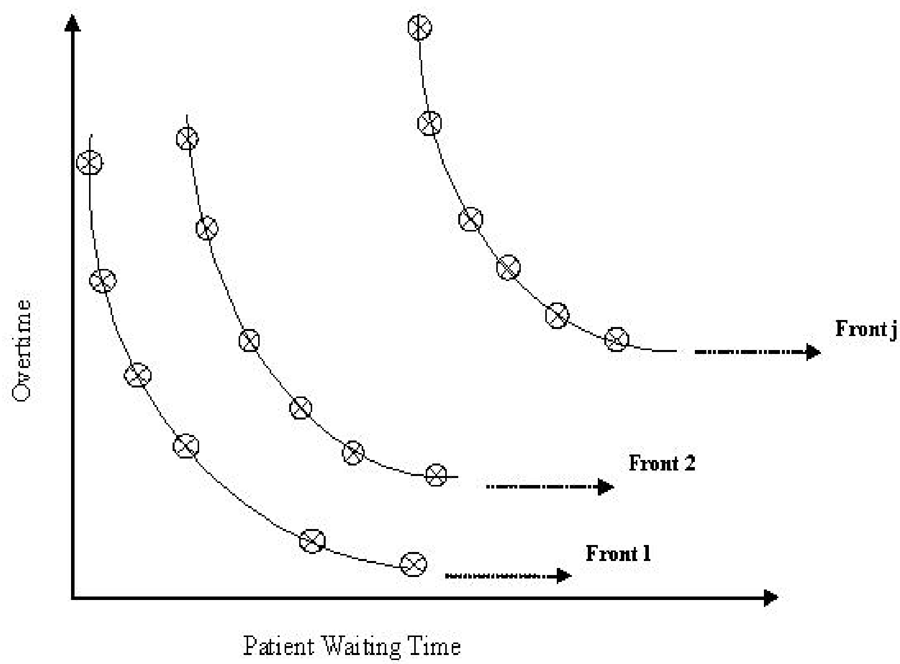

Figure 3. The Assignment of the Solutions to the Fronts in a Bi-Criteria Solution Space

To create the next generation, pairs of solutions are selected based on the ranking and combined via a crossover operator to create new pairs of solutions. A mutation operator is also applied to create near neighbors of current solutions. Repeating the same steps a fixed number of times, a new solution set is constructed at each iteration of the GA. After a defined number of iterations are completed, the algorithm terminates and the solutions on the (near) Pareto optimal set (first front) are stored as the output.

The following provides additional information about my bi-criteria Genetic Algorithm (GA). First, the pseudocode for the GA is provided. Next, the specific details about various aspects of the GA are presented.

2.5.2.1. *Pseudocode*

t = generation counter

i = chromosome index

G = number of generations

N = number of chromosomes in a generation

P_t = parent population in generation t

O_t = offspring population in generation t

C_t = pool of chromosomes in generation t

F_i = front value for chromosome i

CD_i = crowding distance value of chromosome i

Step 0. Set generation number t as 0. Form initial population P_0 having size N and set it as the current pool of chromosomes (C_0).

Step 1. Simulate chromosomes (surgery schedules). Take the two criteria values (expected patient waiting time and expected surgical suite overtime) as the returned parameter values. If $t = 0$, then skip step 2.

Step 2. Combine parent (P_t) and offspring (O_t) population to update the current pool (C_t).

Step 3. Rank each chromosome i in C_t based on the front they belong to (F_i) and their crowding distance (CD_i).

Step 4. Eliminate the poorest N chromosomes of C_t and hence leave the best N chromosomes of the current pool.

Step 5. Use binary selection tournament operator to select two candidate chromosomes from the current pool to generate a chromosome for the next generation.

Step 6. Apply crossover using the two chromosomes to generate offspring. If the GA model is A, then there is no need for resetting the day, skip Step 7. Otherwise, go to Step 7.

Step 7. Set the days of procedures by considering daily capacity thresholds set for each OR.

Step 8. Set the patient appointment times which are the key attributes of genes in the chromosomes using the time-setting heuristic type associated with the chromosome.

Step 9. Apply mutation by changing the orders of two random procedures selected from the surgery schedules. Increment generation number t and set the resulting population as O_t (offspring population). If $t > 1$, O_{t-1} becomes P_t. Otherwise (at the first iteration), P_0 is set as P_t.

Step 10. Check if the limit on the number of generations is reached (stopping criterion). If yes ($t \geq G$), terminate. Otherwise ($t < G$), go to Step 1.

2.5.2.2. *GA Operator: Selection*

The chromosomes in the pool are sorted to have a lexicographical order of chromosomes according to the *front value* (has higher importance) and *crowding distance value* (see below). Then the last N chromosomes are eliminated in the sorted list to leave the N best chromosomes in the pool. The *binary selection tournament* method (Brindle 1981) is used to select mating chromosomes from the pool. The binary selection tournament operator works as follows: Two chromosomes are selected randomly and compared with each other with respect to the front values. The crowding distance value is used as a tie breaker of the competition. The one that wins the tournament attends the crossover operation as one of the mating chromosomes. The other mating chromosome is also selected by applying the operator once again.

2.5.2.3. *GA Operator: Crowding Distance*

Chromosomes are ranked based on the front they appear on as well as a *crowding distance operator* (Deb et al. 2000). The crowding distance operator encourages diversity in the solutions with respect to the (near) Pareto optimal set to avoid generating a large number of solutions with similar expected patient waiting time and expected surgical suite overtime values.

2.5.2.4. *GA Operator: Crossover*

After selecting mating chromosomes, *uniform crossover* (Syswerda 1989) is applied to generate N offspring for the next generation. Crossover determines the order of procedures in a schedule as well as the *job hedging* level that would be used later in order to set appointment times. The crossover operation is applied independently for each procedure list of n-days to sequence procedures and then the resulting independent partial sequences are combined to have a full sequence.

2.5.2.5. *GA Operator: Schedule Construction Using Heuristics*

For model A, the patient appointment time setting method is applied since the procedure day is kept fixed there. For model B where the change in a daily procedure mix is examined, the days of the procedures are set first for each list independently. Following this, the appointment time of each patient is set for each OR and day combination.

2.5.2.6. *GA Operator: Procedure Day Setting*

For each of n-days, the surgical procedure list is determined in each OR independently. The procedures are assigned iteratively to daily lists. To control the number of procedures in a daily list, a daily capacity that the OR can serve each day is set and therefore a capacity threshold is set to prevent the method from leading to extreme values of overtime. The average daily workload for a surgical department during the study period is set as the threshold (see Table 3). These thresholds serve as an overtime control parameter in the study, i.e. the estimated duration of the procedures (the sum of the mean durations) is not permitted to exceed this threshold.

Table 3. Daily Surgical Load Capacity Allocated for an OR for Different Departments

Surgical Department	Capacity (in minutes)
OMS	480
Pain Medicine	420
Ophthalmology	350
Urology	330

2.5.2.7. *GA Operator: Mutation*

Following the sequencing and appointment time setting methods, a swap mutation operator is used by changing the orders of two randomly chosen procedures in the surgery schedules. The purpose of applying mutation is to avoid local minima or help sustain the evolution process by favoring further diversity among chromosomes.

2.6. Case Study

Preliminary experiments were performed in which the number of simulation replications was varied to see how many were needed to obtain a satisfactory trade-off between computation time and half width of the generated confidence intervals. Based on these experiments, the results below include 20 simulation replications in the evaluation of each solution.

2.6.1. Analysis of Simple Heuristics

The combinations of four different sequencing heuristics (LPT, SPT, VAR, COV) with various hedging levels are analyzed. Expected patient waiting time and expected surgical suite overtimes are estimated for each sequencing and scheduling heuristic combination. Figure 4 illustrates the results for 12 heuristics and 50%, 65%, 75% indicate the hedging (percentile) levels. The result for the baseline schedule as well as the result for a random schedule generated by randomly assigning procedures to the time slots available in the day of procedure are also plotted to serve as reference points. The 95% *confidence intervals* were calculated for each of the criteria of the 12 heuristics, baseline schedule and the random schedule and it was found they are at approximately 2 % of the mean values.

Figure 4 provides several important insights. First, the baseline schedule is in the dominated set. Second, expected patient waiting time is very sensitive to the choice of percentile used for hedging. As the percentile increases the expected patient waiting time drops while the expected surgical suite overtime increases. Also,

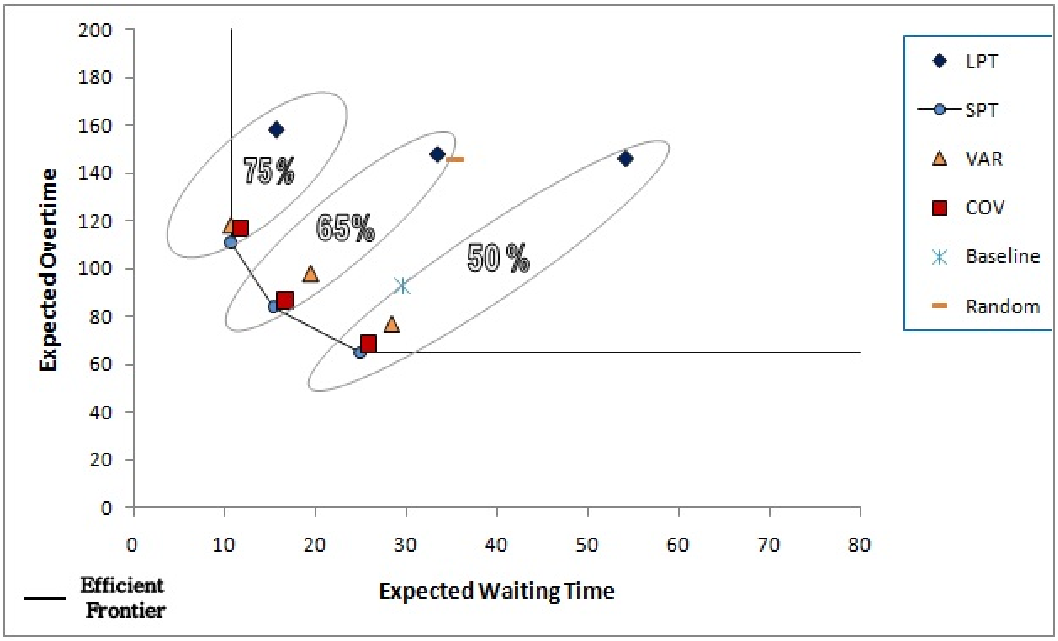

Figure 4. Expected Values for the Resulting Criteria for all Heuristics, a Random Schedule and the Baseline Schedule.

the trade-off between improvements in expected patient waiting time and expected overtime depends on the specific sequencing heuristic used to create an ordered list of surgeries. Third, among the four sequencing heuristics, SPT performs the best as it is always on the efficient frontier, while VAR and COV appear in the vicinity of the frontier. It is intuitive that there is not a considerable difference between the performance measure values from the SPT and VAR rules due to the fact that there is a positive correlation between mean and standard deviations of the procedure durations within a surgical group (see Table 1). Because of the correlation, the two procedure lists sequenced according to increasing mean and increasing variance are generally similar, and hence would yield indifferent criteria values. The correlation between these parameters is the reason for considering coefficient of variation as one of the reference for the sequencing heuristics; however the COV heuristic is outperformed by SPT. Finally, the LPT heuristic generally performs poorly and is dominated by the other heuristics. This result supports the findings of Dexter and Marcon (2006) who found that LCF (longest cases first), while being the most popular rule used in practice, is one of the worst rules they considered with regard to the criteria of their study (see Section 3 Literature Review for more details). It is found that using LPT for sequencing, and 50^{th} percentile for appointment time setting heuristic creates a schedule performing even worse than a random schedule. Intuitively, this seems to stem from the fact that LPT schedules procedures with higher variability first (due to the correlation between mean and standard deviations) which negatively affects the schedule later in the day, causing higher expected patient waiting time and expected surgical suite overtime (for a similar conclusion for a single OR case, see Denton et al. (2007)).

The most notable finding of this section is the following: *Among the sequencing heuristics, SPT yields the best schedules; while the best choice for a job hedging level depends on the heuristic used for sequencing the surgeries.*

2.6.2. Optimization Based Improvements to Simple Heuristics

Using the same data, the GA-based approach is tested in two different contexts. First, the GA is applied to the daily procedure lists assuming the procedure day is fixed (model A). Based on preliminary numerical experiments the number of solutions in a population is chosen as 40, and the number of generations as 50. The combinations of sequencing (SPT, LPT, VAR, COV) and time setting heuristics (50, 55, 60, 65, 70, 75, 80, 85^{th} percentiles) are used to provide 32 different initial solutions. The baseline schedule is also used as one of the initial solutions. The remaining 7 solutions are generated by randomly assigning the procedures to the time slots available in the surgery schedule in the same day.

In Figure 5, the GA solutions are compared with the only solutions located on the efficient frontier of heuristics revealed in Section 2.6.1. (see Figure 4). The (near) Pareto optimal set of solutions for the combination of the methods includes some GA solutions and all heuristics that use SPT as the sequencing heuristic. This indicates that the GA does not help us to improve the efficient solutions found by simple heuristics when the solution space is constrained by fixing the day of procedures. Since SPT is easy-to-implement in practice, it is more advantageous for surgical suite managers compared to the GA that requires computational resources to yield a solution.

Figure 5 also indicates the distribution of the hedging levels used for the (near) Pareto optimal set of solutions. There are 23 efficient GA solutions plotted on Figure 5 and of all, the majority (56%) use the hedging level corresponding to the 65^{th} percentile, while 21% utilize the 70^{th}, 13% 80^{th}, and 8% 60^{th} percentiles. Since it is used in the majority of the schedules on the (near) Pareto optimal set, and also provides a reasonable tradeoff between expected patient waiting time and expected surgical suite overtime, the 65^{th} percentile of the procedure time distributions seems to be a proper choice as the amount of time to allocate to procedures. On the other hand, expected surgical suite overtime values are found to be more than one hour for the other efficient schedules revealed. This would also direct managers towards the selection of 65^{th} percentile. Another insight that the graph yields is that schedules having the same hedging value generally appear in regions close to each other in criteria space. This further supports the observation that the job hedging parameter has a significant effect on both criteria.

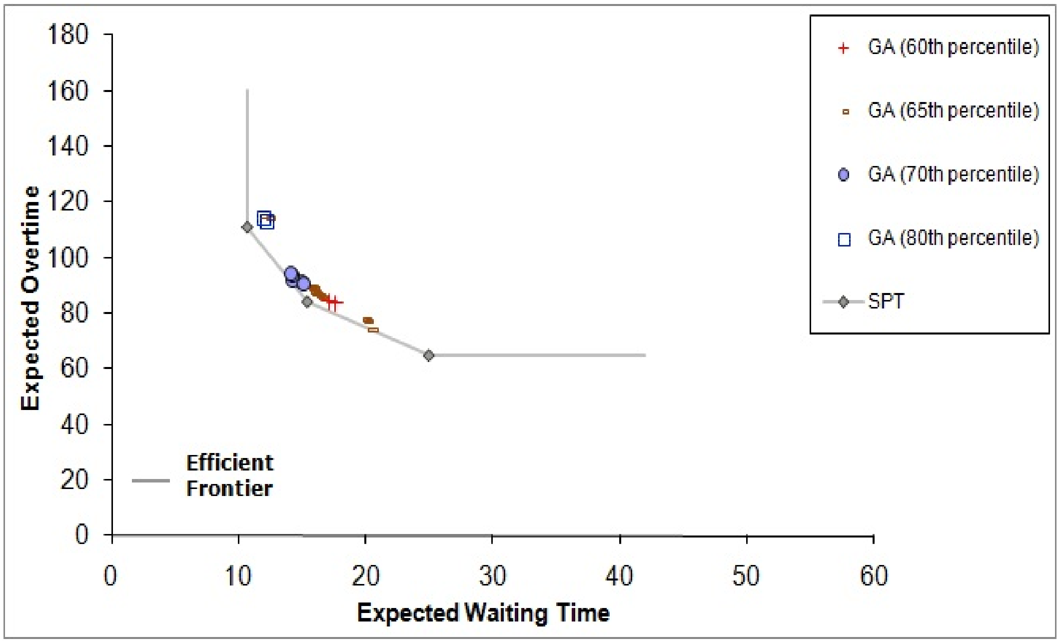

Figure 5. Comparison of the GA Solutions with the SPT Solutions

The most significant finding in this section is: *The performance of SPT based heuristics is similar to performance of the GA when the day of the procedure is fixed. Because it is much easier to implement in practice, SPT based heuristics are recommended over the GA.*

2.6.3. Optimization of Daily Procedure Mix

To answer the third research question defined in Section 1, the requirement that daily mix be fixed is relaxed (model B). This model provides more flexibility since the procedures are allowed to be assigned to any day within an n-day time window. The time windows are defined as mutually exclusive windows (i.e. the days from 1 to n belong to one window, while the days from (n+1) to (2n) belong to a different window), so the days of surgeries are shifted back and forth while fixing the time window they belong to. In the experiments, $n = 3$ and $n = 5$ are tested. In the case of $n = 3$, for example, if the original day of the procedure was Wednesday of the first week, then it can be reassigned to Monday, Tuesday or Wednesday of the first week. On the other hand, if the procedure day was originally set as Friday of the first week, then it can be moved to Thursday or Friday of the first week, or Monday of the second week. The solution space for $n = 5$ corresponds to allowing procedures to be moved within a given week (this is reasonable because procedures scheduled in the OPC are elective). Furthermore, it is consistent with some surgery scheduling practices where scheduling is executed in two steps; first by setting the week of surgery, and afterwards setting the specific times (Gupta 2007).

Figure 6 compares the (near) Pareto optimal sets of GA solutions for $n = 1$, 3, 5. Figure 6 illustrates that reorganizing procedures among days (e.g. $n = 3$ or $n = 5$) considerably improves the two criteria. The main reason for the realization of such an improvement is that the variation of the surgical load among days is better balanced in schedules obtained this way. Besides, the shares of procedure groups using an OR in a given day are now better set due to the flexibility of modifying procedure days. When the procedure mixes among days can be varied, some surgeries that would otherwise have induced overtime can then be assigned to another day where the OR utilization is lower. In Figure 6, similarity is observed between the (near) Pareto optimal sets for $n = 3$ and $n = 5$, i.e. two sets are very close to each other. This indicates the three-day time window is sufficient to balance the surgical load among days.

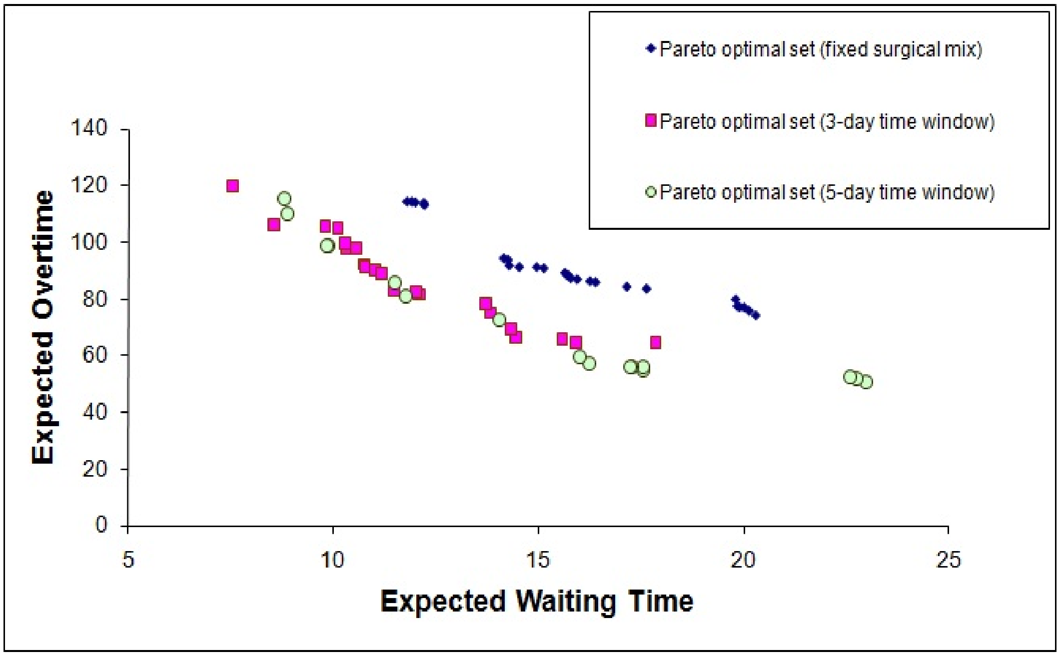

Figure 6. Comparison of Solution Values for Different (Near) Pareto Optimal Set of Solutions of GA for Different Configurations

The most essential finding to be reemphasized is that *controlling surgical mixes among days may help achieve significant improvements in expected patient waiting time and expected surgical suite overtime; a time window of 3 days appears to be sufficient to achieve the benefits.*

2.7. Conclusions

OPCs require the coordination of many activities, including patient check-in, intake, surgical procedure, and recovery. In this chapter, easy-to-implement heuristics are developed for scheduling of an OPC at a large medical center, first. Then, the performance of these heuristics are compared to a GA-based approach. The impact of varying the surgical mix among days is illustrated using the GA. Following are the most significant general insights of this study:

1. Simple heuristics can improve actual schedules used in practice for an OPC. Job hedging may be used to decrease patient waiting times at the expense of increasing surgical suite overtime. Furthermore, the level of trade-off between the patient waiting time and surgical suite overtime due to the increase in job hedging level varies as the heuristic used for sequencing the surgeries changes. Among the sequencing heuristics, LPT (Longest Processing Time First) causes high expected overtime, and should be avoided, while SPT (Shortest Processing Time First) performs quite well.

2. Expending greater computational effort with a more sophisticated GA based method under a restricted environment (no control over daily procedure mix) does not achieve substantial additional improvements. Due to its easy-to-implement nature SPT should be favored over the GA.

3. Controlling daily procedure mix may achieve substantial improvements in performance, though there are diminishing returns as the time window for moving surgeries is increased.

In this study, the schedules are evaluated using a comprehensive model of an OPC and analyze the patient flow through the units (i.e. intake rooms, ORs, recovery rooms). However, since ORs are the major bottlenecks in my model, only the durations of the surgical procedures are considered and the other resources (e.g.

mobile and specialized equipment, materials, nurses, nurse anesthetists, and other human resources) are not considered while designing the surgery schedules. As a future research direction, the plan is to examine the potential benefits of more complicated scheduling techniques considering the impact of other resource types into the schedule efficiency.

CHAPTER 3

A MULTI-STAGE STOCHASTIC PROGRAMMING MODEL FOR SURGERY PLANNING

3.1. Introduction

The rising cost of health care delivery has put pressure on health care managers to reduce expenditures. Since OR costs form a large portion of the total hospital costs (Gul et al. 2010), substantial cost reductions might be achieved through more efficient management of ORs. Typically, a two-phase process is followed to plan for a day of surgery. In the first phase, surgeries are assigned to days and ORs. This is usually done a few weeks prior to the day of surgery. In the second phase, the surgeries are sequenced and patient appointment times are set, generally one day prior to the day of surgery. Some examples of the studies that investigate sequencing surgeries and setting appointment times are Gul et al. (2010), Cardoen et al. (2010), Denton et al. (2007), Denton and Gupta (2003).

Roland et al. (2010) analyzed both phases under a single model. Their model allocates surgeries to days and ORs over a planning horizon, and then assigns surgeries to particular time intervals while considering the staff and medical equipment availability to minimize fixed OR opening costs and overtime costs.

In this chapter, the first phase of the surgery planning and scheduling process is studied. The remainder of this chapter is organized as follows. In the next section, a brief literature review of surgery planning studies is presented. In Section 3.3., the decision making process is described and the multi-stage stochastic mixed integer programming model is formulated. In Section 3.4., progressive hedging algorithm is discussed. In Section 3.5., the experimental results are presented. Finally, the concluding remarks are given in Section 3.6..

3.2. Literature Review

Following is a literature review on surgery planning studies. The literature review is divided into three categories of research. The articles in the first category discuss deterministic models for OR planning. The second category includes papers which consider uncertainties related to the procedure durations, but neglect the demand uncertainty. Since the demand over the planning period is assumed to be known in these studies, the designed models are of a static nature, i.e., all decisions are given at the beginning of the planning period in the model. Papers in the third category relax the assumption of deterministic surgery demands, and thus study the dynamic planning problem, where scheduling decisions are taken, and also revised at each stage of the planning period.

Many of the earliest articles focused on deterministic models. Guinet and Chaabane (2003) used a two-phase approach based on weekly OR planning. Their model assigns surgeries to ORs and particular time blocks of each day over a finite planning horizon. The objective is to minimize the patient's indirect waiting time, i.e., the time between the procedure and hospitalization date, and OR overtime. Their model also considers equipment constraints and availability of surgeons.

Other deterministic models are analyzed and solved in some recent articles. For example, Fei et al. (2008, 2009, 2010) modeled the problem of the optimal assignment of surgeries to ORs and days to minimize OR overtime and maximize

OR utilization using an integer program. They formulated the problem as a set partitioning problem model and applied a column generation based heuristic to solve the model.

Many recent articles have used stochastic optimization models and methodologies for surgery planning. First, the articles on static stochastic models are summarized. Note that the deterministic demand for elective surgeries implies a static nature to these models. Consequently, the models do not include decisions given through different stages to revise the plan for elective surgeries. In other words, surgery cancellations or reassignments are not considered, because the demands for elective surgeries are assumed to be deterministic.

Min and Yih (2010) modeled the problem of allocating surgeries to the blocks reserved for different surgery specialties. They formulated the problem as a two-stage stochastic mixed integer program and used a sample average approximation method to solve the problem. Their model also considers the availability of the Intensive Care Unit (ICU) beds during the block assignment phase. The length of stay in the ICU bed and surgery durations are the stochastic parameters in their model. The objective function minimizes patient priority based waiting costs and OR overtime costs. Lamiri et al. (2008a) solved the problem of assigning elective surgeries to periods over a planning horizon while considering the impact of uncertainty related to emergency case arrivals. They first modeled the problem as a stochastic combinatorial optimization problem and then provided a reformulation in the form of a sample average approximation problem. The authors considered expected overtime costs and patient related costs as the performance measures. The surgery durations are assumed to be deterministic in the study.

Lamiri et al. (2008b) extended the model in Lamiri et al. (2008a) by considering the allocation of surgeries to ORs. Lamiri et al. (2009) proposed several heuristics to solve the same problem in Lamiri et al. (2008a) and compared the heuristics' performance with the performance of a Monte Carlo optimization method. Hans et al. (2008) also solved a stochastic OR-to-day allocation problem, where the stochasticity exists due to the uncertainty of the surgery durations. Their objective is to minimize the planned slack time reserved in the ORs each day which can be used by surgeries running longer than expected. The authors consider the trade-off between the OR utilization and OR overtime. The authors found that the surgeries having similar duration variability should be clustered together and assigned to the same OR-day.

There are only a few papers in the literature that consider revisions to daily surgery lists due to uncertainty in surgery durations. Gerchak et al. (1996) modeled a planning problem as a stochastic dynamic program. The decision process in their study was defined as follows: Each day, new requests for elective and emergency surgeries arise. Surgeries are scheduled to the current or future days and previously scheduled surgeries may be canceled. The objectives include maximizing the expected profit gained by scheduling elective cases, and minimizing the expected overtime and surgery cancellation costs.

Zonderland et al. (2010) also considered a dynamic decision process where the days are assigned to blocks of surgeries at the beginning of every week for a variety of urgency levels. The different urgency levels include elective surgeries as well as the semi-urgent surgeries that must be scheduled within one or two weeks. Based on a Markov decision model, the authors provided a planning guideline by taking the costs related to the OR idle time, OR overtime, and cancellation of elective surgeries into consideration.

The work presented in this chapter differs from the studies in the first and second category due to the stochastic dynamic setting for scheduling the surgeries. Furthermore, this study has contributions different from Gerchak et al. (1996) and Zonderland et al. (2010), which also investigate a similar decision-making process, in the following senses. Gerchak et al. (1996) allows same-day scheduling after a request arises for a surgery, however this is not a very realistic representation of most surgical practices. The surgery durations generated in their model are independent from each other and identically distributed. The authors also acknowledge that the restrictions on the use of probability distributions for this purpose would have a major impact if OR allocations are also considered in the model. However, they did not consider the OR allocations and scheduling complexities related to this issue. On the other hand, this study takes OR allocations into account and do not put limitations on the type of probability distributions. Zonderland et al. (2010) study a higher level planning perspective, because they do not consider the assignment of individual surgeries to days, but rather reserve time slots for elective or semi-urgent surgeries each day. Thus, for example, they do not make distinction between different types of elective surgeries. Furthermore, they assume the surgery requests arise according to a Poisson process, however this is not a justified assumption. Thus, they make strict assumptions on the surgery duration and demand values, while the model in this study does not require such important limitations.

3.3. Problem Description

The model formulated and discussed in the remainder of this chapter considers the decisions for the dynamic allocation of surgeries to operating rooms (ORs) over a finite planning horizon (see Figure 7). The problem is formulated as a multi-stage stochastic mixed integer program. At each stage (day), newly requested surgeries are scheduled to future days; furthermore, some previously scheduled surgeries may be canceled and subsequently rescheduled to a future stage. In addition to assigning each surgery a day, an available OR is also assigned.

At the beginning of each day, it is assumed that random durations for surgeries are observed at the start of the day. Thus, after the final schedule is determined for each day, the cumulative duration of the surgeries assigned to the ORs, total amount of OR overtime, and cancellations are determined.

Total expected OR overtime and postponement costs are the two performance measures considered. To reduce overtime, surgeries might be canceled and rescheduled into future. However, the number of cancellations must be limited, because it results in surgery cancellation and postponement costs. The model includes a per day cancellation and postponement cost associated with the surgeries. Furthermore, there exists a time window within which each surgery

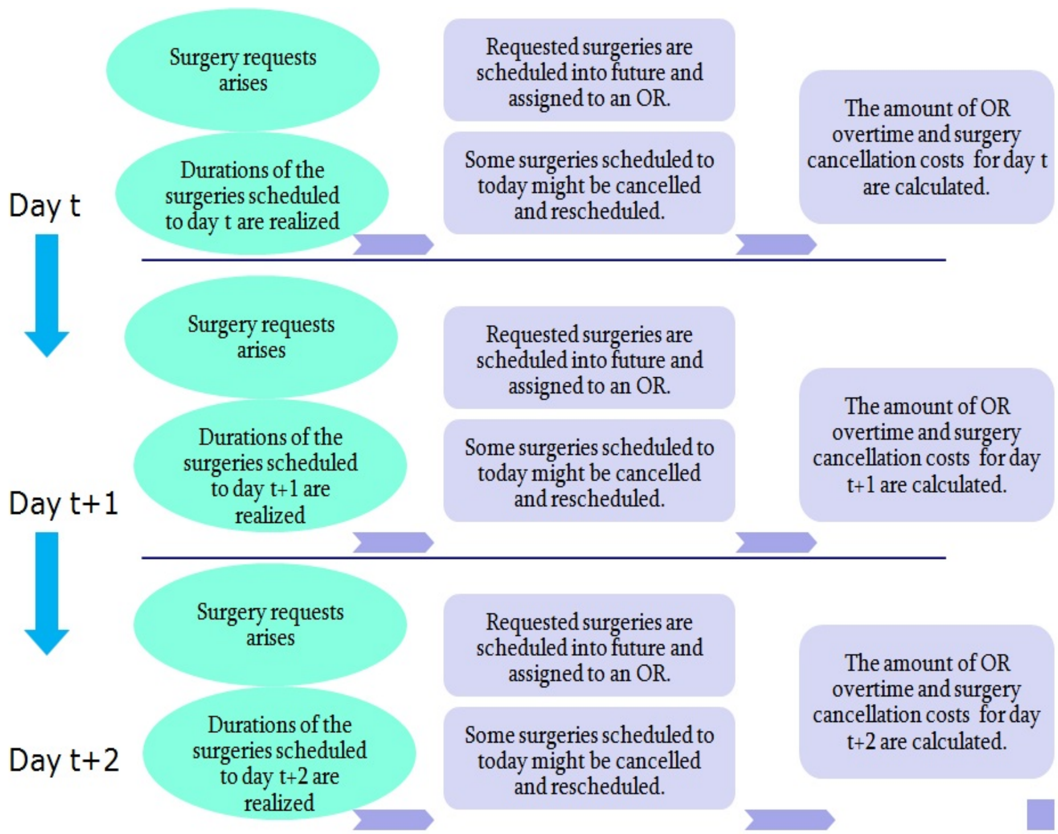

Figure 7. The Pattern Followed While Taking Surgery Scheduling Decisions During a 3-Day Length of Planning Period

must be completed. In other words, there exists a deadline for a surgery and the surgery can not be rescheduled to a day beyond the deadline. The decisions are taken at each stage during the planning horizon. Surgeries may also be scheduled to an additional dummy period at the end of the planning horizon.

An important focus of this study is the cancellation of surgeries, because it is an important decision that significantly influences patient welfare. For example, one study found that the percentage of the canceled surgeries range between 5 % - 20 % across institutions in the US (Argo et al. 2009).

Cancellations result in prolonged hospital stays, delayed perioperative treatments, and repeated preoperative tests and treatments. Cancellations have been found to incur a cost of \$ 1700 - \$ 2000 per case (Argo et al. 2009). A recent study indicates that as much as 50 % of cancellations can be prevented (Gillen et al. 2009). To achieve this, it is necessary to design surgery schedules that carefully consider the uncertainty related to the future. The objective of my model is to minimize the daily cost of overtime and cancellations at a given stage, and the expected daily costs of overtime and cancellations over the following stages in the planning horizon. The following are the indices used in the multi-stage stochastic mixed integer programming model (MSSMIPM):

Indices:

i: surgery index

l,t,u: stage index

j: OR index

ω^t: scenario index for stage *t*

Deterministic Parameters:

$$\lambda_{ij} = \begin{cases} 1 & \text{if there is no equipment constraint restricting the assignment} \\ & \text{of surgery i to OR j;} \\ 0 & \text{otherwise.} \end{cases}$$

g_i = lead time (number of days between the earliest day the surgery can be assigned to and the day the request arises) for scheduling surgery *i*.

h_i = length of time window (number of days between the earliest day and the latest day that the surgery can be assigned to) for scheduling surgery *i*.

$$a_{ijt} = \begin{cases} 1 & \text{if surgery i was already assigned to day t and OR j before the decision} \\ & \text{process starts;} \\ 0 & \text{otherwise.} \end{cases}$$

$P_j^t =$ capacity (in terms of minutes) of OR *j* at stage *t*

$c^i =$ cancellation cost per day for surgery *i*

$c^o =$ OR overtime cost per minute

$N =$ a large number which is higher than the number of surgeries to be requested over a planning horizon

$O =$ number of ORs

$H =$ length of planning horizon for scheduling surgeries

Random Parameters

$d_i(\omega^t) =$ random duration of surgery *i* under scenario ω^t

$$s_i^t(\omega^t) = \begin{cases} 1 & \text{if a request for surgery i arises at stage t under scenario } \omega^t; \\ 0 & \text{otherwise.} \end{cases}$$

$$p_{iu}(\omega^t) = \begin{cases} 1 & \text{if surgery } i \text{ can be assigned to day } u \text{ at stage } t \text{ under scenario } \omega^t \text{ ;} \\ 0 & \text{otherwise.} \end{cases}$$

t^{th} *Stage Decision Variables*

$$x_{iju}^t(\omega^t) = \begin{cases} 1 & \text{if surgery } i \text{ is assigned to OR } j \text{ and day } u \text{ at stage } t \text{ under scenario } \omega^t; \\ 0 & \text{otherwise,} \end{cases}$$

$$\sigma_{ij}^t(\omega^t) = \begin{cases} 1 & \text{if surgery } i \text{ from OR } j \text{ is canceled at stage } t \text{ under scenario } \omega^t; \\ 0 & \text{otherwise,} \end{cases}$$

$k_i^t(\omega^t)$ = number of days surgery *i* is postponed when surgery *i* is canceled at stage *t* under scenario ω^t

$o_j^t(\omega^t)$ = overtime for OR *j* observed at stage *t*

It is assumed in the first stage that the values for d_i, s_i^1, p_{iu} are known in advance, therefore only one scenario (i.e. $\omega^1 = 1$) is assumed to be observed in stage 1. Next, the formulation of the problem is presented.

$$\begin{aligned} \min \sum_{i=1}^{N} c^i k_i^1(\omega^1) + \sum_{j=1}^{O} c^o o_j^1(\omega^1) + E_{\xi^2}[\min \sum_{i=1}^{N} c^i k_i^2(\omega^2) + \sum_{j=1}^{O} c^o o_j^2(\omega^2) + ... \\ + E_{\xi^t}[\min \sum_{i=1}^{N} c^i k_i^t(\omega^t) + \sum_{j=1}^{O} c^o o_j^t(\omega^t) + ... \qquad (3.3.1) \\ + E_{\xi^H}[\min \sum_{i=1}^{N} c^i k_i^H(\omega^H) + \sum_{j=1}^{O} c^o o_j^H(\omega^H)]...]] \end{aligned}$$

s.t.

$$\sum_{u=2}^{H+1} \sum_{j=1}^{O} x_{iju}^1(\omega^1) = s_i^1(\omega^1) + \sum_{j=1}^{O} \sigma_{ij}^1(\omega^1) \qquad \forall\, i \qquad (3.3.2)$$

$$x_{iju}^1(\omega^1) \le \lambda_{ij} p_{iu}(\omega^1) \qquad \forall\, i, j, u > 1 \qquad (3.3.3)$$

$$\sigma_{ij}^1(\omega^1) - a_{ij1} \le 0 \qquad \forall\, i, j \qquad (3.3.4)$$

$$\sum_{j=1}^{O} \sigma_{ij}^1(\omega^1) - p_{i2}(\omega^1) \le 0 \qquad \forall\, i \qquad (3.3.5)$$

$$k_i^1(\omega^1) = \sum_{u=2}^{H+1} u(\sum_{j=1}^{O} x_{iju}^1(\omega^1) - s_i^1(\omega^1)) - \sum_{j=1}^{O} \sigma_{ij}^1(\omega^1) \qquad \forall\, i \qquad (3.3.6)$$

$$\sum_{i=1}^{N} d_i(\omega^1)(a_{ij1} - \sigma_{ij}^1(\omega^1)) - o_j^1(\omega^1) \le P_j^1 \qquad \forall\, j \qquad (3.3.7)$$

$$\sum_{u=3}^{H+1}\sum_{j=1}^{O} x_{iju}^2(\omega^2) = s_i^2(\omega^2) + \sum_{j=1}^{O}\sigma_{ij}^2(\omega^2) \qquad \forall\, i, \omega^2 \tag{3.3.8}$$

$$x_{iju}^2(\omega^2) \leq \lambda_{ij} p_{iu}(\omega^2) \qquad \forall\, i, j, u > 2, \omega^2 \tag{3.3.9}$$

$$\sigma_{ij}^2(\omega^2) - a_{ij2} - x_{ij2}^1(\omega^2) \leq 0 \qquad \forall\, i, j, \omega^2 \tag{3.3.10}$$

$$\sum_{j=1}^{O}\sigma_{ij}^2(\omega^2) - p_{i3}(\omega^2) \leq 0 \qquad \forall\, i, \omega^2 \tag{3.3.11}$$

$$k_i^2(\omega^2) = \sum_{u=3}^{H+1} u(\sum_{j=1}^{O} x_{iju}^2(\omega^2) - s_i^2(\omega^2)) - 2\sum_{j=1}^{O}\sigma_{ij}^2(\omega^2) \qquad \forall\, i, \omega^2 \tag{3.3.12}$$

$$\sum_{i=1}^{N} d_i(\omega^2)(a_{ij1} + x_{ij2}^1(\omega^2) - \sigma_{ij}^2(\omega^2)) - o_j^2(\omega^2) \leq P_j^2 \qquad \forall\, j, \omega^2 \tag{3.3.13}$$

$$\ddots$$

$$\sum_{u=t+1}^{H+1}\sum_{j=1}^{O} x_{iju}^t(\omega^t) = s_i^t(\omega^t) + \sum_{j=1}^{O}\sigma_{ij}^t(\omega^t) \qquad \forall\, i, \omega^t \tag{3.3.14}$$

$$x_{iju}^t(\omega^t) \leq \lambda_{ij} p_{iu}(\omega^t) \qquad \forall\, i, j, u = t+1, .., H, \omega^t \tag{3.3.15}$$

$$\sigma_{ij}^t(\omega^t) - a_{ijt} - \sum_{l=1}^{t-1} x_{ijt}^l(\omega^t) \leq 0 \qquad \forall\, i, j, \omega^t \tag{3.3.16}$$

$$\sum_{j=1}^{O}\sigma_{ij}^t(\omega^t) - p_{it+1}(\omega^t) \leq 0 \qquad \forall\, i, \omega^t \tag{3.3.17}$$

$$k_i^t(\omega^t) = \sum_{u=t+1}^{H+1} u(\sum_{j=1}^{O} x_{iju}^t(\omega^t) - s_i^t(\omega^t)) - t\sum_{j=1}^{O}\sigma_{ij}^t(\omega^t) \qquad \forall\, i, \omega^t \tag{3.3.18}$$

$$\sum_{i=1}^{N} d_i(\omega^t)(a_{ijt} + \sum_{l=1}^{t-1} x_{ijt}^l(\omega^t) - \sigma_{ij}^t(\omega^t)) - o_j^t(\omega^t) \leq P_j^t \qquad \forall\, j, \omega^t \tag{3.3.19}$$

$$\ddots$$

$$\sigma_{ij}^H(\omega^H) - a_{ijH} - \sum_{l=1}^{H-1} x_{ijH}^l(\omega^H) \leq 0 \qquad \forall\, i, j, \omega^H \tag{3.3.20}$$

$$\sum_{j=1}^{O} \sigma_{ij}^{H}(\omega^{H}) - p_{iH+1}(\omega^{H}) \leq 0 \qquad \forall\, i, \omega^{H} \quad (3.3.21)$$

$$k_i^{H}(\omega^{H}) = \sum_{j=1}^{O} \sigma_{ij}^{H}(\omega^{H}) \qquad \forall\, i, \omega^{H} \quad (3.3.22)$$

$$\sum_{i=1}^{N} d_i(\omega^{H})(a_{ijH}^{0} + \sum_{l=1}^{H-1} x_{ijH}^{l}(\omega^{H}) - \sigma_{ij}^{H}(\omega^{H}))$$

$$- o_j^{H}(\omega^{H}) \leq P_j^{H} \quad \forall\, j, \omega^{H} \quad (3.3.23)$$

$$x_{iju}^{t}(\omega^{t}), \sigma_{ij}^{t}(\omega^{t}) \in 0,1 \qquad \forall i; j; \omega^{t}; t = 2, ..., H; u = 3, ..., H+1 \quad (3.3.24)$$

$$k_i^{t}(\omega^{t}), o_j^{t}(\omega^{t}) \geq 0 \qquad \forall i; j; \omega^{t}; t = 2, ..., H; u = 3, ..., H+1 \quad (3.3.25)$$

The objective function includes the costs for the first stage and the expected future costs to go for the remaining H-1 stages. Given the scheduling and cancellation decisions taken in the first stage, the expected costs for the rest of the stages are calculated using the nested expected value to go.

The constraint set has a block diagonal structure. There are H blocks of constraints as well as the nonnegativity and binary restrictions on the decision variables. Each of the first H-1 blocks contain six types of constraints, while the last block has only four types. The constraint blocks for stages 1,2,..., H-1 impose the same types of restrictions into the solution space. The block for stage t that is defined by ((3.3.14)-(3.3.19)) can be regarded as a generic block representing each of these blocks. Constraint (3.3.14) ensures that a surgery must be assigned to an OR in one of the subsequent days after day t if a request arises for this surgery on day t or if the surgery was already assigned to an OR on day t but now appears cancelled. Constraint (3.3.15) ensures that the surgery can be assigned to an OR and some day that follows day t provided that there is no restriction for these assignments. When a request arises for a surgery, it must be scheduled within the allowable time window (h_i) for performing the surgery and at least g_i stages into the future. A restriction on the assignment of a surgery to an OR might also exist, defined by constraint (3.3.15), if the OR does not have all equipment necessary for the surgery. Constraints (3.3.16) and (3.3.17) provide that a surgery in an OR can be cancelled on day t if it was previously assigned to this OR and day; and if it is possible to assign the surgery, at least, to the following day. Note that the cancellation decision for a surgery can be given more than once over the planning period. Constraint (3.3.18) measures the number of days that a surgery is postponed if a cancellation decision is taken for this surgery on day t. Note that a surgery can be canceled more than once over the planning period. Constraint (3.3.19) calculates the overtime for an OR by considering the surgeries scheduled to day t but not cancelled.

The constraint block for stage H differs from the ones discussed above. At this stage, there does not exist any surgery request since this is the last day of the planning horizon and same-day scheduling decisions are not allowed in the model. Therefore, the model may only give a cancellation decision on this day. Due to this fact, the constraint set is more compact than the ones of the previous stages. Constraints ((3.3.20)-(3.3.21)) define the limits on the decision variables related to the cancellation decisions given on day H. Constraints (3.3.22) and (3.3.23) are placed to calculate the number of cancellations and the amount of OR overtime on day

H, respectively. Constraints (3.3.24) and (3.3.25) define binary and nonnegativity restrictions on the decision variables.

The structure of the formulated problem reveals that the problem is NP-hard. An instance of this problem, where the model has only one scenario corresponds to the well known bin packing problem. Since the bin packing problem is NP-hard, the dynamic multi-period OR planning problem is also NP-hard.

3.4. Solution Methodology

The problem is solved using the progressive hedging algorithm (PHA) proposed by Rockafellar and Wets (1991). The PHA proceeds by applying scenario decomposition to the overall problem iteratively, solving the resulting individual scenario subproblems, and finally aggregating individual scenario solutions. Although the PHA is guaranteed to converge to a global optimal solution asymptotically in the convex case (Rockafellar and Wets 1991), it may converge to only a local optimal solution in this case, because the problem has a non-convex nature due to the integer variables at all stages.

The PHA has been applied in several application areas since the time it was proposed by Rockafellar and Wets (1991) (for example, see Mulvey and Vladimirou (1992) for a financial planning application; Helgason and Wallace (1991) for fisheries management application; Santos et al. (2009) for hydrothermal systems operation planning application). The reader is referred to Wallace and Helgason (1991), Watson et al. (2010) for suggestions about the algorithm implementation techniques.

Many authors of PHA based studies analyzed the structural properties of the algorithm, discussed and proposed internal tactics of improving the overall performance of the PHA based on the special structure of the problem of interest (Mulvey and Vladimirou 1991b,a, Wallace and Helgason 1991, Hvattum and Lokketangen 2009, Watson et al. 2010, Crainic et al. 2010). Background information on the PHA is given next by illustrating the main steps of the algorithm.

3.4.1. Problem Reformulation

The problem is reformulated to provide an appropriate program for scenario decomposition. The scenario decomposition can be achieved when a constraint block becomes associated with an individual scenario. In MSSMIPM, however, the constraint blocks exist for each stage of the planning period in the problem. Furthermore, the probability of observing a particular scenario realization at a particular stage, ω^t, is conditioned on the scenario realized in the previous stage, ω^{t-1}. Therefore, the current definition of scenario does not allow us to generate a scenario separable model. In the next formulation, also called a deterministic equivalent model (DEM), a new parameter, η, that represents a sequence of consecutive scenarios aggregated over stages (i.e. $\omega^1, \omega^2, ..., \omega^H$) is defined and introduced. This revised formulation helps break the dependencies that prevent having independent constraint blocks.

Figure 8 illustrates how the reorganization of the model definition impacts the scenario tree. Figure 8-(a) and Figure 8-(b), specifically, show how the uncertainty is modeled in the MSSMIPM and DEM, respectively. Each oval node

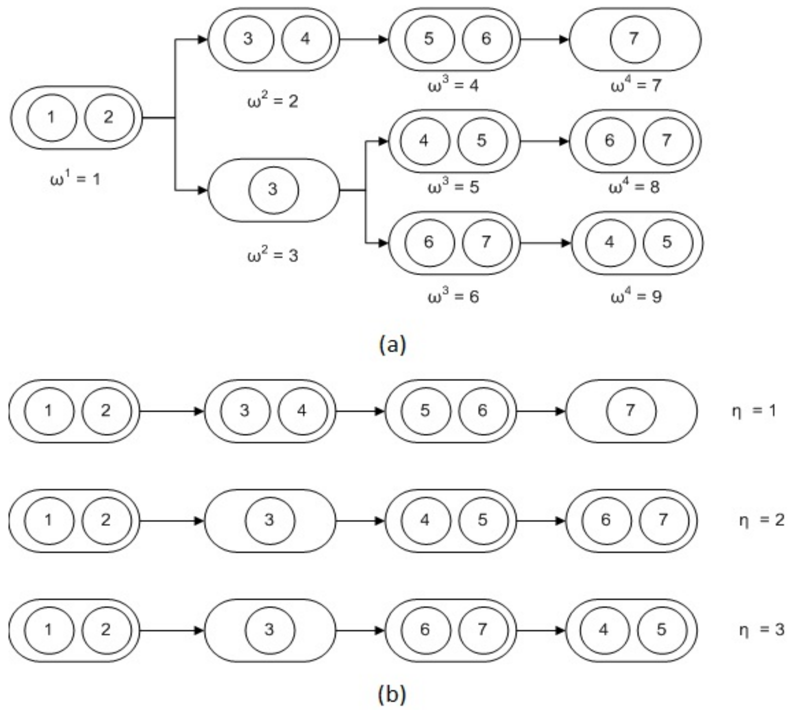

Figure 8. (a) A Scenario Tree Example Illustrating the Surgeries That Might Be Requested at Each Stage over a Four-Day Planning Period (b) The Example in (a) is Shown in Terms of Individual Scenario Sequences

in the scenario tree represents a particular scenario realization, ω^t, at a particular stage *t*. The circle nodes within the oval nodes indicate the surgeries requested at a particular stage under the scenario that the oval node represents. Note that, for simplification purposes, the example in Figure 8 assumes that the uncertainty is based on just the surgery requests (i.e. the surgery durations are deterministic). Figure 8-(a) illustrates that ω^4 varies based on the scenario represented by ω^3. The same relation exists also for ω^1, ω^2 and ω^2, ω^3. On the other hand, Figure 8-(b) illustrates an alternative representation of the scenario tree given in Figure 8-(a) where the individual scenarios observed in the particular stages are aggregated over stages to form three scenario sequences, $\eta = 1, 2, 3$. However, the above redefinition of the scenario tree is not permissible since the solutions found might not be feasible for the overall problem, because they imply decisions that anticipate future uncertain events. The following simple example demonstrates this. Suppose that the solution found for the subproblem for $\eta = 2$ assigns surgery 3 to stage 3 when the request for this surgery arises at stage 2. However, if the subproblem solution for $\eta = 3$ assigns surgery 3 to stage 4 when it is requested at stage 2, then this leads to an infeasible solution for the overall problem. The solutions found for $\eta = 2$ and $\eta = 3$ at stage 2, actually, must be the same, because $\eta = 2$ and $\eta = 3$ share the same history at this stage as can also be seen in Figure 8-(a). Therefore, some constraints should exist in the DEM that would prevent having such infeasible solutions for the overall problem. These constraints in the DEM are called *nonanticipativity constraints*. The nonanticipativity constraints force solutions to satisfy the *nonanticipativity property*. This property is defined as follows: If two scenario sequences, (i.e. $\eta = a, b$), share the same history up to day *t*, the surgery plans created progressively over the planning period should always have the same content until day *t* under the two scenario sequences. In other words, if a scheduling decision is given for a surgery at some stage *l*, where *l* $\leq$ *t* under scenario sequence *a*, the same scheduling decision should be given for the same surgery at the same stage under scenario sequence *b*. A DEM solution satisfying the *nonanticipativity constraint* is also called an *implementable solution* (Rockafellar and Wets 1991).

Next, the additional notation used to formulate the DEM is given:

Additional Indices:

Z: number of scenario sequences

η: scenario sequence index

B(η*,t)*: scenario bundle index of the surgeries considered for scheduling at stage *t* under scenario sequence η

Additional Parameters:

$$s_{i\eta}^{t} = \begin{cases} 1 & \text{if surgery i is requested at stage t under scenario sequence } \eta; \\ 0 & \text{otherwise.} \end{cases}$$

$$p_{i\eta u} = \begin{cases} 1 & \text{if surgery i can be assigned to day u under scenario sequence } \eta; \\ 0 & \text{otherwise.} \end{cases}$$

$d_{i\eta}$ = duration of surgery *i* under scenario sequence η

Pr_{η} = probability of the occurrence of scenario sequence η

Revised Decision Variables:

$$x_{i\eta ju}^{t} = \begin{cases} 1 & \text{if surgery i is assigned to OR j and day u at stage t under} \\ & \text{scenario sequence } \eta; \\ 0 & \text{otherwise,} \end{cases}$$

$$\sigma_{i\eta j}^{t} = \begin{cases} 1 & \text{if surgery i from OR j is canceled at stage t under scenario sequence } \eta; \\ 0 & \text{otherwise,} \end{cases}$$

$k_{i\eta}^{t}$ = number of days surgery *i* is postponed when surgery *i* is canceled at stage *t* under scenario sequence η

$o_{\eta j}^{t}$ = resulting overtime amount for OR *j* on day *t* under scenario sequence η

Additional Decision Variables:

$$x_{iju}^{B(\eta,t)} = \begin{cases} 1 & \text{if surgery i is assigned to day u and OR j at all stage-scenario sequence} \\ & \text{combinations in the bundle, B(}\eta\text{,t), that stage t-scenario } \eta \text{ belongs to;} \\ 0 & \text{otherwise,} \end{cases}$$

The nonanticipativity constraints are also referred to as *bundle constraints* in the context of this study. If the scenario sequences *a* and *b* share the same history up to day *t*, then this indicates they also share the same *scenario bundle* on day *t*: $B(a,t) = B(b,t)$. Thus, the scheduling decisions given on this day are the same among all scenario sequences placed in the same scenario bundle.

Figure 9 illustrates the scenario bundle concept using the example given in Figure 8. The rectangles covering the oval nodes represent the particular scenario bundles that exist in the example. Since all three scenario sequences have the same realization (e.g. $\omega^1 = 1$) at stage 1, $\eta = 1,2,3$ share the same bundle at this stage, thus this yields the following equation: $B(1,1) = B(2,1) = B(3,1) = 1$. The second stage also contains one scenario bundle, because $\eta = 2$ and $\eta = 3$ share the same history by stage 2.

Next, it is shown how the scheduling decisions would be synchronized using the bundle constraints. First, recall that the model would give a scheduling

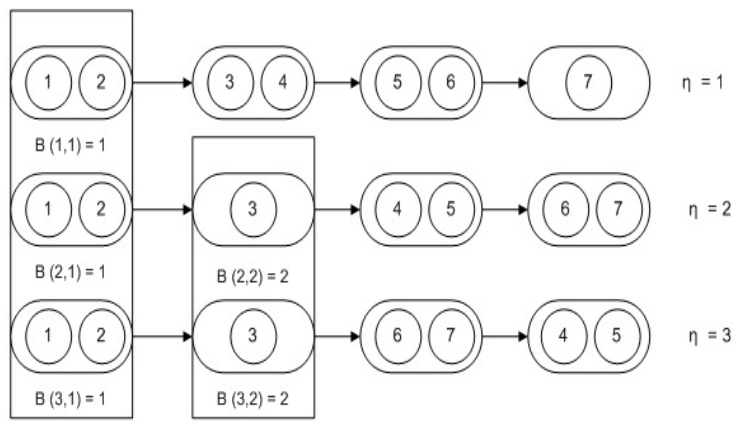

Figure 9. Representation of Scenario Bundles by Rectangles Covering the Scenario Realizations at a Particular Stage.

decision based on two different reasons: (i) request arises for a new surgery; (ii) one of the surgeries scheduled to this stage gets canceled. In stage 1, under all η's, surgeries 1 and 2 should be scheduled into the future due to the reason (i). One can enforce the decision synchronizations using the following chain of equations. The decision variables synchronized among each other are said to form a *decision bundle*.

$$x^1_{i1ju} = x^1_{i2ju} = x^1_{i3ju} \qquad \forall j, u = 2, 3, 4, 5 \text{ and } i = 1, 2$$

Similarly, the equation below can provide synchronization among $\eta = 2, 3$ for scheduling surgery 3 in the second stage:

$$x^2_{32ju} = x^2_{33ju} \qquad \forall j, u = 3, 4, 5$$

Besides, the model might give a scheduling decision for surgeries 1 and 2 due to the reason (ii) in case they were already scheduled to the second stage and gets canceled in this stage. Therefore, these rescheduling decisions for scenario sequences $\eta = 2, 3$ should be bundled using the following equation:

$$x^2_{i2ju} = x^2_{i3ju} \qquad \forall j, u = 3, 4, 5 \text{ and } i = 1, 2$$

To facilitate the generation of a separable program, a new decision variable (i.e. the consensus variable: $x^{B(r,t)}_{iju}$) is defined. Thus, all decision variables in a decision bundle are enforced to be equal to the consensus variable associated with the decision bundle. The deterministic equivalent model (DEM) is formulated next:

$$\min \sum_{\eta=1}^{Z} Pr_\eta(\sum_{t=1}^{H}(\sum_{j=1}^{O} c^o o^t_{\eta j} + \sum_{i=1}^{N} c^i k^t_{i\eta})) \tag{3.4.1}$$

s.t.

$$x^t_{i\eta ju} = x^{B(\eta,t)}_{iju} \qquad \forall i, \eta, j, t, u > t \tag{3.4.2}$$

$$\sum_{u=t+1}^{H+1} \sum_{j=1}^{O} x^t_{i\eta ju} = s^t_{i\eta} + \sum_{j=1}^{O} \sigma^t_{i\eta j} \qquad \forall i, \eta, t \tag{3.4.3}$$

$$x^t_{i\eta ju} \leq \lambda_{ij} p_{i\eta u} \qquad \forall\, i, \eta, j, t, u > t \tag{3.4.4}$$

$$\sigma^t_{i\eta j} - a_{ijt} - \sum_{l=1}^{t-1} x^l_{i\eta jt} \leq 0 \qquad \forall\, i, r, j, t \tag{3.4.5}$$

$$\sum_{j=1}^{O} \sigma^t_{irj} - p_{irt+1} \leq 0 \qquad \forall\, i, r, t \tag{3.4.6}$$

$$k^t_{i\eta} = \sum_{u=t+1}^{H+1} u(\sum_{j=1}^{O} x^t_{i\eta ju} - s^t_{i\eta}) - t \sum_{j=1}^{O} \sigma^t_{i\eta j} \qquad \forall\, i, t, \eta \tag{3.4.7}$$

$$\sum_{i=1}^{N} d_{i\eta}(a^0_{ijt} + \sum_{l=1}^{t-1} x^l_{i\eta jt} - \sigma^t_{i\eta j}) - o^t_{\eta j} \leq P^t_j \qquad \forall\, j, \eta \tag{3.4.8}$$

$$x^t_{i\eta ju}, x^{B(\eta,t)}_{i\eta ju}, \sigma^t_{i\eta j} \in 0, 1 \qquad o^t_{\eta j}, k^t_{i\eta} \geq 0 \qquad \forall i, \eta, j, t, u > t \tag{3.4.9}$$

The objective function (3.4.1) is the weighted sum of the total scenario costs over all scenarios. The total scenario cost is weigted by the probability associated with the scenario, Pr_η. The total cost for a scenario includes the total OR overtime cost and surgery cancellation and postponement cost over all days.

Constraint (3.4.2) is the bundle constraint. Constraints ((3.4.3)-(3.4.9)) have the same structure and meaning as the set ((3.3.14)-(3.3.19)) in MSSMIPM has. Constraint (3.4.3) sets the conditions to be satisfied to give a scheduling decision at a particular stage. Constraint (3.4.4) defines the allowable days and ORs for the assignment of a particular surgery. Constraints (3.4.5) and (3.4.6) together ensure that the cancellation decision for a surgery from an OR can be taken on a day only if the surgery was already assigned to the OR and decision day; and if the surgery is allowed to be postponed, respectively. Constraint (3.4.7) calculates the number of days the surgery is delayed for when a surgery cancellation decision is given on a particular day. Constraint (3.4.8) measures overtime values for each OR, each day. Constraints (3.4.9) define the nonnegativity and binary restrictions on the decision variables.

As previously mentioned, the overall DEM is not a scenario separable formulation due to the bundle constraint. Therefore, an augmented Lagrangian relaxation technique is applied by dualizing the bundle constraint. The relaxed formulation still includes the constraints ((3.4.3)- (3.4.9)) in the constraint set. However, the objective function (3.4.1) is now revised as:

$$\min \sum_{\eta=1}^{Z} Pr_{\eta}(\sum_{t=1}^{H}(\sum_{j=1}^{O} c^{o}o_{\eta j}^{t} + \sum_{i=1}^{N} c^{i}k_{i\eta}^{t}) + \sum_{i=1}^{N}\sum_{t=1}^{H}\sum_{j=1}^{O}\sum_{u=t+1}^{H+1} \mu_{i\eta ju}^{t}(x_{i\eta ju}^{t} - x_{iju}^{B(\eta,t)})$$
$$+ \frac{\rho}{2}\sum_{i=1}^{N}\sum_{t=1}^{H}\sum_{j=1}^{O}\sum_{u=t+1}^{H+1} \|x_{i\eta ju}^{t} - x_{iju}^{B(\eta,t)}\|^{2}) \tag{3.4.10}$$

where $\mu_{i\eta ju}^{t}, \forall i, \eta, t, j, u$ denote the Lagrangian multipliers; ρ is the penalty parameter; and $\|.\|$ is the ordinary Euclidean norm. The additional components in the function (3.4.10) penalizes the violation of the bundle constraint. Since $x_{i\eta ju}^{t}, x_{iju}^{B(\eta,t)} \in 0, 1$, the penalty component in (3.4.10) is rewritten as follows:

$$\|x_{i\eta ju}^{t} - x_{iju}^{B(\eta,t)}\|^{2} = x_{i\eta ju}^{t} - 2x_{i\eta ju}^{t}x_{iju}^{B(\eta,t)} + x_{i\eta ju}^{t} \tag{3.4.11}$$

The next step to make the deterministic equivalent formulation scenario separable requires fixing the consensus variable, $x_{iju}^{B(\eta,t)}$, using the *proximal point method* (Rockafellar 1976). This value can be estimated using the weighted sum calculation:

$$\hat{x}_{iju}^{B(\eta,t)} = \sum_{\eta \in B(\eta,t)}^{Z} \frac{Pr_{\eta}}{\sum_{\eta \in B(\eta,t)} Pr_{\eta}} x_{i\eta ju}^{t} \qquad \forall i, \eta, t, j, u \tag{3.4.12}$$

As can be noted, (3.4.11) does not contain a quadratic term anymore after replacing $x_{iju}^{B(\eta,t)}$ with its estimation, $\hat{x}_{iju}^{B(\eta,t)}$, which facilitates the solution of the subproblems following the scenario decomposition.

Equation (3.4.12) calculates the weighted sum of the individual scheduling decision variables within a decision bundle. The weights are set by normalizing the probability of the scenario associated with a decision variable. Next, the formulation of the separable deterministic equivalent model (SDEM) is given:

$$\min \sum_{\eta=1}^{Z} Pr_\eta (\sum_{t=1}^{H} (\sum_{j=1}^{O} c^o o_{\eta j}^t + \sum_{i=1}^{N} c^i k_{i\eta}^t) + \sum_{i=1}^{N} \sum_{t=1}^{H} \sum_{j=1}^{O} \sum_{u=t+1}^{H+1} \mu_{i\eta ju}^t (x_{i\eta ju}^t - \hat{x}_{iju}^{B(\eta,t)})$$
$$+ \frac{\rho}{2} \sum_{i=1}^{N} \sum_{t=1}^{H} \sum_{j=1}^{O} \sum_{u=t+1}^{H+1} (x_{i\eta ju}^t - 2x_{i\eta ju}^t \hat{x}_{iju}^{B(\eta,t)})) \tag{3.4.13}$$

s.t.

$$\sum_{u=t+1}^{H+1} \sum_{j=1}^{O} x_{i\eta ju}^t = s_{i\eta}^t + \sum_{j=1}^{O} \sigma_{i\eta j}^t \qquad \forall i, \eta, t \tag{3.4.14}$$

$$x_{i\eta ju}^t \leq \lambda_{ij} p_{i\eta u} \qquad \forall\, i, \eta, j, t, u > t \tag{3.4.15}$$

$$\sigma_{i\eta j}^t - a_{ijt} - \sum_{l=1}^{t-1} x_{i\eta jt}^l \leq 0 \qquad \forall\, i, r, j, t \tag{3.4.16}$$

$$\sum_{j=1}^{O} \sigma_{irj}^t - p_{irt+1} \leq 0 \qquad \forall\, i, r, t \tag{3.4.17}$$

$$k_{i\eta}^t = \sum_{u=t+1}^{H+1} u (\sum_{j=1}^{O} x_{i\eta ju}^t - s_{i\eta}^t) - t \sum_{j=1}^{O} \sigma_{i\eta j}^t \qquad \forall\, i, t, \eta \tag{3.4.18}$$

$$\sum_{i=1}^{N} d_{i\eta} (a_{ijt}^0 + \sum_{l=1}^{t-1} x_{i\eta jt}^l - \sigma_{i\eta j}^t) - o_{\eta j}^t \leq P_j^t \qquad \forall\, j, \eta \tag{3.4.19}$$

$$x_{i\eta ju}^t, \sigma_{i\eta j}^t \in 0, 1 \qquad o_{\eta j}^t, k_{i\eta}^t \geq 0 \qquad \forall i, \eta, j, t, u > t \tag{3.4.20}$$

Some of the constant terms in the objective function defined by ((3.4.10)-(3.4.11)) are ignored, because they do not have any impact on the decision variables. This revision yields the objective function (3.4.13) of the SDEM model. Constraints ((3.4.14)-(3.4.19)) define exactly the same feasible space as the constraints ((3.4.3)-(3.4.8)) do. Constraint (3.4.20) defines the integrality and nonnegativity restrictions on the surgery scheduling decision variables.

Note that the consensus variable in DEM is represented by its estimation in SDEM, $\hat{x}_{iju}^{B(\eta,t)}$ which is called as *concensus parameter*. The consensus parameter is also an estimation of the implementable solution. However, there is no guarantee that the estimated implementable solution would be a feasible solution for DEM. If this solution is also feasible in SDEM, then it is also labeled as an *admissible solution*. The target of the PHA is not to find any arbitrary solution that is both admissible and implementable. On the contrary, the algorithm seeks a good solution, preferably the best one, among all admissible and implementable solutions.

The scenario subproblems derived after decomposing SDEM into scenarios are presented next. The mixed integer programming formulation for a particular scenario subproblem model (SSM) is given as:

$$\min \sum_{t=1}^{H}\left(\sum_{j=1}^{O} c^o o_{\eta j}^t + \sum_{i=1}^{N} c^i k_{i\eta}^t\right) + \sum_{i=1}^{N}\sum_{t=1}^{H}\sum_{j=1}^{O}\sum_{u=t+1}^{H+1} \mu_{i\eta ju}^t (x_{i\eta ju}^t - \hat{x}_{iju}^{B(\eta,t)})$$
$$+ \frac{\rho}{2}\sum_{i=1}^{N}\sum_{t=1}^{H}\sum_{j=1}^{O}\sum_{u=t+1}^{H+1} (x_{i\eta ju}^t - 2x_{i\eta ju}^t \hat{x}_{iju}^{B(\eta,t)}) \tag{3.4.21}$$

s.t.

$$\sum_{u=t+1}^{H+1}\sum_{j=1}^{O} x_{i\eta ju}^t = s_{i\eta}^t + \sum_{j=1}^{O} \sigma_{i\eta j}^t \qquad \forall i,t \tag{3.4.22}$$

$$x_{i\eta ju}^t \leq \lambda_{ij} p_{i\eta u} \qquad \forall\, i,j,t,u > t \tag{3.4.23}$$

$$\sigma_{i\eta j}^t - a_{ijt} - \sum_{l=1}^{t-1} x_{i\eta jt}^l \leq 0 \qquad \forall\, i,r,j,t \tag{3.4.24}$$

$$\sum_{j=1}^{O} \sigma_{i\eta j}^t - p_{i\eta t+1} \leq 0 \qquad \forall\, i,r,t \tag{3.4.25}$$

$$k_{i\eta}^t = \sum_{u=t+1}^{H+1} u\left(\sum_{j=1}^{O} x_{i\eta ju}^t - s_{i\eta}^t\right) - t\sum_{j=1}^{O} \sigma_{i\eta j}^t \qquad \forall\, i,t \tag{3.4.26}$$

$$\sum_{i=1}^{N} d_{i\eta}(a_{ijt} + \sum_{l=1}^{t-1} x^{l}_{i\eta jt} - \sigma^{t}_{i\eta j}) - o^{t}_{\eta j} \leq P^{t}_{j} \qquad \forall\, j \tag{3.4.27}$$

$$x^{t}_{i\eta ju}, \sigma^{t}_{i\eta j} \in 0,1 \qquad o^{t}_{\eta j}, k^{t}_{i\eta} \geq 0 \qquad \forall i, j, t, u > t \tag{3.4.28}$$

The objective function (3.4.21) corresponds to one of the scenario costs which are aggregated in the objective function (3.4.13) of the SDEM. Constraint set ((3.4.22)-(3.4.28)) is also a subset of the constraint set ((3.4.14)-(3.4.20)) which should be satisfied for all scenarios rather than only for one scenario.

3.4.2. Progressive Hedging Algorithm

The SSM is utilized as the progressive hedging algorithm (PHA) is applied to solve the SDEM. Let *k* denote the index for the iteration number of the PHA, then the general steps of the PHA are stated as follows:

1. Initialize the algorithm. Set $k = 0, \rho = 0, \mu^{t(k)}_{i\eta ju} = 0\ \forall i, \eta, t, j, u.$

2. Solve the SSM for each scenario η to obtain $x^{t(k)}_{i\eta ju}\ \ \forall i, \eta, t, j, u.$ Next, calculate the consensus parameter $\hat{x}^{B(\eta,t)}_{iju}\ \forall i, B(\eta, t), j, u.$

3. Adjust the common penalty parameter and the Lagrangian multipliers associated with each bundle constraint of the SDEM.

$$3.a.\ \rho^{(k+1)} = \begin{cases} \rho_0 & \text{if k=0;} \\ \alpha\rho^{(k)} & \text{otherwise,} \end{cases}$$

where ρ_0 is some initial value and α is some constant value.

$$3.b.\ \mu^{t(k+1)}_{i\eta ju} = \mu^{t(k)}_{i\eta ju} + \rho^{(k)}(x^{t(k)}_{i\eta ju} - \hat{x}^{B(\eta,t)}_{iju})$$

4. Check the stopping criterion. If all bundle constraints (3.4.2) are satisfied, then the algorithm terminates. Otherwise, let $k = k + 1$ and go to 2.

The ideal condition for the termination of the PHA is when all the bundle constraints are satisfied. However, in practice this may not be achieved in a reasonable amount of time. Instead, a tolerance level, ϵ, is set and the algorithm is terminated when the dual convergence is approximately satisfied.

$$\sum_{\eta=1}^{Z} Pr_{\eta} \sum_{i=1}^{N} \sum_{t=1}^{H} \sum_{j=1}^{O} \sum_{u=t+1}^{H+1} |x_{i\eta ju}^{t(k)} - \hat{x}_{iju}^{B(\eta,t)(k)}| \leq \epsilon$$

The same termination criterion is also used by Takriti and Birge (2000). It can be interpreted as the PHA terminating when the level of the bundle constraint violation is sufficiently low.

3.5. Case Study

This section introduces a particular model instance used to test the PHA. The example illustrates what conditions motivate the cancellation of a surgery. Table 4 presents the values set for the important input types characterizing the test instance. The length of the surgery planning period is selected as four days. Also, it is assumed there is only one OR open over the course of the planning horizon. The OR overtime cost per minute, c^o, is the same for all ORs. The daily capacity of the OR allows only one surgery to contribute to the OR overtime on average each day.

Table 5 shows the cost of cancelling the surgeries, c^i. The selected cancellation costs sometimes favor the cancellation of surgeries when a trade-off between scheduling and cancellation decisions is considered. The purpose of designing such an environment is to observe conflicting scheduling decisions for a surgery among different scenario sequences. The probabilities associated with scenario sequences are listed in Table 6.

Table 4. The Main Characteristics of the Problem Instance

type	# of surgeries	# of ORs	# of stages	# of scenarios
values	16	1	4	10

Table 5. The Surgery Cancellation Costs

c^0	c^1	c^2	c^3	c^4	c^5	c^6	c^7
312	594	712	447	1205	712	1418	447

c^8	c^9	c^{10}	c^{11}	c^{12}	c^{13}	c^{14}	c^{15}
1020	1000	779	775	314	357	1020	712

Table 6. Probability of Scenario Realizations

index	0	1	2	3	4	5	6	7	8	9
probability	0.05	0.08	0.05	0.15	0.1	0.02	0.2	0.15	0.06	0.14

For simplicity, surgery durations are kept constant over the scenario sequences to decrease the size of the problem instance. The durations of the surgeries are given in Table 7. Table 8 lists the surgeries requested at each stage of each scenario sequence. Finally, Table 9 and Table 10 show the lead time and length of the time window necessary for scheduling the surgeries, respectively.

Table 7. Surgery Durations

index	0	1	2	3	4	5	6	7
duration	41	54	39	58	77	59	74	28

index	8	9	10	11	12	13	14	15
duration	83	60	55	28	61	48	53	29

Table 8. Indices of the Surgeries Requested at Each Stage of Each Scenario

Scenario	Stage			
index	1	2	3	4
0	0,6,7,8	1,3,5,9	2,4,10,11	12,13,14,15
1	0,6,7,8	1,3,5,9	12,13,14,15	2,4,10,11
2	0,6,7,8	1,3,5,9	11,13,14,15	2,4,10,12
3	0,6,7,8	1,3,4,10	2,5,9,11	12,13,14,15
4	0,6,7,8	1,3,4,10	12,13,14,15	2,5,9,11
5	1,3,4,6	0,7,8,10	12,13,14,15	2,5,9,11
6	2,5,9,11	1,3,4,10	0,6,7,12	8,13,14,15
7	2,4,10,11	0,6,7,8	1,3,5,9	12,13,14,15
8	12,13,14,15	1,3,5,9	2,4,6,8	0,7,10,11
9	1,3,5,9	0,6,7,8	2,4,10,11	12,13,14,15

Table 9. Lead Times for Scheduling Surgeries

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
lead time	1	1	2	1	1	2	1	1	1	1	2	1	1	1	1	1

The DEM for this particular instance was also coded and solved to optimality to compare with the PHA solutions. This comparison was particularly useful for the verification of the PHA. Both the PHA and the DEM model are coded in

Table 10. Width of Time Windows for Scheduling Surgeries

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
width	2	1	1	1	2	1	1	2	1	2	1	1	1	1	1	1

Microsoft Visual C++ 2005 using CPLEX 11 concert technology. The experiments are conducted on Intel Core i5 PC with processors running at 2.27 GHz and 4 GB memory under Windows XP.

3.5.1. Performance of the PHA

First, the convergence characteristics of the PHA are analyzed. In this particular instance, the PHA is capable of finding the optimal solution. The number of iterations or the computational time it takes to converge to the optimal solution varies based on the values set for the penalty parameter, ρ. Table 11 shows how the convergence rate can be substantially increased by increasing the ρ value.

Although the optimal solution is found for each ρ value tested, this approach is likely to lead to a lower solution quality while solving the large size problem instances. The reason is that the approach enforces the fast convergence of the dual solutions without considering the convergence behavior of the primal variables. Hence, the algorithm finds a solution quickly, but there is no guarantee that this solution is the optimal solution.

The trade-off between the solution quality and convergence rate can be controlled through the use of penalty parameter. Based on the computational time and ρ values shown in Table 11, Figure 10 illustrates that the convergence rate decreases asymptotically as the value of ρ increases.

Table 11. Variation in the PHA Performance Based on the Changes in the Penalty Parameter

ρ	computational time (seconds)	number of iterations
25	83	50
50	20	25
100	7	12
150	4	9
500	2	3
1000	1	2

3.5.2. Analysis of the PHA Solutions

There are three scenario bundles in this instance. The scenario sequences -0,1,2,3,4- at stage one are in the first bundle. The scenario sequences -0,1,2- at stage two are in the second bundle. Finally, the scenario sequences -3,4- at stage two are in the third bundle. Note that one can observe the cancellation decisions only under the scenario sequences that belong to a scenario bundle, because the subproblem solutions for the rest of the scenarios are not required to be synchronized with any other scenario solutions. Thus, the scheduling and cancellation decisions taken under scenario sequences -0,1,2,3,4- at stages 1 and 2 are particularly analyzed.

The PHA solutions are examined at each iteration after setting $\rho = 100$ and running the code. The results indicate that the scheduling decisions for surgeries -4,7,9- are the ones for which the consensus could not be achieved until the last iteration. The analysis revealed that the values taken by the majority of the variables within a decision bundle at the first iteration were also the ones on

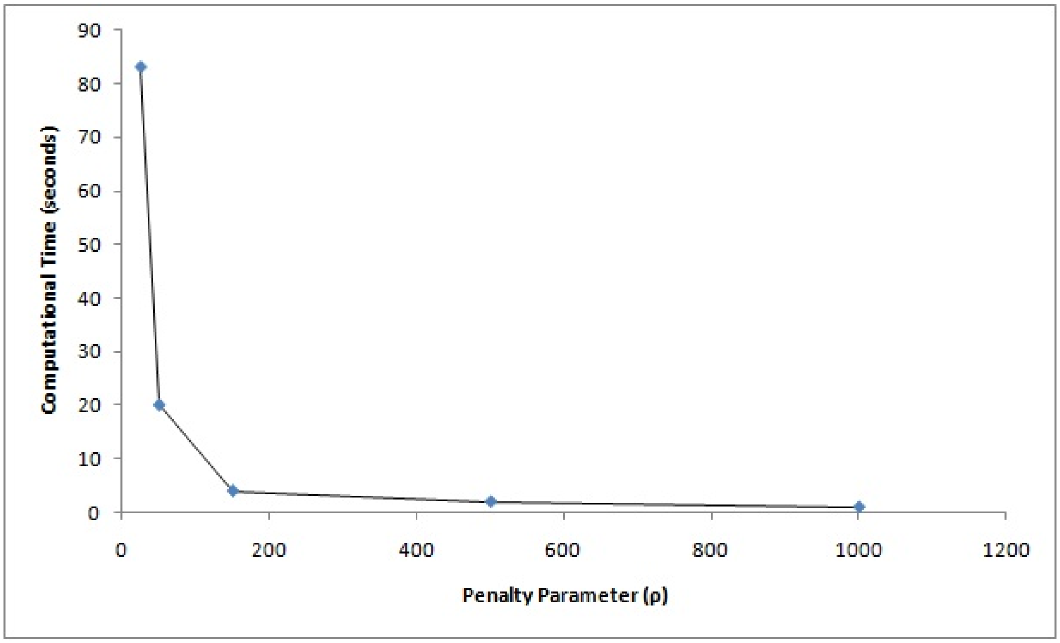

Figure 10. Computational Time as a Function of the Penalty Parameter

which the consensus was achieved in the end. Though there is no guarantee that this can be a valid statement for all instances, the statement is likely to be made if the number of bundles in a problem instance is relatively low, which provides less complicated interactions between the subproblems.

Furthermore, this observation can be utilized to build a heuristic to solve the dynamic multi-period OR planning problem. The solutions found under the majority of the scenario sequences at the first PHA iteration could be the initial solutions for the heuristic to be developed. Next, a better solution can be sought by using a *local improvement heuristic* to find a good solution for the overall problem.

To my knowledge, there is no study in the literature that discusses the impact of the number of bundles or the structure of bundles on PHA performance. Thus, the following questions deserve investigation:

1. What function can determine the relationship between the number of bundles and PHA performance?

2. What is the largest number of bundles that can be allowed to exist in a scenario set for PHA to yield a good solution in a reasonable amount of time?

3. Can the information about the number of bundles that cover a particular scenario sequence be useful to estimate the performance of the PHA?

4. What bundle structures inherent to the problem instance can reveal that solving the DEM is more favorable than using the PHA in terms of the trade-off between the performance and quality?

3.6. Conclusions

This chapter presented a dynamic multi-period OR planning problem where the dynamic nature exists due to the stochastic demand for surgeries. The study is different from many other multi-period OR planning problems (see Literature Review for details), because of the stochastic demand for surgeries. It also has different contributions from the contributions of the studies that built stochastic dynamic models for the problem, because the model constructed does not require strict assumptions related to surgery durations and demands.

A Progressive Hedging Algorithm (PHA) was implemented to solve the problem. The algorithm finds the optimal solutions for the small-size instances. In Chapter 4, the algorithm is tested on larger instances. The PHA solutions will be compared with the optimal solutions found after solving the deterministic equivalent model (DEM). The comparison will also shed light to decide on which direction the improvements should be conducted to enhance the PHA convergence speed and solution quality. There are a number of options for further improvements: (1) Valid inequalities can be derived to facilitate the solution of the subproblems to improve the computational speed. (2) The calibration of the PHA setting (e.g. parameters and checking conditions) can be thoroughly analyzed by utilizing the special structure of the model. (3) A heuristic can be constructed to solve the dynamic multi-period OR planning problem based on the insights gained by the PHA solutions. In addition, the questions posed in the previous section that are related to the impact of the bundles into the PHA performance and solutions deserve major investigation.

CHAPTER 4

A PROGRESSIVE HEDGING ALGORITHM TO SOLVE A DYNAMIC MULTI-PERIOD OPERATING ROOM PLANNING PROBLEM

4.1. Introduction

In this chapter, the progressive hedging algorithm (PHA) proposed in Chapter 3 is extended to take advantage of the special structures of the model formulation and algorithm. The new algorithm proposed in this chapter is referred to as *enhanced progressive hedging algorithm (EPHA)*. The difference between EPHA and PHA is that the EPHA uses novel methods to accelerate the computational performance of the PHA and improve the quality of the PHA solutions. The convergence pattern of the primal and dual variables provides the basis for the proposed penalty parameter update method. The degree of violation of the bundle constraints and decisions taken by the majority of the variables in the decision bundles motivate the Lagrangian multiplier update method. Several other algorithm improvement ideas (subproblem heuristics, warm start, variable locking etc.) are also discussed in this chapter.

The EPHA is developed to solve the surgery planning problem formulated in Chapter 3, i.e. (3.3.1) - (3.3.25). The EPHA also requires the reformulation of the model. Thus, the deterministic equivalent model (DEM), defined by the equations (3.4.1) - (3.4.9), is used to create the separable deterministic equivalent model, i.e. (3.4.13) - (3.4.20). The SDEM is decomposed to generate scenario subproblems, the equations (3.4.21) - (3.4.28). The EPHA follows a sequence of steps structured similarly as the one of the PHA. However, different operators are introduced and used to implement the main steps of the EPHA.

The remainder of this chapter is organized as follows. In the next section, a brief literature review on the progressive hedging algorithm is presented. The review covers studies which discuss and propose penalty update and Lagrangian multiplier methods, subproblem solution methods and several other aspects affecting the PHA performance. In Section 4.3., the EPHA is discussed. In Section 4.4., the experimental results are presented. Finally, the concluding remarks are given in Section 4.5..

4.2. Literature Review

First, the works from the literature related to algorithm design for setting and updating the penalty parameters are discussed. Mulvey and Vladimirou (1991b,a) discussed the trade-off between the selection of high and low values for the penalty parameters and the impact of the problem structure into this selection. They also discussed the benefits of the dynamic penalty adjustment methods. Helgason and Wallace (1991), Listes and Dekker (2005) discussed the sensitivity of the convergence of the PHA to the choice of penalty parameter. Recently, Hvattum and Lokketangen (2009) proposed a method to set the direction while updating the penalty parameters at an iteration of the PHA. They tested the case where there exists parameters for individual nonanticipativity constraints in the model. Watson et al. (2010) also proposed methods to set the penalty parameters for individual nonanticipativity constraints based on a class of resource allocation problems.

Due to the typically large number of subproblems to be solved following the scenario decomposition at each PHA iteration, computational efficiency in subproblems is important. What is more, it is reported in the literature that the PHA is a reasonable heuristic to use if there exists an efficient algorithm to solve the subproblems of a very large scale stochastic mixed integer problem (Watson et al. 2010). Therefore, Takriti et al. (1996) needed to improve existing methods to solve the subproblems of their multi-stage stochastic production planning problem. Furthermore, the heuristics solutions for the subproblems would be sufficient for the convergence of the PHA (Hvattum and Lokketangen 2009, Haugen et al. 2001, Lokketangen and Woodruff 1996, Helgason and Wallace 1991, Barro and Canestrelli 2005, Kall and Wallace 1994). Helgason and Wallace (1991) solved subproblems approximately using a Lagrangian approach and illustrated the convergence of the PHA. Similarly, Takriti and Birge (2000) used a Lagrangian approach to solve the subproblems of a multi-stage loosely coupled mixed integer stochastic programming formulation of a production planning problem. Lokketangen and Woodruff (1996) used a tabu search algorithm to solve the subproblems of a multi-stage stochastic mixed integer problem. Barro and Canestrelli (2005) further decomposed the subproblems of a dynamic portfolio management problem into stages to solve those efficiently.

Another important reason which necessitates the implementation of an efficient solution method on the subproblems is that each subproblem has a quadratic objective function due to the inherent penalty component. Haugen et al. (2001) relaxed the quadratic term in the subproblem objective function and applied a dynamic programming approach to find an optimal solution for the relaxed subproblems. Listes and Dekker (2005) solved the linear relaxation of the subproblems of a robust airline fleet composition problem, which contained integer variables, and used simple rounding procedure to find a feasible solution for the overall problem.

Warm start for the PHA is an important issue related to the role of the Lagrangian multipliers in the overall performance of the algorithm. Mulvey and Vladimirou (1991a), Santos et al. (2009) discussed the importance of the initial estimates for the Lagrangian multipliers and tested simple heuristics to find reasonable initial values.

Alternative termination criteria are proposed in the PHA literature. Watson et al. (2010) used two different criteria for the termination of the PHA according to the class of resource allocation problem they studied. Lokketangen and Woodruff (1996) forces only the integer variables to converge exactly to the consensus parameters. They then set the values of the real variables by solving the deterministic equivalent form of the model having the values of the converged integer variables fixed. However, they acknowledge that terminating the algorithm based on only the integer convergence does not have a teoretical support that indicates this approach is better than the regular methods enforcing the convergence of all variables.

There exists other methods proposed in the literature to provide further improvements on the PHA performance. For example, Mulvey and Vladimirou (1991b) proposed an aggregation scheme for the individual subproblem solutions different from the weighted sum approach. However, their method is designed according to a special case of the stochastic network problems. Based on a similar idea, Hvattum and Lokketangen (2009) picked one of the subproblem solutions rather than taking weighted average of the solutions while updating the Lagrangian multipliers. The same authors also proposed a heuristic that is applied after the termination of the PHA to convert the inadmissible solutions into admissible ones. Crainic et al. (2010) focused on the inadmissable solutions found at some intermediate iteration of the PHA and proposed a simple method to convert them into admissible solutions. They then used those converted values as the upper bounds of the decision variables.

Another common approach used as an algorithm acceleration scheme is *variable locking* or *variable fixing*. Watson et al. (2010) selected some variables within the model and fixed their values at a certain iteration of the PHA in an attempt to reduce the total amount of variables in the overall model. Particularly, they fixed the variables whose value do not change for a certain number of consecutive iterations. Once the variable is fixed, its value stays constant until the termination of the algorithm. Hvattum and Lokketangen (2009) tested both a partial (some variables are fixed at an iteration) and complete (all variables are fixed) variable locking mechanisms for the same purpose.

Finally, it is well known that in the non-convex case, the PHA is not guaranteed to converge (Takriti and Birge 2000), so Watson et al. (2010) defined some techniques to detect the non-convergence cases.

4.3. Enhanced Progressive Hedging Algorithm

In the following subsections, penalty update and Lagrangian multiplier update methods are proposed, and the EPHA termination criterion is presented. In Section 4.4., experimental results are presented to compare the proposed methods.

4.3.0.1. *Penalty parameter setting and update*

In this study, a constant value for the penalty parameter is first set after conducting some experimental analysis, because finding a good value for the parameter depends on the problem structure and program scaling (Mulvey and Vladimirou 1991b). The experimental analysis is based on the observation of the trade-off between fast convergence to a suboptimal solution (i.e. ρ is too large) and slow convergence to a near optimal solution in the primal feasible space (i.e. ρ is too low). A well designed approach to utilize this trade-off tends to set a low value for the penalty parameter at the initial steps of the PHA and then increases this amount gradually, depending on the convergence rates in the primal and dual spaces.

Next, a method was tested based on the method proposed in Hvattum and Lokketangen (2009). The method compares the convergence rate at iteration *k* with the one in the immediately preceding iteration, $k-1$, and then increases ρ if it appears that the convergence rate in the dual space is decreasing. This leads to a faster convergence to a PHA solution. However, if the convergence rate in the primal space decreases, then this reflects in a decrease in ρ. Let $\Delta_D^{(k)}$ and

$\Delta_P^{(k)}$ are indicators of the convergence rates in the dual space and in the primal space, respectively. Let b index a unique bundle among the ones represented by all $B(\eta, t)$'s, and B represent the total number of unique bundles. Then, equations ((4.3.1) $-$ (4.3.2)) define the penalty update method as follows:

$$\Delta_P^{(k)} = \sum_{i=1}^{N} \sum_{b=1}^{B} \sum_{j=1}^{O} \sum_{u=t+1}^{H+1} (\hat{x}_{iju}^{b(k)} - \hat{x}_{iju}^{b(k-1)})^2 \tag{4.3.1}$$

$$\Delta_D^{(k)} = \sum_{i=1}^{N} \sum_{\eta=1}^{Z} \sum_{t=1}^{H} \sum_{j=1}^{O} \sum_{u=t+1}^{H+1} (x_{i\eta ju}^{t(k)} - \hat{x}_{iju}^{B(\eta,t)(k)})^2 \tag{4.3.2}$$

$$\rho^{(k+1)} = \begin{cases} \delta \rho^{(k)} & \text{if} \Delta_D^{(k)} - \Delta_D^{(k-1)} > 0 \\ \\ \frac{1}{\delta} \rho^{(k)} & \text{if} \Delta_P^{(k)} - \Delta_P^{(k-1)} > 0 \end{cases} \tag{4.3.3}$$

where $\delta > 1$ in (4.3.3) is a fixed multiplier.

4.3.0.2. *Lagrangian multiplier update*

A variant of the method that the basic PHA uses to update the Lagrangian multipliers is used (see Crainic et al. (2010) for a similar approach). Crainic et al. (2010) propose an update method for the coefficients of the variables in the nonanticipativity constraints. They do not consider the penalty component in their overall algorithm, so the algorithm is actually not a PHA, but a Lagrangian heuristic. The EPHA considers their coefficient update method as the Lagrangian multiplier update technique. The purpose of the Lagrangian multiplier update method is to use the knowledge provided by the difference between consensus parameter and the individual scheduling decision variables within the decision bundles. This knowledge might help shorten the computational time it takes for the scheduling decision variable values to converge to the consensus parameter values.

In particular, a steepest ascent method is applied to update the Lagrangian multipliers and the approach is based on the following observation. If $\hat{x}_{iju}^{B(\eta,t)(k)}$ is greater than a constant value that is large enough, then this indicates that the majority of the scenario subproblem solutions within the associated decision bundle dictate the assignment of surgery i to day *u* and OR *j* on day *t*. Suppose that such an assignment is actually done under one of the scenario sequences. Then, the method would keep the Lagrangian multiplier associated with the decision variable that represents this assignment constant. The aim here is to preserve the consensus among the decision variables within the decision bundle. On the other hand, if this assignment is not done under another scenario sequence, then the associated Lagrangian multiplier is decreased, so that the decision variables can have a better chance of reaching consensus. The Lagrangian multipliers are updated based on the same idea when $\hat{x}_{iju}^{B(\eta,t)(k)}$ is less than a constant value. If $x_{i\eta ju}^{t(k)}$ is zero, the associated Lagrangian multiplier is kept constant, otherwise it is increased to approach the consensus condition. Updates are computed as follows:

$$\mu_{i\eta ju}^{t(k+1)} = \begin{cases} \mu_{i\eta ju}^{t(k)} + \rho^{(k)}|(x_{i\eta ju}^{t(k)} - \hat{x}_{iju}^{B(\eta,t)(k)})| & \text{if } |(x_{i\eta ju}^{t(k)} - \hat{x}_{iju}^{B(\eta,t)(k)})| \geq \theta; x_{i\eta ju}^{t(k)} = 1 \\ \mu_{i\eta ju}^{t(k)} - \rho^{(k)}|(x_{i\eta ju}^{t(k)} - \hat{x}_{iju}^{B(\eta,t)(k)})| & \text{if } |(x_{i\eta ju}^{t(k)} - \hat{x}_{iju}^{B(\eta,t)(k)})| \geq \theta; x_{i\eta ju}^{t(k)} = 0 \\ \mu_{i\eta ju}^{t(k)} & \text{otherwise,} \end{cases} \tag{4.3.4}$$

4.3.0.3. *Termination criteria*

The algorithm is terminated when the dual convergence is approximately satisfied. The EPHA terminates, particularly, when the condition (4.3.5) (Takriti and Birge 2000) is satisfied (i.e. the level of the bundle constraint violation is sufficiently low).

$$\sum_{\eta=1}^{Z} Pr_\eta \sum_{i=1}^{N} \sum_{t=1}^{H} \sum_{j=1}^{O} \sum_{u=t+1}^{H+1} |x_{i\eta ju}^{t(k)} - \hat{x}_{iju}^{B(\eta,t)(k)}| \leq \epsilon \tag{4.3.5}$$

4.4. Experimental Study

The experimental study was conducted in two parts. First, each of the parameter values were varied to find a good setting for the algorithm in terms of the solution quality and computational performance. Second, the EPHA solutions were compared with the optimal solution.

For the case study, the data is generated for a moderate-size problem instance where 45 surgeries are requested over 20-day planning period for a single OR according to 9 different scenario sequences of a scenario set. The surgery durations are generated according to a probability distribution given in Chapter 2, Table 1. The density function for the procedure durations of the urology surgeries is used in particular. The cancellation cost per day is uniformly distributed between \$1700 and \$2000, which are the estimated lower and upper bounds of the cancellation costs in the US hospitals (Argo et al. 2009). The overtime cost per minute and mean number of surgery requests per day are set to avoid having extreme cases in the solution space (i.e. zero cancellation, zero total overtime).

The values of the initial penalty parameter, $\rho^{(0)}$, and update multiplier, δ, are varied to analyze the changes in the EPHA performance. $\delta = 2$ is found as a reasonable multiplier. The current value does not allow radical changes from one iteration to the other as it leads to a reasonable level of variation. Having δ fixed, the best $\rho^{(0)}$ is sought for. For each $\rho^{(0)}$, the PHA was run to find a solution for a number of instances. The difference between instances results from the variation in the number of bundled decision variables. The instances having a higher number of bundled decision variables represent relatively more complex instances. The objective function value and number of iterations used to reach a solution for each run for three different instances are compared in Table 12. $\rho^{(0)} = 1000$ is likely to be a better selection than the others, because it always finds the minimum objective value that the PHA can find. Furthermore, among the solutions that yield the best value, the ones found when $\rho^{(0)}$ is set to 1000 are reached in the lowest number of iterations. Also in some cases, even if $\rho^{(0)}$ is set to a different value, it converges to 1000 before finding the solution.

Note that when $\rho < 250$, cycling is observed. This prevents the EPHA from finding a solution. To cope with this situation, the penalty update method would need to be modified. The modification, which we leave for future research, considers cyclic behavior and likely to result in improved solutions.

Table 13 compares the optimal and EPHA solutions for the three instances discussed above. The level of gaps between the optimal and PHA solutions are not at negligible levels. The gaps are to be partially eliminated by preventing the cyclic behavior that occurs when the initial penalty parameter value is low, because lower values forces the PHA to converge to a good solution.

Table 12. The Trade-off between the EPHA Performance and Solution Quality is Illustrated for Different Type of Instances by Varying the Initial Penalty Parameter

# of bundled elements	$\rho^{(0)}$	objective value	# of iterations
1032	500	11393.2	6
1032	800	11393.2	5
1032	1000	11393.2	3
1032	2000	11393.2	3
1032	5000	11393.2	3
1032	20000	11656.7	3
1032	50000	11656.7	2
1032	75000	11656.7	2
3675	500	9308	5
3675	800	9308	4
3675	1000	9308	3
3675	2000	9428	3
3675	5000	9661.2	3
3675	20000	9661.2	3
3675	50000	9661.2	3
3675	75000	9661.2	3
5050	500	10961.5	5
5050	800	10961.5	4
5050	1000	10961.5	3
5050	2000	11081.8	3
5050	5000	11314	3
5050	20000	11314	3
5050	50000	11314	3
5050	75000	11314	3

4.5. Conclusions

This chapter proposed a number of techniques to improve the Progressive Hedging Algorithm (PHA) solution quality and the convergence characteristics.

Table 13. The Performance of the EPHA with Respect to the Optimal Solution of the DEM is Shown

# of bundled elements	best PHA solution	objective value	optimality gap
1032	11105.1	11393.2	2.5%
3675	9024.5	9308	3%
5050	10741.9	10961.5	2%

Future research will evaluate the impact of the proposed techniques will be tested on real data which will be gathered from different type of major medical centers (e.g. government type institutions, not-for-profit private academic institutions). The optimal solutions under different scenario sets will be analyzed to reveal the insights related to the optimal scheduling, cancellation and rescheduling policies.

For moderate size instances, the current form of the PHA can not outperform the typical solvers. The most important reason is the requirement of solving many subproblems, all of which are mixed integer programming models. As previously indicated, it is not necessary to solve the subproblems to optimality. Thus, a fast running heuristic for the subproblems is likely to improve the computational speed, significantly. A logical method would solve the subproblems with a lower accuracy level at the beginning iterations as suggested by Kall and Wallace (1994). Then, computational effort can be increased to better approximate the optimal solutions of the subproblems as the iteration number increases. Hvattum and Lokketangen (2009) used this approach and proposed a method to give decisions on how to vary the amount of computational effort spent to solve the subproblems based on the convergence pattern of the primal and dual variables. Furthermore, since only the parameters of some surgery scheduling decision variables vary from one iteration to the next, the subproblem solutions of the prior iteration have the potential to perform well in latter iterations. Thus, this special structure of the PHA algorithm will benefit from an efficient heuristic to solve the subproblems of the dynamic multi-period operating room planning problem.

CHAPTER 5

CONCLUSIONS AND BROADER IMPACTS

5.1. Conclusions

Optimization of surgery delivery systems is a challenging managerial problem. In this dissertation, a number of operations research models and solution techniques are proposed to solve operating room (OR) planning and scheduling problems. The analysis of the models and solutions provides significant insights into the planning and scheduling of surgeries. Algorithms (i.e. a bi-criteria Genetic Algorithm and Progressive Hedging algorithm) are developed with the aim of finding good solutions for practical surgery planning and scheduling problem instances in a reasonable amount of time.

In chapter 2, several scheduling methods were utilized to find near optimal sequences and patient appointment times for outpatient surgical procedures were proposed and discussed. First, a discrete-event simulation model was constructed to test a number of sequencing and patient appointment time setting heuristics with respect to the expected patient waiting time and expected surgical suite overtime for a single-day scheduling problem. The analysis of the solutions yields the following results. Simple heuristics can enhance actual schedules used in an Outpatient Procedure Center. Job hedging is useful to decrease patient waiting times at the expense of increasing surgical suite overtime. The trade-off between the patient waiting time and surgical suite overtime which is affected by the job hedging level depends on the sequencing heuristic used priorly. Among the sequencing heuristics, LPT (Longest Processing Time First) causes high expected overtime, and should be avoided, while SPT (Shortest Processing Time First) performs quite well. Second, a bi-criteria Genetic Algorithm was used to determine whether better solutions can be obtained for the single day scheduling problem. The analysis indicates that expending greater computational effort with a GA approach does not achieve substantial additional improvements when there is no control over daily procedure mix. Since it is easy to implement in practice, SPT should be favored over the GA. Third, the bi-criteria Genetic Algorithm was tested under the setting that the surgeries are allowed to be moved to other days. The results indicate that controlling daily procedure mix may achieve substantial improvements in performance, though the returns diminish as the time window for moving surgeries is extended.

In Chapter 3 and Chapter 4, a multi-period operating room planning problem was studied. Surgery scheduling, cancellation, and rescheduling decisions made each day over a finite planning horizon were investigated. The resulting model was formulated as a multi-stage stochastic mixed integer program. A Progressive Hedging Algorithm (PHA) was proposed to solve the problem in Chapter 3, and the structural properties of the model and algorithm were leveraged to enhance the algorithm performance in Chapter 4. Future research study directions were also proposed and discussed.

5.2. Broader Impacts

The models and solution methods that are proposed aim to simultaneously improve the patient-centered characteristics of the surgery delivery systems while keeping hospital costs at a reasonable level. The objectives of the models are oriented towards maintaining a balance between patient satisfaction and safety, and OR costs. For example, if patient waiting time is decreased using the model in Chapter 2, patients will be happier. This could also lead to an increased motivation for the surgical staff. Thus, the quality of the care could be positively affected. The mathematical model in Chapters 3 and 4 includes the objective of decreasing the number of surgery cancellations. The cancellation of a surgery can increase risk of adverse events for patients. Lowering the number of cancellations could also lead to decreases in the amount of time that the staff waits idle. Consequently, the use of the models and the solution techniques in this dissertation would provide a positive long-term benefit for many stakeholders of the health care delivery systems.

REFERENCES

Argo, J. L., C. C. Vick, L.A. Graham, K.M.F. Itani, M.J. Bishop, M.T. Hawn. 2009. Elective surgical case cancellation in the veterans health administration system: identifying areas for improvement. *The American Journal of Surgery* **198** 600–606.

Banks, J., J.S. Carson, B.L. Nelson, D.M. Nicol. 2005. *Discrete-event System Simulation*. Pearson Prentice Hall, Upper Saddle River, New Jersey.

Barro, D., E. Canestrelli. 2005. Dynamic portfolio optimization: Time decomposition using the maximum principle with a scenario approach. *European Journal of Operational Research* **163** 217–229.

Berg, B., B.T. Denton, H. Nelson, H. Balasubramanian, H. Rahman, K. Lindor, A. Bailey. 2010. A computer simulation model to evaluate operations performance of an endoscopy suite. *Medical Decision Making* **in press**.

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

Brindle, A. 1981. Genetic algorithms for function optimization. *Doctoral dissertation and technical report TR81-2. Edmonton: University of Alberta, Department of Computer Science* .

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

Cayirli, T., E. Veral, H. Rosen. 2006. Designing appointment scheduling systems for ambulatory care services. *Health Care Management Science* **9** 47–58.

Charnetski, J. 1984. Scheduling operating room surgical procedure with early and late completion penalty costs. *Journal of Operations Management* **5** 91–102.

Crainic, T.G., X. Fu, M. Gendreau, W. Rei, S.W. Wallace. 2010. Progressive hedging-based meta-heuristics of stochastic network design .

Deb, K., S. Agrawal, A. Pratap, T. Meyarivan. 2000. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimisation: Nsga-ii. *Lecture Notes in Computer Science* **1917** 849–858.

Denton, B., D. Gupta. 2003. A sequential bounding approach for optimal appointment scheduling. *IIE Transactions* **35** 1003–1016.

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

Dexter, F., A. Macario, R. Traub. 1999a. Which algorithm for scheduling add-on elective cases maximizes operating room utilization? *Anesthesiology* **91** 1491–1500.

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

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

Erdogan, S.A., B.T. Denton. 2010. Surgery planning and scheduling: A literature review .

Fei, H., C. Chu, N. Meskens. 2009. Solving a tactical operating room planning problem by a column generation based heuristic procedure with four criteria. *Annals of Operations Research* **166** 91–108.

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

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

Gerchak, Y., D. Gupta, M. Henig. 1996. Reservation planning for elective surgery under uncertain demand for emergency surgery. *Management Science* **42** 321–334.

Gillen, S. M. I., K. Catchings, L. Edney, R. Prescott, S. M. Andrews. 2009. What's all the fuss about? day-of-surgery cancellations and the role of perianesthesia nurses in prevention. *Journal of Perianesthesia Nursing* **26** 396–398.

Gruber, J. 2010. The cost implications of health care reform. *The New England Journal of Medicine* **362** 2050–2051.

Guinet, A., S. Chaabane. 2003. Operating theatre planning. *International Journal of Production Economics* **85** 69–81.

Gul, S., B. Denton, J. Fowler, T. Huschka. 2010. Bi-criteria scheduling of surgical services for an outpatient procedure center. *Production and Operations Management* .

Gupta, D. 2007. Surgical suites' operations management. *Production and Operations Management* **16** 689–700.

Gupta, D., B. Denton. 2008. Appointment scheduling in health care: Challenges and opportunities. *IIE Transactions* **40** 800–819.

Hans, E., G. Wullink, M. van Houdenhoven, G. Kazemier. 2008. Robust surgery loading. *European Journal of Operational Research* **185** 1038–1050.

Haugen, K.K., A. Lokketangen, D.L. Woodruff. 2001. Progressive hedging as a meta-heuristic applied to stochastic lot-sizing. *European Journal of Operational Research* **132** 116–122.

Helgason, T., S.W. Wallace. 1991. Approximate scenario solutions in the progressive hedging algorithm. *Annals of Operations Research* **31** 425–444.

Ho, C.J., H.S. Lau. 1992. Minimizing total cost in scheduling outpatient appointments. *Management Science* **38** 750–764.

Holland, J. H. 1975. Adaptation in natural and artificial systems. *The University of Michigan Press, Ann Arbor, MI* .

Huschka, T., B. Denton, S. Gul, J. Fowler. 2007. Bi-criteria evaluation of an outpatient surgery procedure center via simulation. *In: Henderson, S.G., B. Biller, M.H. Hsieh, J. Shortle, J.D. Tew, R.R. Barton (ed) Proceedings of the 2007 Winter Simulation Conference, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc* 1510–1518.

Hvattum, L.M., A. Lokketangen. 2009. Using scenario trees and progressive hedging for stochastic inventory routing problems. *Journal of Heuristics* **15** 527–557.

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

Kall, P., S.W. Wallace. 1994. *Stochastic Programming*. Wiley, New York.

Lamiri, M., F. Grimaud, X. Xie. 2009. Optimization methods for a stochastic surgery planning problem. *International Journal of Production Economics* **120** 400–410.

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

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

Lehtonen, J.M., J. Kujala, J. Kouri, M. Hippelainen. 2007. Cardiac surgery productivity and throughput improvements. *International Journal of Health Care Quality Assurance* **20** 40–52.

Listes, O., R. Dekker. 2005. A scenario aggregation-based approach for determining a robust airline fleet composition for dynamic capacity allocation. *Transportation Science* **39** 367–382.

Lokketangen, A., D.L. Woodruff. 1996. Progressive hedging and tabu search applied to mixed integer (0,1) multistage stochastic programming. *Journal of Heuristics* **2** 111–128.

Lovejoy, W.S., Y. Li. 2002. Hospital operating room capacity expansion. *Management Science* **48** 1369–1387.

Lowery, J. 1992. Simulation of a hospital's surgical suite and critical care area. *In: Swain, J., D. Goldsman, R. Crain, J. Wilson (ed) Proceedings of the 1992 Winter Simulation Conference, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc* 1071–1078.

Lowery, J., J. Davis. 1999. Determination of operating room requirements using simulation. *In: Farrington, P., H. Nembhard, D. Sturrock, G. Ewans (ed) Proceedings of the 1999 Winter Simulation Conference, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc* 1568–1572.

Magerlein, J.M., J.B. Martin. 1978. Surgical demand scheduling: A review. *Health Services Research* **13** 418–433.

Marcon, E., S. Kharraja, N. Smolski, B. Luquet, J. Viale. 2003. Determining the number of beds in the postanesthesia care unit: A computer simulation flow approach. *Anesthesia and Analgesia* **96** 1415–1423.

Min, D., Y. Yih. 2010. Scheduling elective surgery under uncertainty and downstream capacity constraints. *European Journal of Operational Research* **206** 642–652.

Mulvey, J.M., H. Vladimirou. 1991a. Applying the progressive hedging algorithm to stochastic generalized networks. *Annals of Operations Research* **31** 399–424.

Mulvey, J.M., H. Vladimirou. 1991b. Solving multistage stochastic networks: An application of scenario aggregation. *Networks* **21** 619–643.

Mulvey, J.M., H. Vladimirou. 1992. Stochastic network programming for financial planning problems. *Management Science* **38** 1642–1664.

Przasnyski, Z.H. 1986. Operating room scheduling: A literature review. *AORN Journal* **44** 67–79.

Rockafellar, R.T. 1976. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization* **14** 877–898.

Rockafellar, R.T., R.J.B. Wets. 1991. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research* **16** 119–147.

Roland, B., C. Di Martinelly, F. Riane, Y. Pochet. 2010. Scheduling an operating theatre under human resource constraints. *Computers and Industrial Engineering* **58** 212–220.

Santos, M.L.L.D., E.L.D. Silva, E.C. Finardi, R.E.C Goncalves. 2009. Practical aspects in solving the medium-term operation planning problem of hydrothermal power systems by using the progressive hedging method. *Electrical Power and Energy Systems* **31** 546–552.

Syswerda, G. 1989. Uniform crossover in genetic algorithms. *In: Schaefer, J. (ed) Proceedings of the Third International Conference on Genetic Algorithms, Los Altos, California: Morgan Kaufmann Publishers* 2–9.

Takriti, S., J.R. Birge. 2000. Lagrangian solution techniques and bounds for loosely coupled mixed-integer stochastic programs. *Operations Research* **48** 91–98.

Takriti, S., J.R. Birge, E. Long. 1996. A stochastic model for the unit commitment problem. *IEEE Transactions on Power Systems* **11** 1497–1508.

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

Tyler, D., C.A. Pasquariello, C.H. Chen. 2003. Determining optimum operating room utilization. *Anesthesia and Analgesia* **96** 1114–1121.

Wallace, S.W., T. Helgason. 1991. Structural properties of the progressive hedging algorithm. *Annals of Operations Research* **31** 445–456.

Watson, J.P., D.L. Woodruff, D.R. Strip. 2010. Progressive hedging innovations for a class of stochastic resource allocation problems .

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

Yellig, E.J., G.T. Mackulak. 1997. Deterministic scheduling in stochastic environments: the method of capacity hedge points. *International Journal of Production Research* **35** 369–379.

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

Zonderland, M.E., R.J. Boucherie, N. Litvak, C.L.A.M. Vleggeert-Lankamp. 2010. Planning and scheduling of semi-urgent surgeries. *Health Care Management Science* .