Multi-objective Optimization for Scheduling Elective Surgical Patients at the Health Sciences Centre in Winnipeg

by

Yin Yin Tan

A thesis submitted to the Faculty of Graduate Studies at the University of Manitoba in partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE

Department of Mechanical and Manufacturing Engineering

University of Manitoba

Winnipeg, Manitoba, Canada

Abstract

Health Sciences Centre (HSC) in Winnipeg is the major healthcare facility serving elective and emergency surgical patients in Manitoba, Northwestern Ontario, and Nunavut. An extensive evaluation of HSC's adult surgical patient flow revealed that one of the major barriers to smooth flow was the facility's Operating Room (OR) scheduling system. This thesis presents a new two-stage elective OR scheduling system for HSC, which generates weekly OR schedules that reduce artificial variability in order to facilitate smooth patient flow. The first stage reduces day-to-day variability by smoothing bed occupancy and patient volumes, increasing bed utilization, and evenly distributing OR time throughout the week. The second stage reduces the variability that may occur within a day by minimizing overtime, evenly distributing OR time among each operating theatre, and smoothing Post-Anaesthesia Care Unit (PACU) bed occupancy volumes and the number of cases that finish simultaneously. The scheduling processes in both stages are mathematically modelled as multi-objective optimization problems. An attempt was made to solve both models using lexicographic goal programming. However, this proved to be an unacceptable optimization method for the second stage, so a new multi-objective genetic algorithm, called Nondominated Sorting Genetic Algorithm II – Operating Room (NSGAII-OR), was developed. Results indicate that if the proposed system is implemented at HSC, the facility's surgical patient flow will likely improve.

Acknowledgements

First and foremost, I cannot thank Dr. ElMekkawy enough for providing this wonderful opportunity to enter the world of operations research in healthcare. His guidance led me to where I am today.

I would also like to thank Brock Wright and Helga Bryant for facilitating such an exciting research venture. Of course I could not have learnt as much as I did were it not for the efforts and support of Leanne Bernaerdt, to whom I am especially grateful.

Throughout this project, there were many people who helped me acquire the knowledge that I needed. I would love to list their names here so that each one of them knows how much I appreciated their efforts. However, the project's policy on ethics prohibits me from doing so. Therefore, without naming names, I must express my gratitude to all of the managers who allowed me to invade their respective departments and bombard them with questions! Even more surprising was their willingness to read my reports and give me feedback. I am also very appreciative of the numerous staff members who patiently answered my questions in great detail and gave me an understanding of their world. Thank you to all of the surgeons who took time out of their busy schedules to be interviewed. Without them I would have missed some incredibly important, and insightful, perspectives about patient flow and the health care system.

Finally, I wish to thank Gail Grimsen, James Nick, Laura Shea, Patricia Rothney, and Roberta Giltrow for supplying me with the remaining pieces of the puzzle.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Patient Flow in Healthcare

Although the complexity of healthcare has greatly increased during the past few decades, the design of key processes has remained relatively primitive. Patients often experience long waiting times and encounter delays or cancellations. To address these problems, many facilities have turned to quick fixes, such as downsizing or adding more resources such as beds or equipment. However, most of these changes have not resulted in the desired outcome. Recent assessments have discovered that improved patient flow can reduce or eliminate these problems. Flow refers to the way in which patients, staff, information, and materials move throughout a facility. When patient flow is not smooth, staff and patient satisfaction, hospital revenue, and patient safety are all negatively affected (Haraden and Resar 2004). Therefore, the factors that influence patient flow should be analyzed, and methods for developing and improving this flow should be generated (Lambert 2004).

Areas that have non-interchangeable resources, such as emergency departments (ED), intensive care units (ICU), surgical pre and post-operative units, and OR departments, are major bottlenecks in most healthcare facilities. Bottlenecks represent care, safety and cost issues. For example, if a patient is waiting to be transferred from the ICU to a post-operative unit, a safety issue arises because there may be another patient who urgently requires the ICU care, and a cost is incurred because expensive ICU resources are being wasted. The majority of these problems are not caused by a lack of effort from staff and cannot be resolved by working harder. Rather, it is the patient flow between and among the different departments that is the source of problems. Therefore, it must be understood that a healthcare facility is made up of several interdependent departments, where the actions of one can adversely affect another. As such, patient flow throughout the entire system must be improved, rather than just in isolated departments.

Flow depends on the inherent variation found in the healthcare delivery system (Haraden and Resar 2004). Studies have found that the variations caused by the healthcare delivery structure, termed artificial variations, are much greater than the natural variations caused by random patient arrivals and the disease state they present (IHI 2003, Brideau 2004, Henderson et al. 2004, Horton 2004). For example, a facility may experience high resource utilization at the beginning of the week, when all of its surgeons scramble to perform operations. Consequently, resource limitations may result in delays and cancellations. On the other hand, resources may be severely underutilized at the end of the week, when the surgeons decide that they would like to leave work early. It is this kind of artificial variation that causes avoidable problems. Therefore, they must be reduced to improve patient flow.

1.1 Thesis Objectives

Health Sciences Centre (HSC) in Winnipeg is the major surgical centre serving the residents of Manitoba, Northwestern Ontario, and Nunavut. In 2005, the facility embarked on a collaborative effort with researchers at the University of Manitoba to analyze the facility's adult surgical patient flow, considering the entire journey from pre to post-operative care, and to develop ideas on how it could be improved.

Over several months, researchers visited HSC's major surgical departments carrying out observations and corresponding with staff and patients. Researchers mapped the surgical patient flow processes (summarized in Chapter 3) and identified areas of concern regarding patient flow. It was discovered that the major issue hampering smooth flow was the way in which elective surgeries, mainly referred to as *cases* in this thesis, were scheduled in the operating room (OR) department. Indeed, these results were not surprising, considering that many articles in literature (IHI 2003, Haraden and Resar 2004) have pointed out that highly variable elective surgical admissions are often the major barrier to achieving smooth patient flow. This spurred the motivation to develop a new elective OR scheduling system for HSC, which is the main objective of this thesis.

It should be noted that the research for this thesis commenced before HSC's OR department, Post-Anaesthesia Care Unit (PACU), and Surgical Intensive Care Unit (SICU) moved to a new building. However, the majority of surgical patient flow processes has remained the same, and the results presented here are still relevant.

1.2 Thesis Overview

This thesis summarizes the adult surgical patient flow processes at HSC and presents a new weekly elective OR scheduling system for the facility. The system is comprised of two stages and is designed to reduce artificial variability to facilitate smooth patient flow. In the first stage, surgeons present a list of cases to be scheduled in a particular week. Each case is then assigned to different days of the week so that day-to-day variability is reduced. In the second stage, the cases scheduled on a particular day are assigned to operating theatres and given start times in a manner that reduces the variability that may occur within a day. Thus, a complete elective OR schedule is generated for each day.

Both stages are multi-objective optimization problems. The selection of a particular optimization method is driven by the type of information available and the specific characteristics of the problem. Because the goals in both problems have very obvious priorities, a biased search can be conducted to find the optimal solutions that satisfy the objectives with higher importance. Therefore, lexicographic goal programming was chosen as the multi-objective optimization method for the scheduling system.

In the first stage of the system, the developed lexicographic programming model had no trouble finding feasible, optimal solutions in a short amount of time. However, the model in the second stage was unable to find optimal solutions for the given problem size in a reasonable amount of time. Therefore, another multi-objective optimization method had to be developed for stage 2. This method needed to be able to solve large problems in acceptable amounts of time, which is a characteristic that genetic algorithms match very well. Therefore, a new multi-objective genetic algorithm for daily OR scheduling (NSGAII-OR) was developed. Through lexicographic goal programming in stage 1 and NSGAII-OR in stage 2, the schedules generated by the proposed OR scheduling system significantly reduced artificial variability compared to the actual schedules employed at HSC.

In this thesis, chapter 2 is a literature review that is comprised of two parts. The first part focuses on multi-objective optimization while the second part deals with elective OR scheduling. This is followed by chapter 3, which provides an overview of the surgical patient flow at HSC. Chapters 4 and 5 describe the first and second stages of the proposed elective OR scheduling system, respectively. Finally, the research presented in this thesis is summarized in chapter 6, where future work is also discussed.

Chapter 2: Literature Review

This chapter is comprised of two literature reviews. First, multi-objective optimization is addressed, followed by elective operating room (OR) scheduling.

2.1 Introduction to Multi-objective Optimization

Optimization is a process whereby feasible solutions are found and compared until the best solution(s) is discovered. Solutions are rated good or bad according to one or more objectives. Real-world problems are frequently expressed as optimization problems, where the objectives to be achieved are represented in the objective function while the parameters of the problem are represented by constraints. Traditionally, research has focused on the development of single objective optimization. In most real world problems, however, there are often several objectives that have to be taken into consideration. These types of problems are known as multi-objective, multi-criteria, or vector optimization problems.

Multi-objective optimization theory was first approached by Kuhn and Tucker (1950). Since then, numerous multi-objective optimization techniques have been developed and reviewed in literature (Cohon 1978, Hwang and Masud 1979, Zeleny 1982, Changkong and Haimes 1983a, Osyczka 1985, Steuer 1986, Stadler 1988, Miettinen 1999, Ehrgott 2000, Collette and Siarry 2003).

In multi-objective optimization, Pareto-optimality is the core concept. In single objective optimization, there exists a unique, optimal value for the objective in question. In multi-objective optimization, however, it is usually not possible to find a solution that is simultaneously optimal for all objectives. Instead, there exists a set of Pareto-optimal solutions, which is often called the Pareto set (Pareto 1964, 1971). Each solution in this Pareto set is *non-dominated*, meaning that there is no other solution that is better than it with regards to all objectives. In a problem's objective space, the plot of Pareto-optimal solutions is usually called the Pareto front. Essentially, each solution on this front is a trade-off of another. Figure 2-1 gives an example of the Pareto front formed by the solutions in the feasible region of a two-objective (f_1 and f_2) minimization problem.

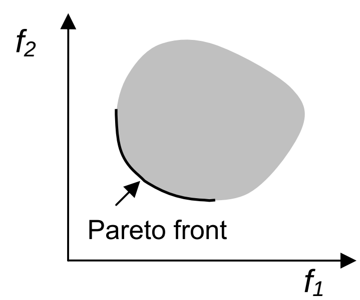

Figure 2-1 Example of a Pareto front

Multi-objective problems will have multiple solutions if the objectives are conflicting. In the rare case where no objectives are conflicting, there will only be one Pareto-optimal solution. If there is sufficient information about a decision maker's preferences regarding a problem's objectives, a biased search can be performed to find the most satisfactory solution for the decision maker. If this information is unavailable, then all Pareto optimal solutions are considered equally important, and as many solutions as possible should be found. As such, there are mainly two goals in multi-objective optimization:

1. Obtain a set of non-dominated solutions that are as close as possible to the true Pareto optimal front (i.e. convergence)
2. Obtain a set of non-dominated solutions that are diverse as possible (i.e. diversity)

The main concept in optimization consists of the first goal, regardless of whether the problem is single or multi-objective. On the other hand, the second goal is specific to multi-objective optimization and is important because it ensures that the decision maker has a good set of trade-off solutions to choose from. To evaluate the performance of different multi-objective optimization methods, several metrics (Van Veldhuizen 1999, Zitzler and Thiele 1999, Khor et al. 2005) have been developed and are all related to those two goals in some way.

Most multi-objective optimization techniques can be classified as either classical or evolutionary. Classical multi-objective optimization methods typically find only one solution in a single run, while evolutionary algorithms are generally able to find multiple solutions in a single run. Both of these methods come with their own advantages and disadvantages, and there is no particular method that will always be the best for every problem. The selection of a particular multi-objective optimization method depends on many factors, such as the type of information available and the specific characteristics of the problem, and is often a multi-objective dilemma itself. For example, classical optimization methods are usually not used to solve large-scale problems because the required computational time is normally unacceptably long. On the other hand, if an analyst has information regarding the decision maker's preferences for each objective, it will be easier to use a classical method to perform a biased search for a solution that matches the decision maker's desires, rather than spending lengthy development time on an evolutionary algorithm.

Sections 2.1.1 and 2.1.2 provide detail on classical multi-objective optimization methods and evolutionary algorithms, respectively.

2.1.1 Classical Multi-objective Optimization Methods

Classical multi-objective optimization methods normally use scalarizing functions to convert problems into single objective optimization ones. Therefore, each run is only able to produce one solution. To obtain different solutions, the parameters of the problem must be changed. Therefore, *n* runs will have to be made in order to find *n* solutions. The advantage of most classical optimization techniques is that they are easy to understand and implement. Constraints can be easily incorporated into the problem so that feasible solutions are found. Furthermore, they are often accompanied by theoretical proofs that the obtained solutions are truly Pareto-optimal. However, many classical methods normally require user-defined parameters, which may be difficult to set and is often done arbitrarily. In addition, these methods may have difficulty finding certain solutions in non-convex problems. Finally, the majority of these techniques are impractical if the user wishes to generate a large set of solutions.

Many authors (Hwang and Masud 1979, Buchanan 1986, Lieberman 1991, Miettinen 1999) have classified multi-objective optimization techniques into the following categories, based on the participation of the decision maker in the solution process: No-preference, A Priori, Interactive, or Posteriori. Figure 2-2 displays the classification used by Diwekar (2003), adapted from Cohon (1985):

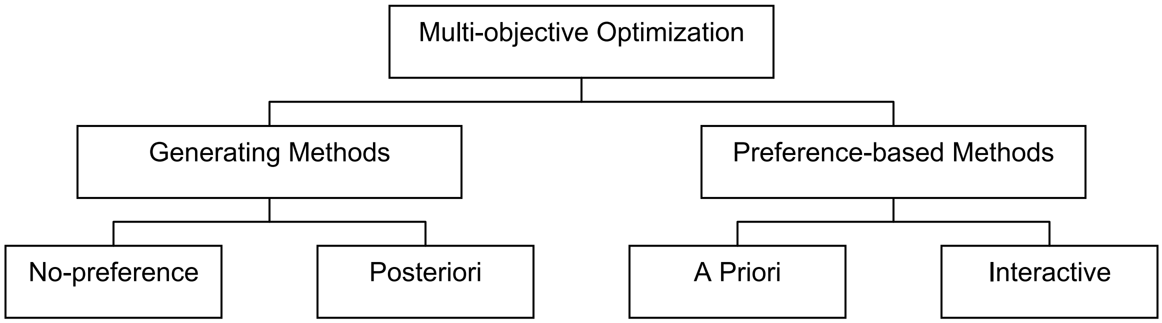

Figure 2-2 Classification of multi-objective optimization methods

Generally, classical multi-objective optimization methods are either generating or preference-based, depending on how the decision maker's input will be used in the optimization process.

2.1.1.1 Generating Methods

In generating methods, a decision maker's preferences do not have to be explicitly identified. Instead, they are implicitly determined after the decision maker has chosen his/her most preferred solution. An additional advantage is that these methods allow the decision maker to compare the trade-offs between each solution. In terms of drawbacks for generating methods, the algorithms are often complex and difficult for the decision maker to understand. Furthermore, the size of the solution set may be too large for the decision maker to realistically analyze. Finally, these methods are often coupled with large computational times as the number of objectives increases. Generating methods are often classified as *no-preference* or *posteriori*.

In no-preference methods, the decision maker's preferences are not used at all in the optimization process. Rather than generating the Pareto set, a single feasible solution, or a set of feasible solutions, is obtained and presented to the decision maker. For example, in the multi-objective proximal bundle (MPB) method (Miettinen, 1999), only a single solution is obtained which the decision maker must accept or reject.

Unlike no-preference methods, posteriori methods iteratively obtain multiple solutions in the Pareto set. After a Pareto-optimal set has been obtained for a particular problem, it is presented to the decision maker who selects the most preferred solution. Unfortunately, the optimization process is usually computationally extensive. Furthermore, it may be hard for the decision maker to select a solution from a large set of alternatives. Weighting methods and constraint methods are two common and generalized posteriori techniques reported in literature. The following is a short list of some posteriori methods (Siarry and Collette 2003, Deb 2001, Diwekar 2003):

1. Keeney-Raiffa method (Keeney et al. 1993)
2. Distance to a Reference Objective method (Wierzbicki 1982)
3. Weighting methods
    a. Weighted Sum method
    b. Non-inferior Set Estimations (NISE) (Cohon 1978, Changkong and Haimes 1983b)
4. Constraint methods
    a. ε-Constraint method (Haimes et al. 1971)
    b. Normal Boundary Intersection (NBI) method (Das and Dennis 1998)

2.1.1.2 Preference-based Methods

Preference-based methods use information about the decision maker's preferences as a base for determining how the optimal solution will be found. Hence, the optimization analyst has to quantify the importance of each objective according to the decision maker's preferences before the optimization process begins. These techniques are generally lower in computational costs than generating methods because fewer solutions have to be obtained. However, it can be difficult for the decision maker to accurately quantify his/her preferences for each objective. Furthermore, these preferences have to be made before the decision maker can see the alternative solutions, and hence they may be inconsistent with his/her true preferences. Preference-based methods are often classified as *a priori* or *interactive*.

In a priori methods, the decision maker must specify some preferences before the optimization process begins. Usually, this information is used to find only one preferred Pareto-optimal solution. The value function approach (Keeney and Raiffa 1976), the goal attainment method (Gembicki and Haimes 1975), and goal programming (Charnes et al. 1955) are examples of a priori approaches.

Goal programming is one of the oldest and most commonly cited a priori approaches (Ignizio 1978, Sayyouth 1981, Clayton et al. 1982, Romero 1991, Sandgren 1994). In goal programming, the decision maker sets goals for each objective. The problem is then transformed into a single objective one, where the aim is to minimize the total deviations from those goals. Weighted, lexicographic, and min-max goal programming are three of the most popular variations in literature. Lexicographic goal programming is one of the optimization methods used in the elective OR scheduling system proposed in this thesis.

In lexicographic goal programming, each goal is divided into a priority level. Goals in a lower priority level are infinitely more important than all of the goals in higher priority levels. Hence, no trade-offs between the goals in different priority levels are allowed. A lexicographic goal programming problem is first solved by only considering the goals in the lowest (most important) priority level. The resulting solutions for those goals are then turned into equality constraints, and the problem is solved for the goals in the second priority level. In this way, the obtained solution will not violate the goals achieved for the first priority level. This process continues until the goals in the last priority level have been addressed. This lexicographic process allows the user to sequentially filter out alternatives until one is left (Ignizio 1976). This method is suitable for problems where the decision maker is able to clearly prioritize goals.

In interactive methods, the decision maker works together with the optimization analyst or interactive computer program so that his/her preferences can be determined in an interactive way. At each iteration, the decision maker is presented with some solutions and is asked to provide information about which ones are preferable. New solutions are then generated based on the decision maker's preferences. After a number of iterations, a solution is generated which should satisfy the decision maker. Eschenauer et al. (1990) and Miettinen (1999) provide reviews of several interactive multi-objective optimization methods. The following is a list of some interactive approaches:

1. Fandel method (Eschenauer et al. 1990)
2. GUESS method (Buchanan 1997)
3. Interactive Surrogate Worth Tradeoff (ISWT) method (Changkong and Haimes 1983b)
4. Reference point (Wierzbicki 1980)
5. Simplex method (Nelder et al. 1965)
6. Step method (STEM) (Benayoun et al. 1971)

2.1.2 Evolutionary Algorithms for Multi-objective Optimization

The majority of the most recent work on multi-objective optimization has centred on various evolutionary algorithms that are based on principles of evolution (Fonseca and Fleming 1995). In general, evolutionary methods function by taking a random population of candidate solutions and determining the fitness of each solution. The higher a solution's fitness, the better it is with regards to the problem's objectives. A selection operator then chooses the better solutions to join a mating pool. From there, a search operator creates new solutions by exchanging information from the solutions in the mating pool. This is often called *crossover* or *recombination*.

Additionally, the search operator often perturbs the new solutions in their neighbourhood in a process called *mutation*. While mutation does help maintain diversity in a population of solutions, the mutation probability is usually kept very small in order to reduce the computational costs of checking the outcome of every possible mutation. Therefore, evolutionary algorithms often use operators that place more emphasis on solutions situated in less crowded regions in order to help preserve diversity among solutions. For example, *niching* techniques (Deb and Goldberg 1989) are often employed to help uniformly distributed individuals in the objective space. Cavicchio (1971), DeJong (1975), Goldberg and Richardson (1987), Oei et al. (1991), Davidor (1991), and Goldberg and Wang (1998) have all proposed various niching strategies. A drawback of many niching strategies is that an algorithm's performance is usually highly dependent on the parameters employed (Chipperfield and Fleming 1995).

When an algorithm's mutation operator is done, a new population will be created by selecting existing individuals from the parent population, the offspring population, or both. In some *elitist* algorithms, the best members from both populations are carried over into the new population. When a new population is finally created, it is now ready to go through another process of evolution, which will hopefully produce an even better, fitter population. Each iteration of this process, called a *generation*, will be repeated until some condition is satisfied, such as when the maximum number of generations has been reached or when the population fails to improve in fitness.

The attraction to evolutionary algorithms is mainly due to the fact that they can naturally produce multiple solutions in a single run and are therefore naturally suited for multi-objective optimization. This advantage is the main reason why these evolutionary algorithms have experienced considerable growth in recent years, whereas classical optimization techniques have not. Furthermore, these methods can easily deal with discontinuous or non-convex Pareto fronts, which pose great difficulties for most classical optimization methods.

Although most evolutionary methods are flexible and may be adapted for solving a wide range of problems, they come with greater development and computational costs. In addition, an algorithm often requires many parameters to be set and adjusted in order to ensure good performance. Finally, the search for feasible solutions in a constrained problem can be challenging, and these algorithms are designed to seek good solutions rather than guaranteed optimal ones (Jones et al. 2002). Classical methods, on the other hand, are easier to use if the problem is simple and they generally come with theoretical properties that ensure truly optimal solutions, provided the problem is modeled correctly.

Schaffer (1985) introduced the first evolutionary algorithm with his Vector Evaluation Genetic Algorithm (VEGA). Since then, numerous algorithms have been developed and several reviews of the different approaches have been published (Coello 1999, Deb 2001). Genetic algorithms, evolution strategies, evolutionary programming, genetic programming, simulated annealing, tabu search, particle swarm and ant colony optimization are all examples of evolutionary algorithms (Deb 2001, Collette and Siarry 2003). Deb (2001) and Khor et al. (2005) provide reviews of numerous algorithms.

2.1.2.1 Constraint Handling

In classical multi-objective optimization methods, constraints are considered by simply adding their mathematical formulas to the problem. This is not the case for evolutionary algorithms. The simplest way to handle constraints in an evolutionary algorithm is to disregard any solution that violates a constraint (Coello and Christiansen, 1999). However, this may cause the algorithm to have difficulty finding feasible solutions. Another popular constraint-handling technique is the penalty function approach, where a solution's fitness is reduced according to its constraint violation. However, this requires penalty parameter values to be chosen with care or else infeasible or poorly distributed solutions will be obtained. Two other notable approaches have been developed by Jimenez et al. (1999) and Ray et al. (2001).

Deb (2001) presented the constraint tournament approach, which modifies the definition of domination. A solution x_i will *constrain-dominate* a solution x_j if:

1. Solution x_i is feasible and solution x_j is not
2. Solutions x_i and x_j are both infeasible, but solution x_i has a smaller overall constraint violation
3. Solutions x_i and x_j are both feasible, but solution x_i dominates solution x_j with regards to the objectives

This strategy is almost the same as the one proposed by Fonseca and Fleming (1998) except for the way two infeasible solutions are handled. In their method, only the number of violated constraints is considered, rather than the extent of constraint violation.

2.1.2.2 Classification of Evolutionary Algorithms

Evolutionary algorithms can be classified into one of the following three categories, depending on the way they handle multiple objectives (Coello 2005): Aggregating, Population-based, or Pareto-domination based.

In aggregating approaches, weights are used to combine all of the objectives into a single objective. The Weight-based Genetic Algorithm (Hajela and Lin 1993) and Random Weighted Genetic Algorithm (Murata and Ishibuchi 1995) are examples of such an approach. These methods are easy to implement, but are unable to find certain solutions in non-convex problems (Das and Dennis 1997). Furthermore, choosing the correct weights can be difficult, especially if little is known about the problem. Aggregating approaches have been proposed for a few applications, such as real-time scheduling (Montana et al. 1998) and truck packing (Grignon et al. 1996).

In population-based algorithms, solutions are selected sequentially for each objective. These approaches can find Pareto optimal solutions for non-convex problems. The disadvantage is that the distribution of the Pareto optimal solutions is usually non-uniform and biased towards some objectives. VEGA (Schaffer 1985) is an example of such an approach, which has been proposed for various applications (Wilson and Macleod 1993, Cienawski et al. 1995, Coello et al. 2000).

Finally, Pareto-based approaches select individuals based on the concept of domination. In essence, these methods rank a solution according to its level of non-domination compared to others in the population. Unlike aggregating and population-based methods, Pareto-based approaches do not require a priori knowledge about a decision maker's preferences for each objective. This advantage has made these approaches very popular, despite the computational costs required by the domination check. In literature, they have been proposed for a wide variety of applications (Chipperfield and Fleming 1995, Coello et al. 2000, Goldberg 1989, Marcu 1997, Bagchi 1999, Emmanouilidis et al. 2000), and the following is a list of some approaches:

1. Multi-objective Genetic Algorithm (MOGA) (Fonseca and Fleming 1993)
2. Niched Pareto Genetic Algorithm (NPGA) (Horn et al. 1993, Horn et al. 1994)
3. Nondominated Sorting Genetic Algorithm (NSGA) (Srinivas and Deb 1993, 1994)
4. Nondominated Sorting Genetic Algorithm II (NSGA-II) (Deb et al. 2000a, Deb et al. 2000b)
5. Pareto Archived Evolution Strategy (PAES) (Knowles and Corne 2000)
6. Simple Evolutionary Algorithm for Multiobjective Optimization (SEAMO) (Valenzuela 2002)

2.1.2.3 Multi-objective Genetic Algorithms

Holland (1975) developed the concept of genetic algorithms, which are now the most popular evolutionary method for multi-objective optimization (Jones et al. 2002). This is mainly attributed to its vast applicability, global perspective, and ease of use (Goldberg 1989). Genetic algorithms were inspired by Darwin's theory of evolution, where fitter individuals will survive in a competitive environment and pass their genetic information to the next generation. Hence, a population's overall quality will increase over time. Due to the variety of applications for which genetic algorithms have been used, there is no exact way in which they function. However, most have the following elements: a population of solutions, selection according to fitness, crossover resulting in new solutions, and random mutation of the new solutions. A typically genetic algorithm process is illustrated below.

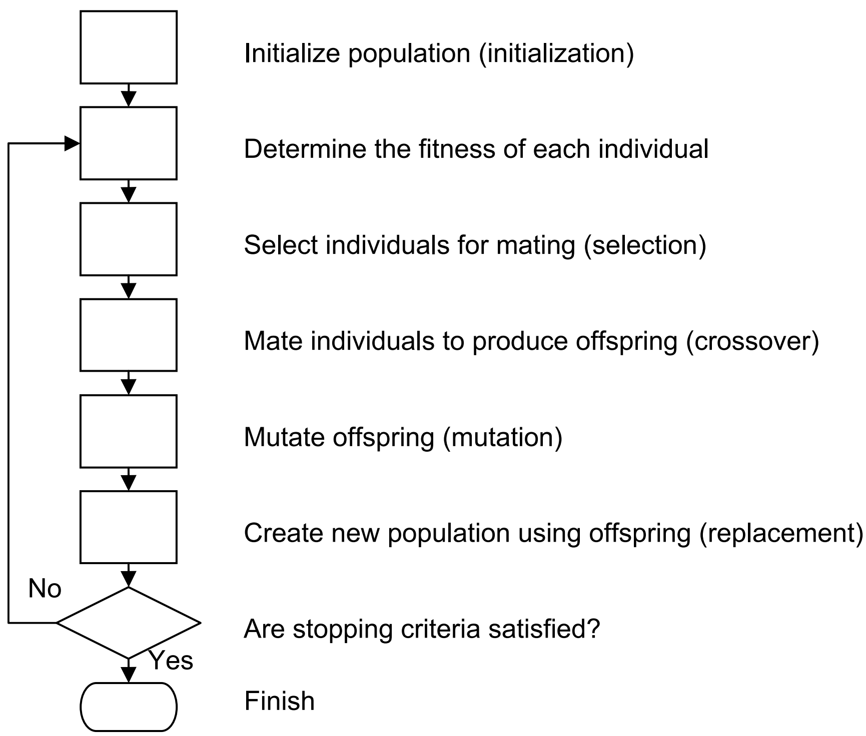

Figure 2-3 Flowchart of a genetic algorithm

Fonseca and Fleming (1993) developed Multiple Objective Genetic Algorithm (MOGA), which was the first multi-objective genetic algorithm that classified population members according to non-domination. It functions like a standard genetic algorithm except for the way in which fitness is assigned to solutions. Each solution has a rank equal to one plus the number of solutions that dominate it. Hence, a lower rank signifies a better solution.

Each solution is then assigned a fitness value based on their rank. This fitness assignment scheme is very easy to use, and MOGA can easily be applied to many problems. However, results are sometimes biased towards certain solutions. Furthermore, there may be sensitivity towards the shape of the Pareto-front, along with the density of solutions in the search space.

Another popular multi-objective genetic algorithm is Nondominated Sorting Genetic Algorithm (NSGA) developed by Srinivas and Deb (1993, 1994). When it was developed, some researchers found that its overall performance was lower than compared to the popular MOGA (Fonseca and Fleming 1993) and was more sensitive to the niching parameters used (Coello 2001). Deb et al. (2000a, 2000b) modified NSGA to improve its performance, which resulted in the birth of NSGA-II. NSGA-II is a more computationally efficient algorithm than its predecessor, and employs elitism and a crowded comparison operator that instils diversity without the setting of additional parameters. However, the non-dominated sorting required by this algorithm is performed on a population that is double the size evaluated by most other algorithms. The genetic algorithm developed for this thesis is based upon NSGA-II, and hence it will be described further in chapter 5.

2.1.2.4 Multi-objective Evolution Strategies

Evolution strategies are another popular concept used in multi-objective optimization. These strategies were first applied during the 1960s (Lichtfuss 1965, Rechenberg 1965, Schwefel 1968). Two recently developed strategies are the Predator-Prey Evolution Strategy (Laumanns et al. 1998) and the Pareto Archived Evolution Strategy (PAES)

(Knowles and Corne 2000). Evolution strategies differ from genetic algorithms in that they normally do not use crossover operators (Deb 2001). A basic evolutionary strategy is illustrated below.

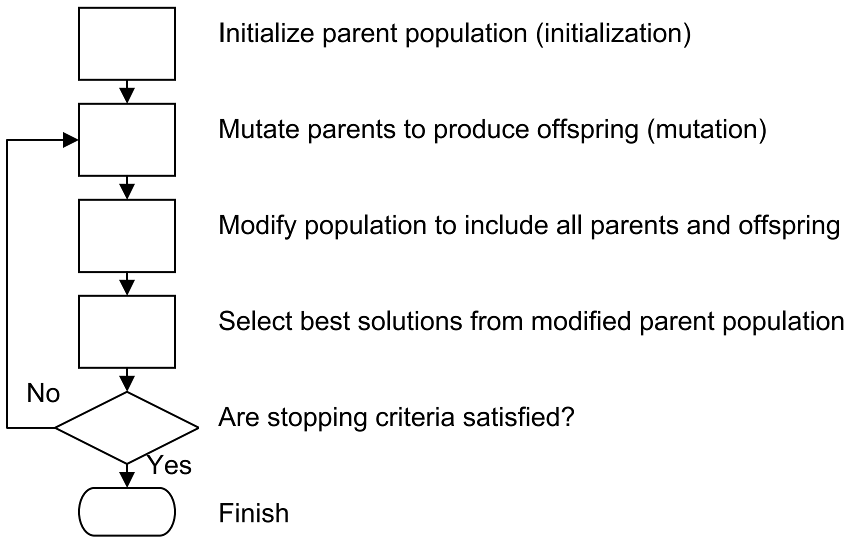

Figure 2-4 Flowchart of an evolutionary strategy

A typical evolutionary strategy begins by creating a parent population. Randomly chosen parent solutions are first mutated to produce offspring. The best solutions from both the parent and offspring populations are then selected to form a new parent population which will undergo another process of mutation and selection. Hence, an evolutionary strategy is elitist. In some variations, the best solutions are only chosen from the offspring population, or elitism is controlled by selecting a limited number of parent solutions.

2.2 Introduction to Elective Operating Room (OR) Scheduling

Elective operating room (OR) scheduling refers to the broad range of activities required to achieve a fully functioning OR department. Determining staffing amounts, allocating OR time, prioritizing emergencies, and scheduling surgical cases/operations are all examples of OR scheduling activities. These activities are dictated by the type of elective OR scheduling system being used, along with the type of elective OR scheduling policy being employed.

Dexter and Epstein (2003) group elective OR scheduling systems into three categories: Any Workday, Fixed Hours, or Reasonable Time. Because the Reasonable Time system is rarely used, only the Any Workday and Fixed Hours systems will be discussed.

The Any Workday system is the most widely used in the US. In this system, there are no restrictions on the amount of OR time that a surgeon is allowed to use. Provided that they can be done safely, cases will be performed on the dates chosen by their surgeons and patients, even if there is not enough available OR time (i.e. overtime is needed). However, facilities may not be able to provide accurate start times for some cases since there are a limited number of operating theatres.

In the Fixed Hours system, cases are only scheduled if they will not utilize more OR time than available. In this type of system, the operational objective is to minimize underutilized OR time. Some facilities have no choice but to use this system, such as ones who have fixed annual budgets. In Canada, the provincial governments are the key providers of health care and hospitals are provincially funded in advance through a fixed annual budget. Therefore, the Fixed Hours system is used by most Canadian healthcare institutions, such as Health Sciences Centre (HSC) in Winnipeg, on which this thesis is based.

In the literature, there are three main elective OR scheduling policies described: Block, Open, or Modified Block (Patterson 1996, Marcon and Kharraja 2003). Since most literature deals with either block or open scheduling, modified block scheduling will not be addressed further. Sections 2.2.1 and 2.2.2 detail the block and open scheduling approach, respectively.

2.2.1 Block Scheduling

The development of OR schedules through block scheduling can be thought of as a three-stage process. In the first stage, blocks of staffed OR hours are allocated to each surgical group. A surgical group may be made up of an individual surgeon or a collection of surgeons. Usually, this group is made up of surgeons belonging to the same service (i.e. cardiac, neurosurgery, orthopaedic, etc.). The blocks of staffed OR hours may take up a full day, half of a day, or some other variation. Ideally, blocks should be whole days, although this may not always be practical (OR Manager 2003, Patterson 2004). If a block belongs to a collection of surgeons, it may be broken down into individual blocks belonging to a specific surgeon.

In an Any Workday system, the first stage is accomplished by estimating the future amount of OR time required by each surgical group and then allocating OR time according to the expected demand. In a Fixed Hours system, on the other hand, the hours of OR time must first be calculated, which is usually based upon the facility's budget for peri-operative nursing. The available OR time is then divided among the different surgical groups based on some criterion, such as utilization or contribution margin.

In the second stage of a block scheduling policy, a master surgical schedule (MSS), consisting of the blocks allocated to each surgical group in stage 1, is created. This MSS is a cyclic timetable that must be feasible to implement in terms of available resources (e.g. staffing, number of operating theatres, etc.). The cycle time for a MSS is usually one week, two weeks, or one month. The MSS can be thought of as being equivalent to the aggregate production plan in a manufacturing environment. Figure 2-5 is a simple example of a MSS with a cycle time of one week in a facility with 3 operating theatres (OR1–OR3), 5 surgical groups (A–E), and half days blocks.

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
OR1	A	B	C	B	A	C	C
	A	E	C	C	A	C	C
OR2	B	C	A	E	E	C	E
	B	C	A	E	E	C	E
OR3	C	D	A	D	B	D	D
	D	D	A	D	B	D	A

Figure 2-5 Example of a master surgical schedule (MSS)

In the third stage of block scheduling, cases are scheduled into the MSS. If the MSS created in stage 2 did not specify operating theatre assignments for each block, then this will have to be carried out. This finally results in daily OR schedules, which list the actual operations to be performed, along with their surgeons, operating theatres, and start and end times. They are similar to production schedules where the tasks to be completed (i.e. operations) are defined and assigned to production resources (i.e. operating theatres). Figure 2-6 depicts the basic block scheduling process.

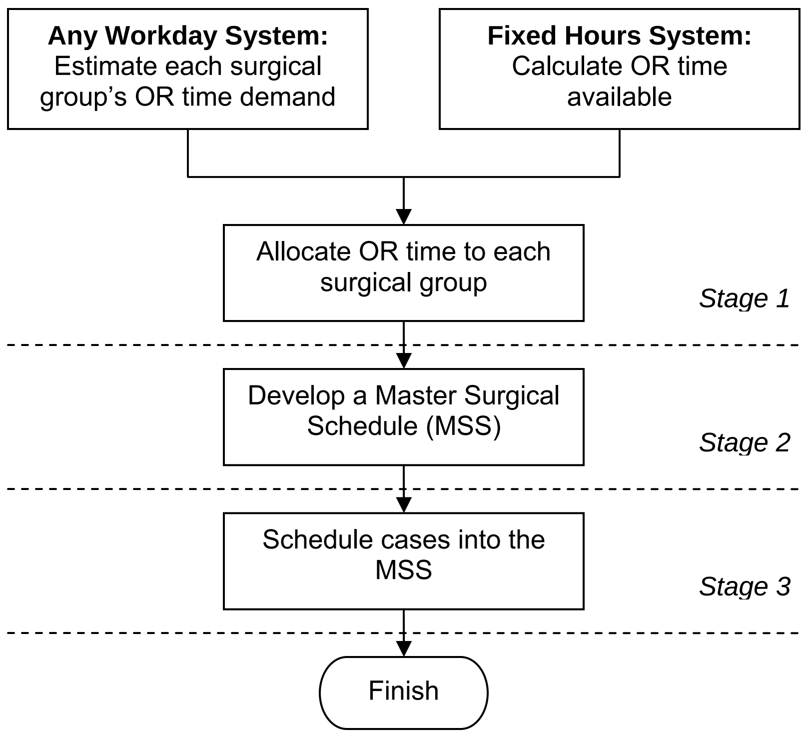

Figure 2-6 Flowchart of Block Scheduling

The following sections 2.2.1.1 to 2.2.1.3 describe the literature that can be found regarding the three stages of block scheduling.

2.2.1.1 Stage 1 – Block Time Allocations

In the first stage of block scheduling, blocks of OR time are allocated to each surgical group. Since the steps in this stage differ according to the scheduling system being used, the literature regarding this stage will be presented separately.

2.2.1.1.1 Any Workday System

When allocating block time in an Any Workday system, the future amount of OR time required by each surgical group must first be estimated. Dexter et al. (1999b) determined that using historical data from the twelve most recent four week periods is an effective way of predicting a surgical group's future demand. These estimates are then used to assign blocks to each group. Hence, estimates must be sufficiently accurate so that each group will be allocated enough OR time to complete its elective cases without incurring underutilized OR time or overtime, thereby maximizing OR efficiency and reducing costs (Dexter et al. 1999a, Dexter et al. 2001, OR Manager 2003). Indeed, Dexter et al. (1999b) found that allocating slightly more block time than a surgeon needs to complete his/her elective cases can cause an 8% decrease in OR utilization, whereas allocating slightly less time can cause a 7% increase.

Dexter et al. (2000) demonstrated how a three-step mathematical technique can be used to determine how much block time should be allocated to surgical groups during a future four week period. First, each group's total hours of elective cases is forecasted using the method presented by Dexter et al. (1999b). From there, the relative cost of overtime versus the cost of underutilized OR time is calculated as shown by Dexter and Traub

(2000a). The subsequent cost ratio, estimated future demand, and the number of hours in each staff shift are then used in a formula for optimally allocating block time to maximize OR efficiency as described in Strum et al. (1999).

Most of the time, each surgical group consists of a collection of surgeons. A group's block time is then broken down into individual blocks belonging to a specific surgeon. To minimize OR costs, some authors (Dexter et al. 1999c, Dexter et al. 2000) suggest that overflow block time should be made available for performing cases that cannot be completed during an individual surgeon's regular block time. For example, each group's OR time can be divided into regular block time for each surgeon, and overflow time that can be shared by each surgeon in the group.

2.2.1.1.2 Fixed Hours System

The first step that needs to be carried out in a Fixed Hours system using a block scheduling policy is the determination of how much OR time will be made available. This time can then be distributed to surgical groups.

Dexter and Epstein (2003) described a simple way of allocating OR time in a Fixed Hours system. First, each surgical group is ranked in descending order, according to their utilization (or whatever quantity is desired to be maximized) per allocated block. The highest ranked groups are then allocated the maximum amount of OR time blocks allowed, until the stage is reached where the remaining available OR time is less than a group's maximum allowance. When this occurs, that group will be given whatever OR time is left, and all other groups ranked below it will not receive any allocated OR time. If more constraints need to be added to the allocation process, more sophisticated methods can be used, such as linear programming.

For example, there has been literature on how to determine the case mix (i.e. number, type, and price of services) that will result in the achievement of some monetary objective, such as the maximization of profits or minimization of costs. This resultant case mix may then be used to determine how much OR time each surgical group should be allocated. Blake and Carter (2002) used goal programming for case mix planning so that a facility would be able to break even in terms of revenue and costs while simultaneously preserving the income of physicians and minimizing disturbances. This method is particularly useful for public Canadian healthcare facilities who are faced with restricted funding and need to determine how much OR time should be allocated to each surgical group.

Similarly, Kuo et al. (2003) used linear programming to allocate OR time by determining the required case mix that would maximize total weekly revenue. Mulholland et al. (2005) also used linear programming to calculate the appropriate case mix that would maximize financial outcomes for both the hospital and surgeons. However, their model took more resource constraints into account than Kuo et al. (2003). They developed several models to test the impact of limited resources, different financial objectives, and changes in expected procedure volumes or procedure mix using actual data from the Department of Surgery at the University of Michigan.

To determine how much time to allocate to each surgical group, the first phase of the scheduling method presented by Testi et al. (2007) used a bin packing model that considered each group's waiting list and the time required to clear it.

2.2.1.2 Stage 2 – Master Surgical Schedule (MSS) Development

The second stage of block scheduling requires a MSS consisting of each surgical group's blocks to be created. Blake et al. (2002) presented an integer programming method to generate MSSs so that the difference between each surgical group's target and actual OR time allocation is minimized. Similarly, Blake and Donald (2002) used integer programming for the development of new MSSs at Mount Sinai Hospital, Toronto, in a quick, unbiased way after OR time was reduced due to a decline in funding. Based on their work, Rohleder et al. (2005) used goal programming to improve OR scheduling and increase patient flow. Their model reduced variability in terms of daily patient volumes, although overall patient volumes generally remained the same. Testi et al. (2007) also used mathematical programming to create MSSs which maximize surgeon preferences with respect to their assigned dates for each block, along with expected patient length of stays (LOS).

Recent works have addressed controlling bed occupancy through MSS development. Belien and Demeulemeester (2007) proposed and evaluated several mixed integer programming (MIP) and simulated annealing models. They assumed that each surgeon only performs one type of surgery with a LOS following a multinomial distribution and a deterministic case duration. Similarly, Calichman (2005) developed a mathematical linear programming model to create weekly MSSs that maximize revenue while respecting constraints such as bed availability. OR time is allocated to each surgical group by assuming that each group's cases have a duration and LOS equal to the group's average.

Van Oostrum et al. (2006) demonstrated how to maximize OR utilization while levelling demand for the Intensive Care Unit (ICU) and surgical wards. Case durations are determined using a lognormal distribution. Like Calichman (2005), however, the estimated LOS for each group's case is equal to the group's average LOS. The problem is constructed as an integer linear program and solved using a column generation approach.

2.2.1.3 Stage 3 – Surgical Case Scheduling

In the third stage of the block scheduling policy, each surgical group selects the cases they want to perform in their blocks. These cases are then scheduled into the MSS. If the MSS created in stage 2 did not define the operating theatre assignment for each block, these assignments can easily be made by the OR department.

Often, the surgeons in each surgical group schedule cases into their blocks on a first come first serve basis, or each surgeon is allocated a portion of the group's blocks with which they can schedule in whatever manner they desire. Therefore, the OR schedule is created as each surgeon schedules their own cases. In this type of situation for a Fixed Hours system, the scheduling process is very simple because a surgical group can only schedule cases if the required durations do not surpass their allocated OR time. In an Any Workday system, however, the scheduling process is slightly more complex because surgeons may schedule as many cases as they want, regardless of whether their allocated OR time has been exceeded. For regular block time scheduling in an Any Workday system, Dexter and Traub (2002) offer three rules for scheduling to maximize OR efficiency. First, a surgical group should schedule cases into their own allocated OR time until there is none left. Second, a case should not be scheduled for completion in overtime if it can start earlier in another of the group's operating theatres. Finally, if a surgical group has fully scheduled its allocated OR time, additional cases should be scheduled into another group's allocated OR time, rather than into overtime.

In other facilities, a surgical group may select the cases to be scheduled into each of their blocks, but the sequencing of cases in each block is determined by the OR department. Dexter and Traub (2000b) presented a statistical approach for case sequencing when limited resources are required. Cases are sequenced in order to achieve a relatively low probability of overlap between cases needing the same equipment or personnel. Lebowitz (2003) presented the notion that scheduling short procedures before long ones can limit variability in actual case durations. This is because short procedures theoretically have less inherent variability (Strum et al. 1998, Zhou and Dexter 1998). Hence, cases may simply be sequenced from shortest to longest to reduce schedule disruptions. This theory was supported by Monte Carlo simulations using different combinations of short and long procedures. The results indicate that scheduling in such a way can increase the number of procedures that start on time and also decrease overtime without reducing surgical output. His findings help explain why ambulatory surgery centres, where cases are typically short, tend to run more according to schedule than tertiary care centres.

Similarly, Testi et al. (2007) used simulation in the third phase of their proposed scheduling strategy to test the sequencing of cases into a block according to three heuristics: longest waiting time (LWT), longest processing time (LPT), and shortest processing time (SPT). Like Lebowitz (2003), the authors found that SPT resulted in the least overtime and case delays or cancellations, followed by LPT and finally LWT.

Denton et al. (2007) also tested three sequencing heuristics to determine the best one for reducing deviations between planned and actual schedules. Cases were sequenced in order of increasing mean, variance, or variation coefficient with respect to case durations. The authors concluded that the heuristic based on variance performed the best. This is because the early scheduling of cases with high duration variability will likely delay the start of all the following cases in the same operating theatre.

In a Fixed Hours facility described by Fei et al. (2006), each surgical group submits a list of cases to be scheduled, without specifying a block assignment for each case. In this situation, the OR department must allocate each surgical group's case into one of the group's blocks. To allocate cases into blocks over a period of one week while matching utilized and available hours of OR time in each block, the authors used column generation to solve a linear program transformed from a binary set partitioning problem. Following this, a hybrid genetic algorithm with a tabu search local operator was used for sequencing the cases assigned to each block. The resultant occupancy in each recovery room bed was also determined. The hybrid genetic algorithm determined the sequence in each block and recovery room bed that would result in the earliest completion time for the last patient in both the OR department and recovery room.

2.2.2 Open Scheduling

In some facilities, an open scheduling policy is employed. First, surgeons submit a list of cases that they wish to schedule over a given time period specified by the OR department. In an Any Workday system, each case has a surgical date that is dictated by their surgeon. Therefore, a facility functioning under an Any Workday system and open scheduling policy will always have a designated time period of one day. In a Fixed Hours system, the designated time period can be any number of days.

Once the OR department has received the list of cases from each surgical group, the OR department will either schedule all, or just a portion of the cases, depending on the type of scheduling system being used. In an Any Workday system, all cases will be performed provided that they can be done safely. Therefore, the OR department will schedule all of the cases submitted. In a Fixed Hours system, there is a limit on the amount of OR time available. Therefore, if the total duration of all submitted cases exceeds the amount available, the OR department will only be able to schedule a portion of the submitted cases. In order to avoid the burden of selecting which cases should be performed, some facilities may allocate OR time to each surgical group before they submit their list of cases, so the groups can decide how to utilize their given OR time themselves.

Although the steps in open scheduling are much simpler than block scheduling, the scheduling process is usually more complex. However, it allows for more optimal use of resources because the facility has an overall view of the cases to be scheduled. The main disadvantage of open scheduling is that patients and surgeons will not know when their cases are scheduled until the facility has generated the complete OR schedule.

Section 2.2.2.1 describes literature pertaining to scheduling when the OR department must schedule all of the cases submitted by each surgical group, while section 2.2.1.3 describes the literature addressing scheduling when the OR department is only able to schedule selected cases.

2.2.2.1 All Cases Scheduled

This section describes literature pertaining to the situation where the OR department must schedule all of the cases submitted by each surgical group. In an Any Workday system, surgeons choose the surgical dates for each of their cases. Therefore, only daily OR schedules are created. In a Fixed Hours system, OR schedules may be created over any number of days. Literature demonstrates how the use of certain methods may optimize some criteria, such as utilization.

For daily OR scheduling, Sier et al. (1997) used simulated annealing to develop feasible schedules according to patient age and estimated case duration. Cases belonging to younger patients and cases with longer estimated durations are given earlier start times.

Marcon et al. (2001) and Marcon et al. (2003) translated their problems into multiple knapsack models so that deviations between planned and actual daily schedules, called *Risk of No Realization* (RNR), were minimized. These deviations are attributed to variable case durations. Cases are assigned to operating theatres and sequenced from longest to shortest duration.

Jebali et al. (2006) presented two MIP strategies for minimizing overtime, underutilized OR time, and patient waiting time (i.e. hospitalization before surgery). Constraints such as operating theatre opening hours, operating theatre suitability, allowable overtime hours, allowable surgeon operation hours, recovery room bed numbers, and equipment availability are all considered. In the first strategy, cases are first assigned to operating theatres, before the cases in each operating theatre are sequenced. In the second strategy, operating theatre and start time assignments are determined simultaneously. As expected, the second strategy results in slightly better results, but takes longer to solve.

For scheduling over a time horizon of one or two weeks, which can only be used in a Fixed Hours system, Guinet and Chaabane (2003) developed a method for assigning cases to different time periods and operating theatres, while considering constraints due to operating theatre opening hours, allowable overtime hours, and minimum and maximum surgeon operation hours. They modelled the problem as a capacity constraint assignment problem, with the objective of minimizing costs caused by patient waiting time and overtime. They solved the problem using an extension of the Hungarian method.

Velásquez and Melo (2005) offered a set packing approach that is able to generate schedules over a time horizon between one and seven days. Their method produces optimal schedules with regards to case priorities, while satisfying resource constraints.

Roland et al. (2006) presented a two-step MIP approach for scheduling cases over a time horizon of several days, where each case has an associated earliest and latest allowable operation date. The model minimizes costs due to the operating theatre opening hours and overtime. Renewable (e.g. staff) and non-renewable resources (e.g. pharmaceuticals), along with the maximum operation hours for each surgeon, are considered as constraints. As an alternative method for efficiently solving the problem, the authors introduced a genetic algorithm that generates solutions violating the least amount of constraints.

Using Lagrangian relaxation, Perdomo et al. (2006) demonstrated how to schedule cases over a number of days which minimize the sum of completion times for each case, while respecting operating theatre, transport personnel, and recovery room bed constraints.

2.2.2.2 Only Selected Cases Scheduled

In some facilities using the Fixed Hours system, surgeons are not allocated hours of OR time. Instead, they are asked to submit a list of cases that they wish to schedule over a given time period. From these lists, the OR department selects cases to be scheduled, provided that they have a total duration that does not exceed the OR time available.

For daily OR scheduling, Ozkarahan (2000) used a goal programming method to select and assign cases to operating theatres while minimizing idle time and overtime, maximizing surgeons' preferences for operating theatres and case priorities, and minimizing Intensive Care Unit (ICU) bed conflicts.

For weekly OR scheduling, Ogulata and Erol (2003) used a three-stage hierarchical goal programming approach. First, the cases to be performed are picked from a waiting list in order to minimize overtime and underutilized OR time, and to evenly distribute the number of cases with the same time durations (i.e. short, medium, or long). Second, each case is assigned to a surgical group in a way that balances the distribution of cases among each group and minimizes excess use of OR time allocations. Finally, cases are assigned to dates and operating theatres so that waiting time is minimized while the number of cases with the same time durations on each day is levelled.

Chaabane et al. (2006) presented a method for the selection and assignment of cases to operating theatres over a weekly time horizon, while considering constraints due to operating theatre opening hours, allowable overtime hours, and minimum and maximum surgeon operation hours. They used a linear program that minimizes the difference between available OR time and each surgical group's requested hours of OR time.

Oddly, the three methods described above do not sequence the cases in each operating theatre. However, the sequencing methods cited in section 3.4.3 (Dexter and Traub 2000b, Lebowitz 2003, Denton et al. 2006, Fei et al. 2006, Testi et al. 2007) may be used.

2.2.3 Comparison between the Proposed System and Literature

At HSC, like in most facilities, surgical dates are chosen by surgeons. However, resources are utilized better when a facility is given this control. For example, Dexter et al. (1999c) concluded through simulation that in order to maximize utilization during regular block time scheduling, control over choosing surgical dates must be moved from the surgeons to the facility. Similarly, the simulations of Dexter et al. (2000) showed that staffing costs were lowest when surgeon and patient preferences were not taken into consideration when scheduling into overflow block time.

For this reason, the proposed scheduling system was designed to generate weekly elective OR schedules with full control over surgical date, operating theatre, and start time assignments. Hence, the proposed system follows an open scheduling policy, where each surgical group is allocated hours of elective OR time during each week. The OR department then schedules all of the elective cases submitted by each surgical group, which is carried out in two stages.

The first stage aims to reduce day-to-day artificial variability, while the second stage reduces artificial variability within a day. Besides the standard constraints that can be found in literature with respect to resource availability (e.g. surgeon, operating theatre, beds, etc.), there are several additional constraints specific to HSC. For instance, there are different types of elective patients at HSC, who take different paths throughout the hospital depending on their patient type. Furthermore, some patient types have to meet specific discharge deadlines, in addition to keeping within the limits on the number that can be scheduled each day. The following section provides a comparison between the proposed elective OR scheduling system and those found in literature.

2.2.3.1 Stage 1 of the Proposed Scheduling System

In this thesis, the aim of the scheduling system's first stage is to assign cases to different days of the week in a way that will reduce day-to-day variability. In order to achieve this, the system's first goal is to smooth daily bed occupancy in the post-operative units. In literature, several authors have addressed this issue when generating the MSS in a block scheduling policy. For example, Belien and Demeulemeester (2007) demonstrated a way to develop MSSs assuming that each surgeon performs only one type of case with a LOS following a multinomial distribution. However, most surgeons perform a variety of cases, each with their own mean LOS. Similarly, Calichman (2005) and van Oostrum et al. (2006) presented a way of creating MSSs by assuming that each surgical group's cases have the same LOS, equal to the group's average LOS. These methods use simplistic assumptions because there is no data on the actual cases to be performed when an MSS is generated. To avoid this, the proposed weekly scheduling system uses an open scheduling policy.

Many weekly OR scheduling methods reported in the literature do not take post-operative bed occupancy into account. For example, in the models presented by Guinet and Chabaane (2003) and Ogulata and Erol (2003), patients are known to have arrived, or have been asked to arrive, at the hospital on a particular date. Cases are then scheduled in order to reduce costs associated with patients waiting in the hospital for surgery.

However, they do not consider the time a patient will spend in the hospital after surgery, along with whether or not there will be enough available beds to accommodate all of the patients already in the hospital or scheduled to arrive. Furthermore, the studied facilities are very different from HSC because the majority of HSC's elective surgical patients only arrive at the hospital on the day of surgery, and will usually be sent home if their surgery is postponed to another day.

Finally, the first stage of the proposed model also aims to smooth daily OR utilization and daily patient volumes in the pre-operative units. To the author's knowledge, the latter goal has not yet been addressed in literature. Hence, this thesis' attempt to generate OR schedules with both balanced daily OR utilization and patient volumes in the pre- and post-operative units is a new research venture.

2.2.3.2 Stage 2 of the Proposed Scheduling System

After cases have been assigned to the different days of the week in the first stage, the proposed system's second stage carries out daily OR scheduling by attempting to reduce the variability that may occur within a day. In this second stage, the four goals are to minimize overtime, balance utilization among each operating theatre, smooth the bed occupancy in the Post-Anaesthesia Care Unit (PACU), and minimize the number of cases that finish simultaneously.

There are similar works in literature that address daily OR scheduling. Sier et. al. (1997) scheduled cases according to patient age (i.e. youngest first) and estimated case duration (i.e. longest first). This strategy is unsuitable for this thesis because the proposed scheduling system was developed for elective patients. Hence, their cases are not urgent enough to justify scheduling younger patients first. In addition, literature (Lebowitz 2003, Testi et al. 2007) has shown that scheduling cases from shortest to longest, rather than longest to shortest, will likely result in less overtime, delays, and case cancellations. Perhaps Sier et al. (1997) saw other benefits from sequencing in such a way, but they were not disclosed.

Ozkarahan (2000) proposed a daily OR scheduling method using goal programming to minimize under-utilized OR time and overtime, while respecting constraints such as the availability of beds in the Intensive Care Unit (ICU). His method assigns cases to operating theatres, but does not specify their order. On the other hand, the proposed system does not consider ICU bed constraints because the vast majority of HSC's elective patients go to the PACU instead. The proposed method, however, does address two additional goals, in addition to determining the start times for each case. Therefore, the proposed model is a more complete scheduling approach.

Marcon et at. (2001) and Marcon et al. (2003) scheduled cases in a way that would minimize deviations between planned and actual daily schedules, caused by variable case durations. It is felt that this goal will have less impact on HSC's elective surgical patient flow than the ones addressed in the proposed system.

Velásquez and Melo (2005) used a set packing approach to generate schedules that would be optimal with regards to case priorities, while satisfying resource constraints. For example, some cases may be more urgent than others, or a surgeon may wish to perform certain cases earlier in the day. The goals of the proposed system in this thesis differ in that they aim to reduce artificial variability within a day, caused by the times at which cases are scheduled to start. Case priorities are not considered because only elective patients are being scheduled and hence their medical outcomes should not be greatly affected by small changes in their surgical dates or times.

Jebali et. al. (2006), Perdomo et al. (2006) and Fei et al. (2006) all demonstrated how to schedule cases while respecting bed constraints in the recovery room, known as the PACU at HSC. However, the proposed model attempts to smooth the PACU's bed occupancy throughout the day, as opposed to just ensuring that there are enough beds available. This will reduce artificial variability and the chances of delays due to a lack of PACU beds or staff if schedule disruptions occur.

Finally, the proposed method aims to reduce the number of cases that finish simultaneously, which will reduce the number of peri-operative aides (PAs) required to clean up the operating theatres between cases. To the author's knowledge, this issue has not been addressed in literature.

Chapter 3: Analysis of Surgical Patient Flow at Health Sciences Centre

Health Sciences Centre (HSC) in Winnipeg is the major trauma centre serving the entire province of Manitoba, in addition to Northwestern Ontario and Nunavut. HSC's adult Operating Room (OR) department is the main place where surgeries are performed on both adult elective and emergency patients.

In 2005, HSC management and researchers at the University of Manitoba decided to initiate a project that would analyze the facility's adult surgical patient flow and generate improvement ideas. Before the project could begin, approval needed to be obtained from the Education/Nursing Research Ethics Board (ENREB) at the University of Manitoba. Once this approval was obtained, the researchers began visiting HSC's major surgical departments, where they carried out observations and corresponded with staff and patients. The major departments visited were the Pre-Admission Clinic (PAC), Admitting, MS3, B3, OR, Post-Anaesthesia Care Unit (PACU), Surgical Intensive Care Unit (SICU), and A5 (an inpatient unit). Ten days were spent conducting observations in PAC, MS3, B3, and the OR, where 10 staff members and 2 patients were observed and interviewed in each department. Seven days were spent in Admitting, where 5 staff members were observed and interviewed. Finally, five days were spent in PACU, SICU, and A5. Four staff members were observed and interviewed in PACU and A5, while 2 staff members were interviewed in SICU. Three surgeons were also interviewed during the course of the project.

Staff approached for observations and interviews were identified by HSC management, based on who they felt would best be able to help researchers through the learning process. HSC management would explain the study objectives to identified staff before introducing the researchers. The researchers would then read a letter of consent and have them sign it if they agreed to being observed and/or interviewed as per the Management, Nurse or Physician Questionnaire.

On the other hand, patients who were approached for observations and interviews were selected randomly. First, researchers would ask a staff member to inform the patient that a patient flow study was being conducted and inquire if they could be approached by a researcher. If the answer was “yes”, a researcher would go to the patient and describe the study’s objectives, present the Patient Questionnaire, and ask if the patient was willing to be observed and interviewed using the Patient Questionnaire. If the patient agreed, the researcher would obtain the patient’s signature on a letter of consent. A copy of all questionnaires and letters of consent can be found in Appendix C.

In this chapter, sections 3.1 and 3.2 describe HSC’s classification and flow of elective and emergency surgical patients, respectively. Sections 3.3 through 3.8 detail the individual stages making up the facility’s elective surgical patient flow. Section 3.9 explains how the OR department is scheduled at HSC, while section 3.10 describes issues that plague this system. Finally, section 3.11 suggests possible methods of flow improvement. A summary is given in section 3.12.

3.1 Elective Patients

Elective surgical patients are those who have planned surgeries. At HSC, elective surgical patients are classified into three categories:

1. Day
    a. Same Day (SD)
    b. Overnight
2. Same Day Admission (SDA)
3. Inpatient

Day and SDA patients are all admitted to HSC on the day of surgery, and their classification depends on their expected discharge time. Day patients are expected to be discharged within 24 hours after surgery, and are split into either SD or Overnight patients. SD patients are expected to be discharged on the same day of surgery, while Overnight patients are expected to be discharged the morning after. On the other hand, SDA patients are expected to be discharged more than 24 hours after surgery. All elective patients who need to be admitted a day or more prior to surgery are classified as Inpatients. This is usually because their surgeons have ordered extensive pre-operative tests or actions to be carried out.

Figure 3-1 depicts the general flow of elective surgical patients at HSC. For the majority of elective surgical patients, their journey through HSC starts when they see their surgeon, usually at the surgeon's office, and surgery is agreed upon. From there, all patient types, except for Inpatients, may be requested by their surgeon to visit the Pre-

Admission Clinic (PAC) sometime prior to the day of admission. On the day of admission, all patients must first stop at the Admitting department for registration, before proceeding to their pre-operative unit. When it is time for surgery, patients are sent to the OR department and their operation is performed. After surgery, patients are transferred to a recovery unit for their recovery period. This will usually be the Post-Anaesthesia Care Unit (PACU), often called the *recovery room*, although some particular patients requiring acute care may be sent to the Surgical Intensive Care Unit (SICU). Finally, patients are sent to their post-operative unit where they will stay until they are fit enough to be discharged.

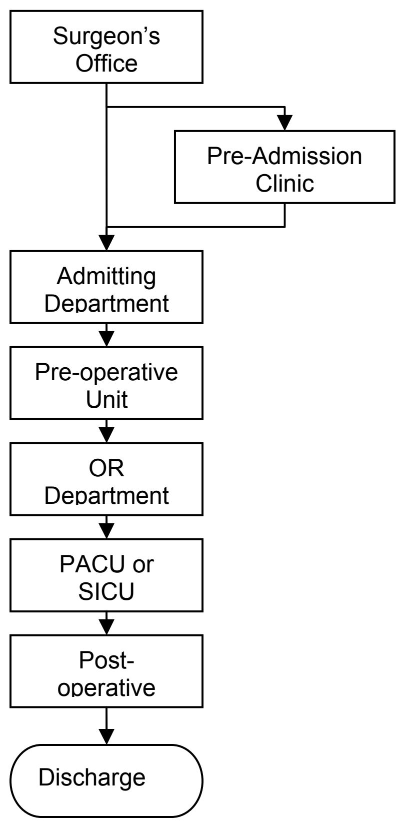

Figure 3-1 Flow of elective surgical patients at Health Sciences Centre (HSC)

Pre-operatively, SD and SDA patients will usually be sent to a unit called MS3, although a small number may go to a unit called B3. For Overnight patients, their pre-operative unit will be B3. Inpatients will go to the inpatient unit (i.e. ward) corresponding to their service, which depends on the type of surgery they require (e.g. cardiac, neurosurgery, etc.). If there are not enough beds in a particular service's unit, their patients may be placed "off service", meaning in a unit that belongs to a different service. For Inpatients, SD, and Overnight patients, their post-operative unit will be the same as their pre-operative unit. Post-operatively, SDA patients will be sent to the inpatient unit corresponding to their service.

3.2 Emergency Patients

Emergency surgical patients are those who need unanticipated surgery. They are classified into four categories, E1 to E4, depending on their acuity. E1 patients need surgery immediately, while E2, E3, and E4 patients require surgery within four, eight, and thirty-six hours, respectively.

Figure 3.2 depicts the flow of emergency surgical patients at HSC. The vast majority of HSC's emergency surgical patients arrive at the Emergency Room (ER) department. If patients need surgery immediately and OR time is available, they are sent to the OR department straight away. If not, they are admitted to a bed in an inpatient unit or the SICU pre-operatively, before being sent to the OR department when OR time is available. After surgery, these patients are transferred to the PACU or SICU, depending on their required level of care. Post-operatively, patients are transferred to an inpatient unit.

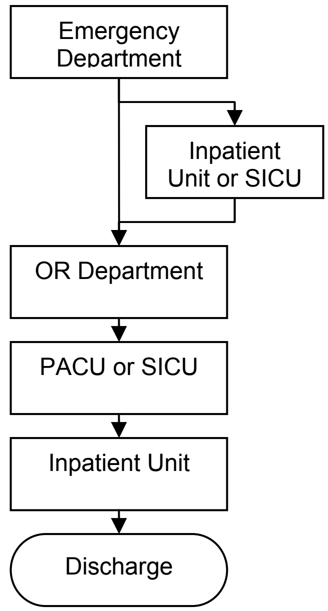

Figure 3-2 Flow of emergency surgical patients at Health Sciences Centre (HSC)

3.3 Pre-Admission Clinic

The Pre-Admission Clinic (PAC) is a clinic that elective surgical patients, excluding Inpatients, may be asked to attend prior to the day of surgery. The purpose of PAC is to assess a patient's medical fitness for surgery and anaesthesia, in addition to carrying out necessary pre-operative work. It is also an opportunity for patients and their families to be provided with information about what to expect before and after surgery.

When a patient is booked for surgery, PAC nurses will review the patient's case and set up an appointment for the patient if they feel it would be beneficial. Additionally, if a surgeon feels it is necessary for a patient to visit PAC, he/she may call PAC directly to arrange an appointment for their patient. Most patients can still proceed with their surgery, even if they have missed their PAC appointment.

Most patients visit PAC between a few days and two weeks of their scheduled operation, and the visit usually lasts between one and three hours. Each patient will be seen by a PAC nurse and an anaesthetist. The nurse will discuss the patient's health and medical history, the operation and necessary preparations, and the patient's discharge plans. The anaesthetist will assess the patient's medical condition, determine their fitness for surgery, and discuss anaesthetic options. If necessary, other professionals may also see the patient, such as a physiotherapist or cardiac surgery nurse. By the end of a patient's PAC visit, all required pre-operative investigations (e.g. blood tests, x-rays, etc.) should be arranged for or complete, and the patient should have a thorough understanding of what to expect with regards to surgery, anaesthesia, hospital stay, and discharge.

3.4 Admitting Department

On the evening before a patient's surgery, the patient will be called by a staff member from the Admitting department, who will register the patient over the phone. Registration involves obtaining standard data on a patient, such as the patient's name, date of birth, address, etc. On the day of admission, all elective surgical patients first stop at the Admitting department. If a patient was previously registered over the phone, the registration process at the Admitting department should only take a few minutes. Otherwise, the process will take no more than ten minutes unless there is a long line.

Excluding Inpatients, most patients are told to arrive about two hours prior to surgery, although some are asked to come earlier if they require additional pre-operative tests or medications. On the other hand, Inpatients are called on their scheduled admission day and only asked to arrive once an inpatient bed is available for them.

3.5 Pre-Operative Units

From the Admitting department, patients go to their pre-operative units. SD and SDA patients go to MS3 or B3, Overnight patients go to B3, and Inpatients go to inpatient units. If a patient has not arrived close to his/her scheduled OR time, the pre-operative unit will inform the OR and Admitting departments, who will then contact the patient.

3.5.1 MS3 and B3

MS3 is open from 5:30am to 6:00pm, Monday to Friday, while B3 and the inpatient units are open 24 hours a day. MS3 has seventeen beds and their patient volume is primarily made up of elective surgical patients. B3, on the other hand, is a unit that handles a wide variety of patient types. As such, one bed is reserved for SDA patients and three beds are reserved for Day surgery patients (SD or Overnight) every weekday. Since B3 only has three available beds for Day patients, priority is given to Overnight patients. If there are not enough B3 beds to accommodate all of the Overnight patients scheduled on a particular day, those patients are converted to SDA patients who will go to MS3 pre-operatively and an inpatient unit post-operatively.

3.5.2 Inpatient Units

There are many inpatient units at HSC, each corresponding to a particular service(s), and each having a different number of available beds. Some inpatient units contain step-down units (SDU), which are meant for patients with high acuity. Patients may be moved from a SDU bed to a regular inpatient unit bed or vice versa, depending on their acuity.

3.5.3 Pre-operative Assessments

All patients will be assessed by a nurse in their pre-operative unit before going to the OR department. A patient's assessment involves documenting the patient's vital signs, medical history, fitness for surgery, and concerns that the OR department should be aware of prior to surgery. On average, this assessment will take about five to ten minutes for patients who have been to PAC, and fifteen to twenty minutes for those who have not. In addition, a nurse must ensure that any remaining pre-operative tests are completed before the patient is sent to the OR department.

If a patient is not prepared for surgery, such as he/she did not fast or stop taking the required medications beforehand, the nurse must inform the patient's surgeon and anaesthetist. The anaesthetist, with input from the surgeon, will then decide whether or not surgery should proceed. Most of the time, the patient's operation will be moved to the end of the day and performed if there is enough available OR time.

3.6 Operating Room (OR) Department

The OR department at HSC has thirteen operating theatres. Each surgery will require the patient's surgeon(s) to be present, along with an anaesthetist and usually three OR nurses (complex cases may require more). The anaesthetist and OR nurses are assigned to a particular operating theatre for their whole shift, although they may be moved to different theatres under certain circumstances. For example, they may be moved to assist an emergency case or relieve someone else. In addition, there is an In-House Anaesthetist (IHA) whose main purpose is to be on hand for possible emergency cases. Since HSC is a teaching hospital, there may also be surgical or anaesthesia residents present during a case. Sometimes, people from other departments (e.g. Radiology) are also at hand.

Besides the surgical team, there are a lot of other people who contribute to patient flow through the OR department. Slating clerks schedule emergency cases and check that the next two day’s elective OR schedules have no resource conflicts. They work with Clinical Resource Nurses (CRNs) to ensure that there are no barriers (e.g. OR time, instrumentation, equipment, etc.) to carrying out each case. Two transport personnel, specifically assigned to the OR department, will be picking up elective patients from their pre-operative units (i.e. MS3, B3, inpatient units). Emergency patients, on the other hand, will be brought to the OR department by transport personnel working in other areas of the hospital. Supply clerks will be making sure that supplies and equipment are stocked up or on hand if required. Case cart personnel will make sure that all of the case carts needed for the day’s elective cases have been set up, before assembling the case carts required during the next day. In addition, they must also set up the case cart for any emergency case that arrives. Sterile Processing Department (SPD) personnel will be reprocessing (i.e. disassembling, sterilize, and reassembling) the case carts, equipment and instruments used during surgery. Meanwhile, peri-operative aides (PAs) will be helping with a variety of activities such as supporting the nurses with operating theatre set-ups, stocking supplies, transporting patients to their operating theatres from the holding area, assisting in the prepping of patients, and cleaning the operating theatres when cases are finished.

3.6.1 A Patient's Visit to the OR Department

Before a patient arrives at the OR, the nurses will have to set up the patient's operating theatre with the assistance of PAs. Once the theatre is ready, one of the nurses will call the patient's pre-operative unit and ask if the patient is ready. If yes, the OR nurse will inform the clerks at the OR department's front desk so that they can find a transport aide to pick up the patient from his/her pre-operative unit. The transport aide's journey from the OR department and back can take anywhere between ten minutes to half an hour depending on the route taken, the presence of family members, elevator waits, etc. On average, the journey takes about fifteen minutes.

In the OR department, the patient is brought to the holding area where a nurse from the patient's operating theatre will interview the patient. This may take ten or more minutes, depending on the patient, his/her case, and whether or not the necessary paperwork is complete. Afterwards, the anaesthetist from the patient's operating theatre will assess the patient, usually taking another ten minutes. The patient is then taken to the operating theatre by a PA, or a nurse if a PA is unavailable.

In the operating theatre, the patient is put to sleep by the anaesthetist, and prepped and positioned by the OR nurses, anaesthetist, and/or PAs. When surgery starts, at least one scrub nurse will set up the instruments and pass them to the surgeon, while the others nurses will be *circulating*, meaning that they will be carrying out any other required activities (e.g. giving supplies to the scrub nurse, charting, emptying the laundry, etc.).

When the case is nearly done, a circulating nurse will call the patient's recovery unit, to let them know that a patient is coming. If the recovery unit does not have any space (i.e. not enough beds), the patient will have to be held in the operating theatre until the recovery unit can accept the patient. The anaesthetist will then wake the patient up and take the patient to the recovery unit together with the surgeon. Meanwhile, the OR nurses in the patient's operating theatre will call for the next patient, in addition to putting away all the instrumentation and sending them to SPD. When they are finished, PAs clean up the operating theatre so it can be prepared for the next patient.

3.7 Recovery Units

3.7.1 Post-Anaesthesia Care Unit (PACU)

The Post-Anaesthesia Care Unit (PACU) is the main recovery unit that patients are transferred to immediately after surgery. It is usually able to accommodate a maximum of twelve patients at a time. Its main purpose is to prevent and treat complications following surgery and anaesthesia. When a patient arrives in the PACU, he/she will be assigned a nurse who will mainly provide care according to the anaesthetist's orders, although some of the surgeon's orders may also be attended to, such as dressing changes or the administering of medications. If the surgeon wants the patient to have some tests completed, such as x-rays, the PACU must make the required arrangements.

The time a patient spends in the PACU depends on the complexity of the patient's surgery and their own individual reaction to the surgery. Most patients stay in the PACU between two and four hours. Once a patient has met discharge criteria, the patient's nurse will call the patient's post-operative unit and confirm that the patient can be accepted there. The nurse will then give a verbal report to one of the nurses in that post-operative unit. Two nursing assistants (NAs) will then transport the patient to the post-operative unit. However, if the patient needs complete monitoring, has an intra-arterial catheter, is going to a SDU, or if there is only one NA available, a PACU nurse must help transport the patient. Meanwhile, other NAs will clean up the patient's area and prepare it for the next patient.

3.7.2 Surgical Intensive Care Unit (SICU)

A small number of elective surgical patients may require care from the Surgical Intensive Care Unit (SICU), which is a ten-bed unit similar to the PACU except they handle patients that have experienced greater trauma or major surgical procedures. When a patient is ready for discharge from the SICU and the patient's post-operative unit has a bed available, the patient's nurse will call the post-operative unit and give a verbal report to a nurse in that unit. The patient's SICU nurse, along with transport personnel, will then transfer the patient. Support staff, such as a NA, may follow if needed. When patients are transferred from the SICU, most of them will go to an inpatient unit, usually into a SDU.

3.8 Post-Operative Units

For Day patients and Inpatients, their post-operative unit will be the same as their pre-operative unit, while SDA patients will go to an inpatient unit. For patients who require intermediate care and are going to an inpatient unit, they may first be placed in a SDU until their condition is stable enough for a move to a regular inpatient unit bed.

Upon the arrival of a patient post-operatively, a nurse will be assigned to the patient and care is provided according to the surgeon's orders. Some surgeons may come to see their patients post-operatively, although it is not a requirement. If a patient is experiencing any complications or questions arise, the nurses contact the surgeons. It is usually easier to estimate the length of stay (LOS), in days, for elective patients than emergency patients. This is because most emergency patients are critically ill or have experienced major trauma.

Patients can be discharged when they are medically stable and their condition is not improving. Most patients are discharged home, although some may be sent back to the facility they were originally transferred from (e.g. rural hospital), or to rehabilitation hospitals. Other patients may be discharged to long term care facilities such as supportive housing, companion care, personal care homes (PCH) or chronic care hospitals. Discharge is fairly easy if the patient's case is straightforward.

If a SD or Overnight patient has not recovered sufficiently for discharge by the time MS3 closes or B3 has to admit new patients, he/she will need to be admitted to an inpatient unit. In this case, the nurse must first inform the patient's surgeon of the situation. If the surgeon agrees, he/she will inform the Admitting department who will in turn find a bed for the patient. Situations like this may occur because the patient should have been booked under a different admission type (e.g. Overnight rather than SD patient), the patient's recovery time was underestimated, or unexpected complications arose.

3.9 The OR Scheduling System at HSC

At HSC, there are two separate schedules for elective and emergency surgeries. During the week, from Monday to Friday, elective cases are scheduled from 7:30am to 3:30pm, while emergency cases are scheduled from 3:30pm to 10:30pm. On Wednesdays, however, elective cases start at 9:00am so that OR staff can attend their respective weekly teaching sessions beforehand. Usually, twelve operating theatres are opened and staffed for elective cases, while only two operating theatres are opened and staffed for emergency cases. On Saturdays and Sundays, two operating theatres are used for only performing emergency procedures throughout the day.

3.9.1 Elective OR Scheduling

At HSC, elective cases are scheduled using a block scheduling policy. First, a master surgical schedule (MSS) is created. This MSS consists of blocks depicting the number of operating theatres that will be opened and staffed on a particular day, along with each theatre's opening hours. The number of operating theatres opened on a given day is based upon anaesthesia availability, OR nursing staff availability, and sometimes PACU staff availability. During the winter, there may be up to thirteen operating theatres opened for elective cases, while during the summer there may be as little as four. The MSS is usually edited every four weeks by HSC management.

The blocks in the MSS may be full days (i.e. from 7:30pm to 3:30pm), half days, or some other variation. Each block is allocated to a particular service, usually based on historical and political reasons rather than on quantifying ones such as waiting list lengths. Often, certain services are regularly assigned the same operating theatres due to size and equipment requirements. Once these allocations have been completed, a copy of the MSS is sent to the managers of each service who break down their respective blocks into individual blocks belonging to a specific surgeon.

From there, surgeons schedule cases into their individual blocks in whatever manner they desire. However, they do have to follow the rule that SD patients (i.e. patients who will be discharged on the same day of surgery) should be scheduled earlier in the day, before their SDA patients. This rule is carried out in order to reduce the number of SD patients that may not be stable enough for discharge by the time MS3 closes. In addition, this allows extra time for patients to be discharged from the inpatient units, resulting in less schedule interruptions due to a lack of post-operative beds for arriving SDA patients.

Surgeons give estimates of the duration of their cases, including set-up and clean up times. If the OR department feels that a particular estimate is too optimistic, the surgeon will have his/her last case of the day put on standby, meaning that the OR department has the right to refuse to perform the case if it will likely result in overtime. Each surgeon must also finalize his/her scheduled cases at least thirty-six hours before the day of surgery. This rule allows the OR department to check the OR schedule for conflicts, such as operating theatre, instrumentation, and equipment limitations. If conflicts exist, some cases may be rescheduled. If a surgeon will not be using some of his/her allocated block time, the unused portion will be given to other surgeons.

In the MSS, blocks for emergency cases may be included. These blocks are solely devoted to pending emergency cases and are scheduled a day beforehand by the OR department. Some blocks are open for any type of emergency, while others may be dedicated to a specific service, such as one that frequently receives trauma cases.

On the day of surgery, an elective case may be cancelled due to a variety to reasons such as a lack of post-operative beds, lack of OR time, the patient did not show up, the patient was medically unfit for surgery, an emergency case replaced it, etc. If a case has to be cancelled, the patient may be converted to an emergency patient, depending on the urgency of his/her case. If not, the surgeon’s office will have to reschedule the patient on another day. When a case is cancelled, all of the elective cases scheduled behind it in the same operating theatre will be moved up earlier, if possible. If there is a gap in the elective OR schedule, it may be filled with a pending emergency case.

3.9.2 Emergency OR Scheduling

An emergency case is performed by the surgeon who initially saw the patient and decided that surgery was needed. Another surgeon may perform the case if the original surgeon will not be available to perform the surgery in a timely manner. On the weekdays, the OR department schedules two operating theatres with emergency cases from 3:30pm to 10:30pm. If a very urgent emergency case (e.g. E1) arrives before 3:30pm, during the time when elective cases are performed, the case will be performed in an unscheduled operating theatre (i.e. one that has not been staffed) provided that the surgeon is available, along with an anaesthetist and OR nurses who are taken from the other operating theatres in use. If there is no available operating theatre or anaesthetist, or if there are not enough available nurses, the emergency case will be performed by the OR team in the first operating theatre that finishes its current case. After 10:30pm, emergency cases are only performed if they are very urgent and cannot wait for the next day.

Any pending emergency case left at the end of the day will be carried over to the next day's emergency OR schedule. Sometimes, if an emergency case is quite urgent or if there is a gap in the elective OR schedule, the OR department may book the case into the elective OR schedule, at which point it is called a *pre-booked emergency*. On the weekends, two operating rooms are opened and staffed. Usually, one room is scheduled with pending emergencies, while the other is left unscheduled and used for any emergency arrivals.

3.10 The OR Scheduling System: Issues and Concerns

3.10.1 OR Time Allocations

The first issue regarding the current scheduling system is that OR time is allocated for elective and emergency cases based on historical and political reasons, rather than on quantifying ones. For example, OR time is not allocated for emergencies based on estimated future demand. When there is not enough time for emergencies, some emergency patients will occupy beds in the hospital waiting for surgery, subsequently affecting HSC's entire bed situation as units become back-logged. Some emergency patients have waited weeks for surgery. This is a particularly rampant problem for less urgent cases, such as E3 cases, because they have to wait until all of the more urgent cases are taken care of, even if those cases arrived after them.

Regarding elective cases, OR time is allocated to each service and surgeon without a consistent formula that considers the demand for urgent cases or the waiting time of patients on waiting lists. When patients' waiting times are not taken into account, some patients may unfairly have a longer waiting time than others. Furthermore, since each surgeon's demand for urgent cases is not taken into account, the scheduling policy has no room for flexibility. When a surgeon suddenly acquires a patient who needs surgery more urgently than another patient already booked into the elective schedule, he/she will often replace the booked patient with the new one, resulting in last minute changes.

3.10.2 Last Minute Changes to the Elective OR Schedule

Last minute changes are often made to the elective OR schedule, despite the thirty-six hour scheduling policy. This not only wastes previous work, but also creates extra work and increases the chances of delays and case cancellations because the scheduling policy is not flexible enough to handle these changes. For example, if a surgeon makes a last minute decision to schedule a patient one who needs to visit PAC beforehand, PAC must accommodate this extra patient on very short notice. Other patients in PAC may then experience long waiting times as this *add-on* patient is squeezed in, and some of them may even have their appointments rescheduled. Meanwhile, the Admitting department will have to update all of their computer records and paperwork. If any patients are subsequently assigned to different pre-operative units, the Admitting department must notify those units and ensure that they will receive the correct patient charts.

The pre-operative units get a copy of the elective OR schedule in the afternoon, on the day before surgery. If any last minute changes occur after this time and the Admitting department is not informed about them, the pre-operative units will also be unaware of those changes. Patients will then arrive at times unexpected to the pre-operative unit, and extra calls to the OR department will have to be made to clarify the changes.

Sometimes, even patients are not informed about changes to their OR time. If a patient's new OR time is later than before, the patient will arrive too early and experience a long waiting time. If a patient's new OR time is earlier than before, he/she will simply not show up. When the pre-operative unit discovers that the patient is late, assuming they are also aware of the change, they will have to inform the OR and Admitting departments so that the patient can be contacted. When the patient finally arrives, the pre-operative unit has to rush to prepare the patient so that surgery can start on time. However, the patient may simply not be able arrive in time, causing delays or cancellations.

Finally, last minute changes also affect the OR department because a lot of extra work is created. For example, new case carts have to be set up while the old ones are dismantled. Equipment and instrumentation requirements have to re-checked, potentially leading to more schedule changes. Moreover, nursing staff assignments must be reviewed.

3.10.3 Variable Elective Patient Volumes

One of the major systemic causes of poor surgical patient flow at HSC is that there is no central scheduling system, so there is no control over the mix of cases (i.e. number and type) performed. For example, all of the surgeons with OR time on a particular day may coincidentally decide to schedule only short cases, resulting in a high volume of patients. Other times, the surgeons may schedule only long cases, leading to a very low patient volume. This variability affects both the workload and bed situation throughout HSC.

3.10.3.1 Bed Situation

HSC's bed situation is hugely affected by the elective OR schedule's case mix. For example, since the PACU handles virtually all elective surgical patients, their bed occupancy volume is directly correlated with the elective OR schedule's patient volume. Similarly, MS3 is also greatly impacted because it handles the majority of SD and SDA patients pre-operatively, in addition to SD patients post-operatively. In the same way, bed occupancy in the inpatient units will be very high on days when many SDA patients and Inpatients are scheduled. This strains the inpatient bed situation, and some scheduled surgeries may be delayed in order to wait for discharges to occur so that beds will be available. Sometimes, cases may be cancelled if no beds become available.

When bed occupancy reaches its maximum limit in a particular unit, patients are put on hold. For example, when MS3 is full, patients cannot be transferred post-operatively from the PACU. In turn, the PACU may reach full capacity, putting patients on hold in the OR department. This causes surgeries to be delayed and patients are put on hold in their pre-operative unit. In another scenario, the inpatient units may reach full capacity, backing up the SICU who cannot transfer patients out. When the SICU becomes full, the overflow of patients and their nurses will usually move to the PACU. Similar to the previous example, the PACU may then become full, adversely affecting the OR department and pre-operative units. These types of situations result in delays, overtime, and elective case cancellations, thereby reducing staff and patient satisfaction. Surgeons do book their SDA patients later in the day so that discharges can occur by the time their patients require post-operative beds. However, delays and cancellations still occur.

3.10.3.2 Workload

Like the bed situation, the workload throughout HSC is also affected by variability in the elective OR schedule. On high volume surgical days, resources may be inadequate and lead to problems. For example, during a high volume elective surgical period, PAC experiences a heavier than average workload in the weeks leading up it. This translates into longer wait times for patients visiting PAC. Similarly, MS3's staffing remains constant on a daily basis and hence they may be short-staffed on days with a high volume of SD and SDA patients. This impacts staff satisfaction, may reduce the quality of patient care, and increases delays in getting patients to the OR department on time. In turn, surgeries may start late, leading to overtime or elective case cancellations.

When there is a high volume of elective surgical patients, there is a greater pace of activities in the OR department. There are more operating theatre turnovers, more preparation of equipment and instrumentation, etc. Similarly, the PACU may not have enough staff, leading to delays in transferring patients to their post-operative units. The OR department may then be unable to transfer patients to PACU, leading to more delays.

3.10.4 Case Duration Estimates

Surgeons estimate the durations of their cases, without the assistance of their historical times. As a result, surgeons may underestimate their case durations and schedule too many cases (i.e. overbooking). This leads to overtime for the OR nursing staff, may cause elective case cancellations, and may also cut into the emergency schedule. On the other hand, durations may be overestimated, resulting in underutilized OR time.

3.10.5 Large Deviations from Elective OR Schedules

Due to a wide variety of reasons, such as inaccurate case duration estimates and delays in the OR department, the OR schedule is far from a true depiction of what the day's schedule will really be like. Because it is hard to predict when a case will actually begin, especially if many delays have occurred, it is difficult for an operating theatre's team (i.e. surgeon, anaesthetist, OR nurses) to estimate when they have to be present and ready for each of their cases. It also makes it hard to gauge how many PAs are required for a particular day because each case's estimated finishing time will likely be inaccurate. Furthermore, these deviations make it difficult for the pre-operative units to assess patients in the order in which they will actually be sent to the OR department. Patients then experience long waiting times or have to be rushed once they arrive.

3.10.6 Delay Accumulation

Many of the issues mentioned demonstrate how the activities in one department have a substantial impact on the activities in other departments. Most individual delays are small. However, they are often correlated and can quickly accumulate, leading to vast amounts of wasted time. For example, surgeons may leave the OR department between their cases because they do not want to wait while their operating theatres are cleaned. Turnover delays, such as an insufficient number of PAs, may also prompt surgeons to leave the OR department. If surgeons have experienced enough turnover delays, they may begin to anticipate them and schedule activities elsewhere during turnovers. This can create more delays if the surgeons get caught up in these activities and return to the OR department later than expected.

3.11 Suggestions for Improving the OR Scheduling System

It is clear from the issues mentioned above that the OR scheduling system used at HSC is the major obstruction to smooth surgical patient flow in their facility. Indeed, Haraden and Resar (2004) point out that although most admissions to a hospital are from the emergency department, the elective surgical admissions often have a greater effect on patient flow due to the arbitrary nature of elective scheduling. This section outlines a few suggestions for improving HSC's OR scheduling system.

3.11.1 Control of Elective Case Mix

If the number and type of elective cases at HSC could be controlled, the bed occupancy and patient volumes in the pre-operative, recovery, and post-operative units can be smoothed. This will reduce elective case delays, overtime and elective case cancellations. Furthermore, day-to-day staffing needs will be more consistent and easier to predict, increasing staff satisfaction and perhaps quality of patient care. The throughput of elective surgical cases may even increase. One way of controlling the case mix would be to implement a centralized scheduling system.

3.11.2 Usage of Consistent OR Time Allocation Formulas

Currently, elective OR time is allocated to each service based on historical and political reasons. This should be changed so that allocations reflect the actual needs of the different services and surgeons, such as waiting times for their patients. Furthermore, historical data should be collected regarding emergency surgical arrivals for each service, which can be used to set aside appropriate OR time so that the supply actually matches demand. Elective schedule disruptions due to emergencies will reduce, and less inpatient unit beds will be tied up by emergency patients waiting for surgery.

3.11.3 Enforcement of Booking Deadlines

There is currently a thirty-six hour scheduling policy whereby elective cases must be scheduled at least a day and a half before surgery. However, this policy is not enforced and last minute changes often occur, affecting many departments. Obviously, this policy was made to avoid these last minute changes and HSC should take steps to make sure it gets followed. Perhaps the policy can be modified so that services have different booking deadlines, dependant on the types of cases they typically receive.

3.11.4 Anticipation of Urgent or Emergent Case Arrivals

Last minute changes to the elective OR schedule often occur because surgeons replace scheduled patients with new, more urgent ones. To reduce the problems caused by these changes, perhaps blocks of flexible OR time should be included in the MSS. Surgeons may then book urgent cases into these blocks, rather than exchanging patients they have already scheduled. This will help incorporate flexibility into the scheduling system.

Another issue is that elective OR schedule disruptions are often caused by emergency case arrivals. Currently, HSC's MSS includes dedicated emergency blocks. However, these blocks are not assigned to days and times based on historical data. Therefore, they are not positioned in the most effective way to reduce elective case disruptions. HSC should collect historical data on the frequency of emergency case arrivals so that emergency blocks can be positioned accordingly.

Similarly, HSC should consider changing the scheduled times for elective and emergency cases. For example, instead of opening twelve elective operating theatres from 7:30am to 3:30pm and two emergency operating theatres from 3:30pm 10:30pm, maybe eleven elective operating theatres can be opened from 7:30am to 4:30pm and one emergency operating theatre can be opened from 8:30am to 10:30pm. This may also help reduce elective case disruptions.

3.11.5 Use of Historical Data for Case Duration Estimates

Currently, surgeons estimate the durations of their cases without the assistance of historical data, which often leads to overbooking or underutilized OR time. Broka et al. (2003) demonstrated that using historical data would substantially reduce overtime compared to simply using subjective estimates given by surgeons. Moreover, Wright et al. (1996) showed that if surgeons are prompted with historical data, they can provide estimates that are only slightly less accurate than good statistical methods. Therefore, historical data should be used to assist the estimation of case durations.

Accurate case duration estimates will reduce deviations between scheduled and actual OR times, leading to many benefits. Patients will experience less variable pre-operative wait times, the pre-operative units will be able to assess patients in the correct order, and potential equipment and instrumentation conflicts may be more accurately predicted. Each member of the OR team (i.e. surgeons, anaesthetists, nurses) will have a better indication of when they must be ready for each of their cases, and the required number of PAs in the OR department can be more precisely anticipated, thereby reducing turnover delays. Surgeons will then have fewer reasons to schedule activities during turnovers.

3.12 Summary

Health Sciences Centre (HSC) in Winnipeg handles a large portion of surgical patients in Manitoba, Northwestern Ontario and Nunavut. Each surgical patient is classified as either elective or emergent and belongs to a particular service (e.g. neurosurgery, dental, etc.), dependant on their type of surgery. Elective patients are classified as Inpatients, Same

Day (SD) patients, Overnight patients, or Same Day Admission (SDA) patients. Their classification dictates the path they will take through HSC. On the other hand, emergency patients are classified into four different categories, E1 to E4, based on their acuity. During the week, elective surgeries are performed in the morning and early afternoon, while emergency surgeries are carried out late in the afternoon and at night. Of course very urgent emergency operations are performed at any time. On the weekends, only emergency procedures are completed.

HSC's operating room (OR) scheduling policy has a major impact on the facility's surgical patient flow. There are six main issues regarding this policy. First, OR time is not set aside for emergency cases based on expected demand, and elective OR time is not consistently allocated to each service and surgeon based on quantifying reasons. Second, last minute changes are often made to the elective OR schedule, which wastes previous work, creates extra work, and increases the chances of delays. Third, there is no control over the elective case mix, so daily elective bed occupancy and patient volumes are often highly variable. Fourth, case duration estimates are not made based on, or even assisted by, historical data. Fifth, there are often big deviations between actual and scheduled case start times, making it difficult for the pre-operative units and the OR department to determine required staffing amounts and patient ready times. Finally, delays are usually correlated and have the tendency to spiral out of control.

To improve the surgical patient flow at HSC, the OR scheduling policy needs to be revised. Potential methods of achieving this are to allocate OR time based on demand and other pertinent factors such as waiting times, use a centralized booking system for elective cases so that the case mix can be controlled, and start using historical data for case duration estimates. In addition, HSC should consider aspects that will help incorporate flexibility into the scheduling system, such as creating blocks for urgent or emergent cases during the elective schedule in order to minimize elective case disruptions.

Chapter 4: Elective Operating Room (OR) Scheduling System – Stage 1

This chapter describes the first stage of the proposed elective operating room (OR) scheduling system for Health Sciences Centre (HSC). The system follows an open scheduling policy, where weekly OR schedules are generated. In this stage, surgeons are allocated hours of OR time during the week to be scheduled. Each surgeon then presents a list of cases that they wish to perform during that week, provided that the total duration does not exceed their OR time allocation. These cases are then assigned to different days of the week, using multiple objectives that aim to reduce day-to-day variability while respecting constraints. Due to the distinct priorities of each objective, the problem was mathematically modelled and solved using lexicographic goal programming.

Section 4.1 describes the problem's objectives, section 4.2 explains the input data required by the model, and section 4.3 details the assumptions that the model is based upon. Section 4.4 portrays the notations and equations making up the mathematical model. Section 4.5 presents the problems that were solved by the mathematical model, their associated results, and a performance evaluation between the actual schedules used at HSC. Finally, section 4.6 summarizes the chapter.

4.1 Objectives

There are four objectives in this stage, classified into three priority levels. The first priority level is the most important, followed by the second level, and finally the third. These objectives, and their priority levels, were determined after assessing the factors affecting HSC's surgical patient flow. If desired, objectives can easily be excluded, or their corresponding priorities can be changed.

4.1.1 Priority Level One

The first priority level consists of the following two objectives, related to bed occupancy resulting from the elective OR schedule:

1. Minimize the largest deviation between the number of inpatient beds occupied by any particular service on any two days of the scheduled week
2. Minimize the total number of inpatient beds occupied on the days following the scheduled week

Inpatient beds are the beds contained in the inpatient units, which are occupied by SDA patients (who need to stay in the hospital for more than 1 day after surgery) and Inpatients (who need to be admitted to the hospital at least 1 night prior to surgery). The two objectives in this priority level aim to efficiently manage inpatient bed resources by keeping their utilization high while minimizing variability in daily bed occupancy volumes. These objectives were considered the most important because HSC suffers from high variability in day-to-day bed occupancy volumes due to their elective schedules.

On days with high bed occupancy, delays often occur because there are not enough post-operative beds available for elective patients. Some patients may have their operations delayed, or find themselves put on hold in their operating theatre or recovery unit. Consequently, this can lead to overtime in the OR department, or some elective cases may be cancelled. On other days, however, HSC's bed occupancy can be very low. This indicates that daily bed occupancy volumes need to be smoothed, and hence the first objective was included in order to achieve this.

The second objective reduces the number of inpatient beds occupied by patients on the days following the scheduled week. In this way, bed utilization will be high during the scheduled week, and future schedules can be created with minimal disturbance from previous weeks.

4.1.2 Priority Level Two

Priority level two consists of one objective, as follows:

- Minimize the largest deviation between utilized and available OR time on any day of the scheduled week

The intention of this objective is to reduce underutilized OR time and overtime in order to ensure balanced daily OR utilization. This is important because overtime results in extra incurred costs and impacts staff satisfaction. On the other hand, underutilized time must be avoided in order to make sure that resources are not wasted.

4.1.3 Priority Level Three

The last priority level consists of the following objective:

- Minimize the largest deviation between the total number of Same Day (SD) and Same Day Admission (SDA) cases on any two days of the scheduled week

Elective surgical patients at HSC are classified as Inpatient, SD, Overnight, or SDA. SD and SDA patients make up the vast majority of HSC's elective surgical patients, and virtually all of them will be sent pre-operatively to the unit MS3. MS3's daily staffing numbers remain relatively constant, and hence they are often short-staffed on days with high patient volumes, which delays getting patients to the OR on time. Therefore, smoothing their daily patient volumes will make it easier to predict their required staffing numbers, which can then be adjusted to adequately meet demand. Furthermore, since most elective patients are SD or SDA, smoothing their daily numbers will help smooth the workload for every unit involved in patient flow.

4.2 Input Data

Table 4-1 depicts the input data needed for this stage, and is split up according to whether HSC already uses the data for elective OR scheduling.

Table 4-1 Input data required for stage 1

Source	Data already used by HSC	Data not currently used by HSC
Surgeons	• The cases to be scheduled: — Estimated case duration (including set-up and clean up) — Service type (e.g. Dental, Cardiac, etc.)	• The days they are available for surgery • The cases to be scheduled: — Estimated recovery time in the recovery unit (i.e. PACU) — Estimated recovery time in the post-operative unit, if less than 24 hours — Estimated length of stay (LOS) — A note if a patient cannot be placed off-service — A note if a patient must be classified as an Inpatient, along with the number of pre-operative days required — The days a patient is absolutely unavailable for surgery
OR Dept.	• The operating theatres that can be used for each case • The hours of OR time that each surgeon has been allocated • The number of equipment sets available for each service, on each day • The opening and closing times for each staffed operating theatre, on each day	• The maximum amount of overtime allowed for any operating theatre, as decided by management • The maximum difference in the number of patients allowed between the day with the highest patient volume and the day with the lowest patient volume[1].
Inpatient Units		• The number of available elective surgical beds for each service, on each day • The number of off-service patients allowed for each service, before elective case cancellations start being considered by management

[1] This will be used to keep the variability in overall daily patient volumes below a certain level.

Source	Data already used by HSC	Data not currently used by HSC
MS3	• The unit's daily closing time, where patients must be discharged	
B3	• The daily number of available elective surgical beds for elective Overnight patients • The time when Overnight patients must be discharged in order to admit new patients	

Obtaining the required input data will not be a problem for the information already used by HSC for elective OR scheduling. The other data that is not currently used by the facility, such as estimated recovery times and LOS, can easily be estimated by surgeons. Moreover, historical data can be retrieved for this purpose if they have been recorded in a database. Data on the number of beds available for each service can easily be obtained because HSC does have bed allocations for each service, although they are not currently used during the scheduling process or split into elective or emergency designations. Finally, the remaining data required, such as the maximum allowable amount of overtime or daily elective patient volume variability, can simply be specified by management.

4.3 Assumptions

At HSC, elective surgical patients are classified as Inpatient, SD, Overnight, or SDA. If a patient needs to be admitted a day or more pre-operatively, that patient can only be classified as an Inpatient. The three other patient types are all admitted on the day of surgery, and their classification depends on their estimated LOS. SD patients are expected to be discharged on the same day of surgery, while Overnight patients are expected to be discharged the morning after surgery. Finally, SDA patients are expected to be discharged at least twenty-four hours after surgery.

B3, the unit that handles Overnight patients, can only admit up to three Overnight patients on each day. Therefore, there cannot be more than three Overnight patients scheduled each day. All remaining patients who fall under the definition of an Overnight patient will instead have to be classified as SDA patients.

Pre-operatively, Inpatients go to an inpatient unit corresponding to their service, Overnight patients go to B3, and SD and SDA patients go to MS3. From their pre-operative units, all patients go to the OR department for surgery, before being transferred to the Post-Anaesthesia Care Unit (PACU) for recovery. The Surgical Intensive Care Unit (SICU) will not be considered because it only admits a very small percentage of elective surgical patients. Finally, SDA patients go to the inpatient unit corresponding to their service, while all others return to their pre-operative units. It is assumed that no other transfers take place before discharge.

Many surgeons have offices nearby or at HSC, and they visit patients throughout the facility when they are not operating. Therefore, it is assumed that each surgeon is able to operate at any time on the days they stipulate they are available.

4.4 Mathematical Model for Stage 1

The following notations are employed in the mathematical model:

MS3 Closing time of MS3

B3 Time at which new Overnight patients must be admitted in B3

Inp_i 1 if the patient of case i must be scheduled as an Inpatient; 0 otherwise

where i = 1,…,n (n = number of cases to be scheduled)

Pre_{ih} 1 if the patient of case i requires to be admitted on day h; 0 otherwise

where h = 1,…,e (e = number of days before surgery is scheduled)

Dur_i Surgical duration of case i, in half hours[2]

Rec_i Recovery duration of case i, in half hours

Post_i Post-operative duration of case i, in half hours, if less than 24 hours; 50 otherwise[3]

Tot_i Total duration of case i, including surgery, recovery, and post-operative care ($Dur_i + Rec_i + Post_i$)

LOS_i Post-operative length of stay for case i, in days, including the day of surgery

OFF_i 1 if the patient of case i cannot be placed off-service post-operatively; 0 otherwise

Ser_{ik} 1 if case i is primarily under service k; 0 otherwise

Sur_{is} 1 if case i is under surgeon s; 0 otherwise

where s = 1,…,p (p = number of surgeons with cases)

All_s Total OR time allocated to surgeon s in the scheduled week, in half hours

Over Daily number of beds available for Overnight patients in B3

---

[2] For example, if a case's duration is 2 hours, then Dur_i will be equal to 4.

[3] If a patient needs to stay for more than 24 hours post-operatively, the number of post-operative hours (i.e. $Post_i$) is not needed in the model's calculations. Instead, the number of post-operative days (i.e. LOS_i) is required. Therefore, all patients who need to stay for more than 24 hours post-operatively are simply assigned a $Post_i$ value of 50.

Bed_{gk} Number of beds available on day g for cases under service k

where $g = 1,\ldots,t$ (t = number of days being considered)

where $k = 1,\ldots,f$ (f = number of services)

ψ_{gk} Number of beds occupied on day g by patients under service k, scheduled during the previous week

φ_{gk} Number of beds occupied on day g by patients under service k who cannot be placed off-service, scheduled during the previous week

A_{il} 1 if the patient of case i will occupy a bed on day l; 0 otherwise

where $l = 1,\ldots,q$ (q = number of days after surgery, including the day of surgery)

B_{js} 1 if surgeon s is available on day j; 0 otherwise

C_{ij} 1 if the patient of case i is available on day j; 0 otherwise

D_{jo} Total OR time available on day j for operating theatre o, in half hours

where $j = 1,\ldots,m$ (m = number of days in the schedule)

where $o = 1,\ldots,r$ (r = number of staffed operating theatres)

Max_j Maximum OR time available by any operating theatre on day j, in half hours

Ω Maximum amount of overtime allowed by any operating theatre, in half hours

Ava_j The total OR time available on day j, in half hours

OR_{io} 1 if case i can be performed in operating theatre o; 0 otherwise

ORs_i Number of operating theatres that case i can be performed in

Equ_{jk} Number of equipment/instrumentation sets available on day j for cases under service k

ω Number of off-service patients allowed for each service, before elective case cancellations start being considered

Λ Maximum daily variability allowed for elective patient volumes

B1a Goal for b_{1a}

B1b Goal for b_{1b}

The following variables represent the mathematical model's solution:

b_{1a} Maximum difference between the number of beds occupied by a particular service on any two days of the scheduled week

b_{1b} Total number of beds occupied during the days following the scheduled week

b_2 Largest deviation between the actual and available OR time experienced on any day of the schedule

b_3 Largest deviation between the total number of SD and SDA cases on any two days of the schedule

w_{ij} 1 if case i is assigned to day j; 0 otherwise

x_i 1 if the patient of case i is scheduled as a SD patient; 0 otherwise

y_i 1 if the patient of case i is scheduled as a Overnight patient; 0 otherwise

z_i 1 if the patient of case i is scheduled as a SDA patient; 0 otherwise

u_{gk} Number of beds occupied on day g by patients from service k, who cannot be placed off-service post-operatively

v_{gk} Number of beds occupied on day g by patients from service k

use_j Total OR time used on day j, in half hours

ms_j Total number of SD and SDA cases scheduled on day j

num_j Number of cases scheduled on day *j*

The following variables ensure a linear mathematical model:

xa_{ij} 1 if case *i* is assigned to day *j* and scheduled as an SD patient; 0 otherwise

ya_{ij} 1 if case *i* is assigned to day *j* and scheduled as an Overnight patient; 0 otherwise

za_{ij} 1 if case *i* is assigned to day *j* and scheduled as an SDA patient; 0 otherwise

a^- Negative deviation between the number of cases scheduled on days *j* and *c*

a^+ Positive deviation between the number of cases scheduled on days *j* and *c*

b_{1a}^- Negative deviation between the number of beds occupied on days *g* and *a* by patients from service *k*

b_{1a}^+ Positive deviation between the number of beds occupied on days *g* and *a* by patients from service *k*

b_2^- Negative deviation between the actual and available OR time on day *j*

b_2^+ Negative deviation between the actual and available OR time on day *j*

b_3^- Negative deviation between the number of SD and SDA cases scheduled on days *j* and *c*

b_3^+ Positive deviation between the number of SD and SDA cases scheduled on days *j* and *c*

Objective Function:

Priority Level 1: Minimize $z = 2 * b_{1a} / B1a + b_{1b} / B1b$ (1)

Priority Level 2: Minimize $z = b_2$ (2)

Priority Level 3: Minimize $z = b_3$ (3)

Subject to:

$$w_{ij} Sur_{is} \le B_{js} \qquad \forall i \in [1..n], \forall j \in [1..m], \forall s \in [1..p] \qquad (4)$$

$$w_{ij} \le C_{ij} \qquad \forall i \in [1..n], \forall j \in [1..m] \qquad (5)$$

$$Max_j \ge D_{jo} \qquad \forall j \in [1..m], \forall o \in [1..r] \qquad (6)$$

$$\sum_{i=1}^{n} w_{ij} Sur_{is} Dur_i \le Max_j + \Omega \qquad \forall j \in [1..m], \forall s \in [1..p] \qquad (7)$$

$$\sum_{i=1}^{n} Sur_{is} Dur_i \le All_s \qquad \forall s \in [1..p] \qquad (8)$$

$$\sum_{i=1}^{n} w_{ij} Ser_{ik} \le Equ_{jk} \qquad \forall j \in [1..m], \forall k \in [1..f] \qquad (9)$$

$$Ava_j = \sum_{o=1}^{p} D_{jo} \qquad \forall j \in [1..m] \qquad (10)$$

$$\sum_{i=1}^{n} D_i \le \sum_{j=1}^{m} Ava_j \qquad (11)$$

$$\sum_{\substack{i=1 \\ ORs_i=1}}^{n} w_{ij} Dur_i OR_{io} \le D_{jo} \qquad \forall j \in [1..m], \forall o \in [1..r] \qquad (12)$$

$$\sum_{i=1}^{n} ya_{ij} \le OVER_j \qquad \forall j \in [1..m] \qquad (13)$$

$$Tot_i = Dur_i + \mathrm{Re}\, c_i + Post_i \qquad \forall i \in [1..n] \qquad (14)$$

$$ya_{ij} = 0 \qquad \forall i \in [1..n] / Inp_i = 0 / Tot_i \ge B3, \forall j \in [1..m] \qquad (15)$$

$$xa_{ij} + ya_{ij} = w_{ij} \qquad \forall i \in [1..n] / Inp_i = 0 / Tot_i < MS3, \forall j \in [1..m] \qquad (16)$$

$$ya_{ij} + za_{ij} = w_{ij} \qquad \forall i \in [1..n] / Inp_i = 0 / Tot_i \ge MS3, \forall j \in [1..m] \qquad (17)$$

$$xa_{ij} \le w_{ij} \qquad \forall i \in [1..n], \forall j \in [1..m] \qquad (18)$$

$$ya_{ij} \le w_{ij} \qquad \forall i \in [1..n], \forall j \in [1..m] \qquad (19)$$

$$za_{ij} \le w_{ij} \qquad \forall i \in [1..n], \forall j \in [1..m] \qquad (20)$$

$$x_i = \sum_{j=1}^{m} xa_{ij} \qquad \forall i \in [1..n] \qquad (21)$$

$$y_i = \sum_{j=1}^{m} ya_{ij} \qquad \forall i \in [1..n] \tag{22}$$

$$z_i = \sum_{j=1}^{m} za_{ij} \qquad \forall i \in [1..n] \tag{23}$$

$$u_{gk} = \sum_{\substack{i=1 \\ OFF_i=1}}^{n} \sum_{j=1}^{m} \left(w_{ij} - xa_{ij} - ya_{ij}\right) Ser_{ik} \left(\sum_{d=dist.(g,j)}^{w} LOS_{id} \right) + \sum_{\substack{i=1 \\ Inp_i=1 \\ OFF_i=1}}^{n} \sum_{j=1}^{m} w_{ij} Ser_{ik} \left(\sum_{e=dist.(g,j)}^{w} \Pr e_{ie} \right) + \psi_{gk}$$

$$\forall g \in [1..t], \forall k \in [1..f] \tag{24}$$

$$v_{gk} = \sum_{i=1}^{n} \sum_{j=1}^{m} \left(w_{ij} - xa_{ij} - ya_{ij}\right) Ser_{ik} \left(\sum_{d=dist.(g,j)}^{w} LOS_{id} \right) + \sum_{\substack{i=1 \\ Inp_i=1}}^{n} \sum_{j=1}^{m} w_{ij} Ser_{ik} \left(\sum_{e=dist.(g,j)}^{w} \Pr e_{ie} \right) + \varphi_{gk}$$

$$\forall g \in [1..t], \forall k \in [1..f] \tag{25}$$

$$u_{gk} \leq Bed_{gk} \qquad \forall g \in [1..t], \forall k \in [1..f] \tag{26}$$

$$\omega \geq v_{gk} - Bed_{gk} \qquad \forall g \in [1..t], \forall k \in [1..f] \tag{27}$$

$$num_j = \sum_{i=1}^{n} w_{ij} \qquad \forall j \in [1..m] \tag{28}$$

$$a^- - a^+ \geq num_j - num_c \qquad \forall j, c \in [1..m] / \; j > c \tag{29}$$

$$\Lambda \geq a^- + a^+ \tag{30}$$

$$b_{1a}^{-} - b_{1a}^{+} \geq v_{gk} - v_{ak} \qquad \forall g, a \in [1..7] / g > a, \forall k \in [1..f] \qquad (31)$$

$$b_{1a} \geq b_{1a}^{-} + b_{1a}^{+} \qquad (32)$$

$$b_{1b} = \sum_{g=8}^{t} \sum_{k=1}^{f} v_{gk} \qquad (33)$$

$$use_j = \sum_{i=1}^{n} w_{ij} Dur_i \qquad \forall j \in [1..m] \qquad (34)$$

$$b_2^{-} - b_2^{+} \geq use_j - Ava_j \qquad \forall j \in [1..m] \qquad (35)$$

$$b_2 \geq b_2^{-} + b_2^{+} \qquad (36)$$

$$ms_j = \sum_{i=1}^{n} x_{ij} + z_{ij} \qquad \forall j \in [1..m] \qquad (37)$$

$$b_3^{-} - b_3^{+} \geq ms_j - ms_c \qquad \forall j, c \in [1..m] / j > c \qquad (38)$$

$$b_3 \geq b_3^{-} + b_3^{+} \qquad (39)$$

$$\sum_{j=1}^{m} w_{ij} = 1 \qquad \forall i \in [1..n] \qquad (40)$$

$$Inp_i + x_i + y_i + z_i = 1 \qquad \forall i \in [1..n] \tag{41}$$

$$w_{ij}, x_i, y_i, z_i, xa_{ij}, ya_{ij}, za_{ij}, \omega_{jp} \in \{0,1\} \ \forall i \in [1..n], \forall j \in [1..m] \tag{42}$$

The multi-objective optimization method used to solve this mathematical model is lexicographic goal programming. In lexicographic goal programming, objectives are divided into different priority levels according to their importance. Goals in a lower priority level are infinitely more important than all goals in higher priority levels. Hence, no trade-offs between the goals in different priority levels are allowed. A lexicographic goal programming problem is first solved by only considering the goals in the lowest (i.e. most important) priority level. The resulting solutions for those goals are then turned into equality constraints, and the problem is solved for the goals in the second priority level. In this way, the obtained solution will not violate the goals achieved for the first priority level. This process continues until the goals in the last priority level have been addressed.

The goals in the mathematical model are split into 3 priority levels. In order to generate a schedule for stage 1, the model will therefore need to be executed 3 times. Equations (1-3) represent the objective functions for priority levels 1-3, respectively. In equation (1), b_{1a} and b_{1b} represent the two priority level one objectives. Based on how delays and elective surgical case cancellations can occur at HSC, b_{1a} was given a weight of 2 because it was considered twice as important as b_{1b}. The objectives in priority levels two and three are represented by b_2 and b_3 in equations (2) and (3), respectively.

Equations (4–5) prevent a case from being scheduled on a day when the corresponding surgeon(s) or patient is unavailable. Equation (6) calculates the largest amount of OR time available in an operating theatre on each day of the schedule. Equation (7) specifies that the total duration of all cases belonging to a surgeon on a particular day cannot exceed the amount found in equation (6), plus the maximum amount of allowable overtime. Equation (8) ensures that the OR time required for all of a surgeon's cases does not exceed the amount allocated to him/her. Equation (9) allows cases to only be scheduled on days when the required equipment/instrumentation is available. Equation (10) calculates the total OR time needed for all cases that must be scheduled, and equation (11) checks that this value does not exceed the total available OR time. For cases that can only be performed in the same operating theatre, equation (12) ensures that if those cases are assigned to the same day, their total surgical duration does not exceed the total OR time available for that particular operating theatre on that day. Equation (13) specifies that the total number of Overnight patients scheduled on any day cannot exceed the number of B3 beds available. Equations (14–23) determine whether a case, excluding ones belonging to Inpatients, should be scheduled as SD, Overnight, or SDA.

For each service, equation (25) calculates the number of inpatient beds that will be occupied on each day, while equation (24) determines the same information for only patients who cannot be placed off-service. In those equations, d is the time between days g and j, defined as $(g - j + 1)$ if day j precedes day g, 0 otherwise. Similarly, e is the time between days g and j, defined as $(j - g)$ if day g precedes day j, 0 otherwise. Equation

(26) ensures there is an inpatient bed available for each patient that cannot be placed off-service. For each service, equation (27) prevents the number of off-service patients from exceeding the threshold at which elective case cancellations are considered. Equation (28) calculates the number of patients scheduled on each day, and equations (29–30) prevent these numbers from exceeding the maximum daily variability allowed.

Equations (31–32) and (33) determine the values obtained for the first and second objectives in priority level one, respectively. Equation (34) calculates the total OR time that will be used on each day of the schedule, while equations (35–36) establish the priority level two objective value. Equation (37) determines the number of SD and SDA cases scheduled on each day, while equations (38–39) establish the priority level three objective value. Finally, equations (40–42) are integrality constraints.

Because there are two objectives in priority level one, their values have to be normalized before weights can be applied. The terms *B1a* and *B1b* in equation (1) are included for this purpose. *B1a* represents the best value that can be achieved for objective b_{1a} when no other objectives are considered. Similarly, *B1b* corresponds to the best value that can be attained for objective b_{1b} when no other objectives are considered. To find the value of *B1a*, the model must be run without equations (34–40), and equation (1) has to be changed to include only b_{1a}. The b_{1a} value obtained should be used as *B1a* in the original equation (1). Similarly, the model must be run without equations (32–33,35–40) and equation (1) should only include b_{1b}, in order to find *B1b*.

When the model is run for the objectives in priority level one, equations (35–40) are not included in the model because they have no bearing on the outcome of the solution, and will only needlessly increase the number of variables and constraints that are considered. Similarly, equations (38–40) are not required when the model is run for priority level two. When the model is run for priority level three, all equations must be included.

4.5 Execution of the Mathematical Model for Stage 1

The mathematical model was executed using Lingo[4], which is a solver for linear optimization models. First, tables were created in Microsoft Office Excel[5]. These tables were used to arrange input data from which the Lingo solver could read from, in addition to providing a place for solution values to be stored.

4.5.1 Test Problems

To determine whether the proposed mathematical model could satisfactorily find feasible solutions, three basic problems were created and solved. As shown in Table 4-2, the problems differed in the number of cases to be scheduled (n), along with the number of surgeons (p), services (f), and operating theatres (o) to consider. The solutions were manually validated to confirm that all constraints and assumptions were satisfied.

[4] Copyright © LINDO Systems Inc
[5] Copyright © Microsoft Corporation

Table 4-2 Parameters of each test problem

	Test 1	Test 2	Test 3
n	50	100	150
p	10	25	40
f	4	8	12
o	4	6	8

Table 4-3 shows the values of *B1a* and *Blb* determined by the model, along with the computational times (*CPU*) required by the Lingo solver.

Table 4-3 *Bla* and *B1b* values determined for each test problem

	Test 1	Test 2	Test 3
B1a	2	3	3
CPU (sec)	4	6	35
Blb	31	47	53
CPU (sec)	1	3	14

Table 4-4 depicts the computational results for each test. The term *Z* corresponds to the objective values obtained and *B* gives the integer programming (IP) bound (i.e. the bound on the best possible value the objective can attain) reported by the solver. The global optimum was found for each priority level in each problem.

Table 4-4 Test results for stage 1

Test	Priority Level	Objective Function	No. of Variables	No. of Constraints	CPU (sec)	Z	B
1	1	$2*b_{1a}/B1a + b_{1b}/B1b$	3811	2556	6	3.032	3.032
	2	b_2	3825	2571	518	12	12
	3	b_3	3880	2626	442	0	0
2	1	$2*b_{1a}/B1a + b_{1b}/B1b$	7553	5057	21	3.021	3.021
	2	b_2	7567	5072	30	9	9
	3	b_3	7622	5127	235	2	2

3	1	$2*b_{1a}/B1a + b_{1b}/B1b$	11150	7423	47	3.	3.
	2	b_2		7438	1205	4	4
	3	b_3		7692	838	0	0

4.5.2 Health Sciences Centre (HSC) Problems

To evaluate the mathematical model, actual data from five consecutive weeks of elective surgical cases was collected from HSC in order to create five different problems. The elective cases performed in each week, excluding pre-booked emergencies, represent the cases that need to be scheduled. For each case, their surgeon(s), service, and estimated surgical duration was recorded. If a patient was admitted prior to the day of surgery, the number of pre-operative days spent in the hospital was rounded to the nearest day. Because these patients are rarely asked to arrive more than a few days prior to surgery, the maximum number of pre-operative days considered was 4.

Since estimated recovery times are not currently used when scheduling at HSC, the actual recovery times were rounded to the nearest half hour and used in the model. For patients who had surgery under local anaesthetic, their recovery duration was set to 0. For each patient, their LOS was rounded to the nearest full day. Because most elective patients do not stay in the hospital for more than two weeks after surgery, the maximum LOS considered was 14. Each patient's post-operative duration was also recorded if a patient was discharged within 24 hours after surgery. This information is needed by the model to determine whether patients can be classified as SD or Overnight patients.

During that five-week period at HSC, surgical cases were classified into eleven services. The OR department consisted of 13 operating theatres, two of which were utilized by the same OR team. Therefore, the number of staffed operating theatres was set to 12. The actual opening and closing times for each operating theatre were used, along with the actual closing time for MS3 and the actual time at which B3 had to admit new Overnight patients. The maximum overtime allowed by any operating theatre was set to 2 hours because cases are rarely allowed to proceed if they will exceed that amount.

Because the majority of the required data was not stored electronically, collecting it was a large task and not all of the required information could be gathered. Estimates were made on the days that each surgeon and patient were available for surgery. In addition, the number of patients that could not be placed off-service was estimated by considering the patients' services. For example, most cardiac (Service A), neurosurgery (Service C), and plastics (Service I) patients cannot be placed off-service after surgery. Therefore, all patients under those three services were modelled as such. In the model, the daily number of elective surgical beds available for each service was assumed to be constant, based on 80% of the amount allocated to each service by the hospital during that period. This had to be done because the surgical beds available for each service at HSC are shared by both elective and emergency patients, and no distinction is made between the two.

The threshold value for a service's allowable number of off-service patients was set to 3, and the maximum allowable daily patient volume variability was set to 5, based on discussions with HSC management. Finally, reasonable estimates were made regarding the operating theatres that could accommodate each case, according to the case's service. For example, cardiac surgeries tend to require more operating theatre space and therefore all cardiac cases were modelled to have greater operating theatre restrictions. It was assumed that the OR time allocated to each surgeon during each week was equal to the total duration of cases that they had performed that week.

Because the equipment and instrumentation requirements for each case can be very extensive and are sometimes tailored for a specific case, they were not recorded. Instead, it was assumed that each service had a specific set of equipment required by each of their cases, which could not be shared with cases belonging to other services. The number of equipment sets available for each service, on each day, was assumed to be 10, which is a reasonable estimate for a typical service's equipment/instrumentation constraints.

4.5.2.1 Health Sciences Centre (HSC) Results

The five problems solved using the first stage of the mathematical model differ in the number of cases to be scheduled in the schedule week (n), along with the number of surgeons (p), as shown in Table 4-5.

Table 4-5 Parameters of each problem

	Week 1	Week 2	Week 3	Week 4	Week 5
n	126	148	101	135	131
p	38	40	41	41	39

Table 4-6 shows the values of *B1a* and *Blb* determined by the model, along with the required computational times.

Table 4-6 *B1a* and *B1b* values determined for each problem

	Week 1	Week 2	Week 3	Week 4	Week 5
B1a	6	3	5	6	5
CPU (sec)	33	46	7	18	8
Blb	80	103	67	79	54
CPU (sec)	6	4	4	6	6

Table 4-7 presents the computational results for the five problems solved. The term *Z* corresponds to the objective value found after the time given under *CPU*, and *B* gives the IP bound reported by the solver. If the solver had not found the global optimum after four hours, the search was terminated and the best solution obtained was recorded. The solver was able to find the global optimum for all priority levels in all problems, except for priority level three in Week 1, and priority levels two and three in Week 2. However, the solutions obtained were still better than the results achieved by the actual schedules used at HSC, as explained later in this section.

Table 4-7 Results for stage 1

Week	Priority Level	Objective Function	No. of Variables	No. of Constraints	CPU (sec)	Z	B
1	1	$2*b_{1a}/B1a + b_{1b}/B1b$	10137	6609	9	3	3
	2	b_2	10151	6624	381	24	24
	3	b_3	10396	6679	14400	3*	0
2	1	$2*b_{1a}/B1a + b_{1b}/B1b$	10643	7366	27	3.058	3.058
	2	b_2	10657	7381	14400	25*	23.4
	3	b_3	10912	7436	14400	2*	0
3	1	$2*b_{1a}/B1a + b_{1b}/B1b$	9628	5749	9	3.015	3.015
	2	b_2	9642	5764	455	17	17
	3	b_3	9902	5789	1517	0	0
4	1	$2*b_{1a}/B1a + b_{1b}/B1b$	10308	6946	8	3	3
	2	b_2	10322	6961	222	20	20
	3	b_3	10582	7016	271	0	0

5	1	$2*b_{1a}/B1a + b_{1b}/B1b$	10294	6787	10	3	3
	2	b_2	10308	6802	7759	25	25
	3	b_3	10558	6857	171	0	0

*Best, but not necessarily optimal, solution found

The first chart in Figure 4-1 compares the largest deviation between the number of inpatient beds occupied by any particular service on any two days of the scheduled week (i.e. b_{1a}), resulting from the actual schedules used at HSC and the model's schedules. On average, the model reduced these deviations by 25.1%. Similarly, the second chart compares the total number of inpatient beds occupied during the days following the scheduled week (i.e. b_{1b}), where the model's schedules reduced these numbers by an average of 19.9%.

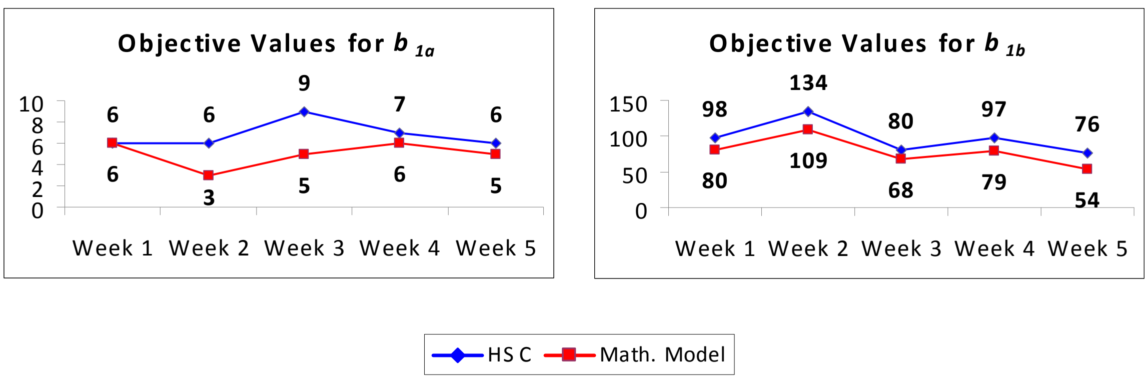

Figure 4-1 Comparison between b_{Ia} and b_{Ib} values obtained by HSC's schedules and the mathematical model's schedules

To smooth daily OR utilization, the model minimizes the largest deviation between available and utilized OR time on any day of the schedule (b_2). Table 4-8 depicts each day's available and utilized OR time during Week 1, according to the model's schedule, where the largest variability in OR utilization between any two days is only 1 half hour.

Table 4-8 Daily deviation between available and utilized operating room (OR) time in Week 1, obtained by the mathematical model's schedule

(half hours)	Mon.	Tue.	Wed.	Thu.	Fri.	Largest Variability
Available OR time	189	184	135	189	175	
Utilized OR time	165	160	112	165	151	
Deviation	24	24	23	24	24	**1**

Table 4-9 displays this variability for each week, according to the model's schedules. Daily OR utilization was balanced within 2 half hours (i.e. 1 hour) during each week.

Table 4-9 Largest operating room (OR) utilization variability between any two days during each week, obtained by the mathematical model's schedules

	Week 1	Week 2	Week 3	Week 4	Week 5
Largest Variability	1	2	1	0	2

At HSC, elective cases are only scheduled if there is enough available OR time for their completion. If a surgeon has unscheduled OR time close to the day of surgery, the time is given to other surgeons, or it is filled with emergency cases. Therefore, the theoretical deviations between available and utilized OR time in HSC's schedules are always very small, and there is no need to compare it with the mathematical model's schedules.

Figure 4-2 illustrates the largest deviation between the total number of SD and SDA patients scheduled on any two days during each week (i.e. b_3), due to the actual schedules and the model's schedules. The model's schedules reduced these deviations by an average of 94.3%, thereby reducing variable patient volumes and smoothing the workload and resource requirements in each department.

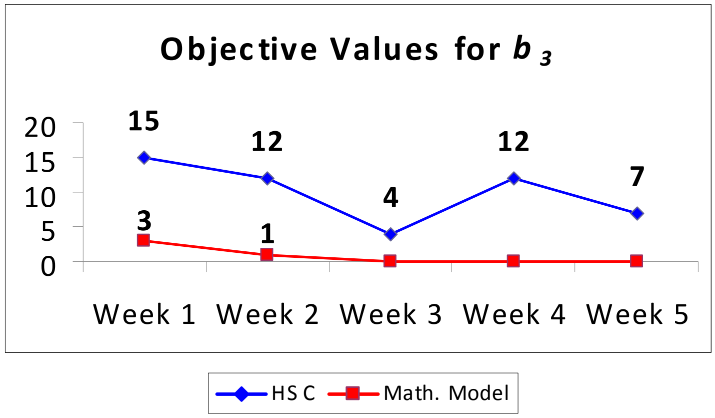

Figure 4-2 Comparison between b_3 values obtained by HSC's schedules and the mathematical model's schedules

The results for these HSC problems were used as the basis for an analysis of the relationships between the mathematical model's objectives in different priority levels. The analysis results can be found in Appendix B.

4.6 Summary

This chapter described the first stage of the proposed elective operating room (OR) scheduling system, based upon the elective surgical patient flow at Health Sciences Centre (HSC) in Winnipeg. This proposed system creates weekly OR schedules through an open scheduling policy. In this first stage, surgeons with allocated OR time in a particular week present the cases that they wish to schedule. Cases are then assigned to the different days of the week, while taking into account resultant inpatient bed occupancy volumes and resource constraints. The problem is comprised of four objectives classified into three priority levels, which strive to reduce day-to-day variability. The objectives in the first priority level minimize daily post-operative inpatient bed occupancy variability while maintaining high bed utilization. The second priority level minimizes the discrepancy between available and utilized daily OR time in order to smooth daily OR utilization. The third priority level smoothes total daily Same Day (SD) and Same Day Admission (SDA) elective patient volumes so that daily workload and resource requirements are balanced.

For this first stage, it was proposed that problems could be mathematically modelled and solved through lexicographic goal programming. The model was tested using data from five consecutive weeks of elective surgical cases at HSC. Compared to the actual schedules used, the model would have reduced the largest daily bed occupancy variability for any service by 25.1%, the inpatient bed occupancy on the days following the scheduled week by 19.9%, and total daily SD and SDA patient volume variability by 94.3%.

Alternatively, HSC can determine the case mix desired in a particular week, and use the model to provide a guideline regarding the number and types of cases that should be performed each day in order to reduce daily variability. HSC can then carry out daily scheduling in whatever manner they like.

The mathematical model can be easily tailored to meet the needs of other facilities by adjusting or adding more constraints, or changing the importance of objectives. Objectives may also be added, removed, or fine-tuned according to user preferences.

Chapter 5: Elective Operating Room Scheduling (OR) Scheduling System – Stage 2

This chapter describes two alternatives for conducting the second stage of the proposed elective operating room (OR) scheduling system. In the first stage, cases were assigned to days in the week to be scheduled. In the second stage, all of the cases scheduled on a particular day will be assigned operating theatres and start times. Hence, daily OR schedules will be created. Unlike the first stage, which aimed to reduce day-to-day variability, the purpose of the second stage is to reduce the variability that may occur within a day. The problem consists of four objectives that are classified into three priority levels.

The objectives for this stage are described in section 5.1, while the input data, assumptions, and constraints are explained in sections 5.2, 5.3, and 5.4, respectively. In section 5.5, the first alternative for scheduling is addressed. This method is a mathematical model, similar to the one used in the first stage, where optimal solutions are obtained through lexicographic goal programming. This method was chosen because the problem's objectives have very distinct priorities. In section 5.6, the second alternative for scheduling is introduced. This method is a genetic algorithm that is able to find good solutions, though not necessarily optimal, in a relatively short period of time. Finally, a summary is given in section 5.7.

5.1 Objectives

There are four objectives in this stage, classified into three priority levels. The first priority level is the most important, followed by the second level, and finally the third. These objectives, and their priority levels, were determined after assessing the factors affecting HSC's surgical patient flow. If desired, objectives can easily be excluded, or their corresponding priorities can be changed.

5.1.1 Priority Level One

The first priority level consists of the following two objectives:

1. Minimize the largest overtime experienced by any operating theatre
2. Minimize the largest deviation between OR time utilized by any two operating theatres

The first goal will minimize overtime in order to avoid unnecessary costs and staff dissatisfaction. The second goal will evenly distribute the hours of work among each staffed operating theatre, which is also important for staff satisfaction. These objectives were considered more important than the ones in other priority levels because costs in the OR department can be huge when overtime occurs, and staff satisfaction is crucial in maintaining an effective and motivated work environment.

5.1.2 Priority Level Two

The second priority level consists of the following objective:

- Minimize the largest number of Post-Anaesthesia Care Unit (PACU) beds occupied at any time

This objective corresponds to the bed occupancy in the PACU during different periods throughout the day. At HSC, elective cases are sometimes put on hold in the OR department due to a lack of PACU beds, which subsequently affects following cases. On the other hand, there are periods of the day when the PACU is quite empty. This shows that the demand for PACU beds can be highly variable within a particular day. By smoothing the unit's bed occupancy throughout the day, artificial variability will reduce and delays will decrease. This objective was considered more important than the one in priority level three because delays caused by PACU bed unavailability occur more frequently and have a bigger impact on surgical patient flow.

5.1.3 Priority Level Three

The objective in the third priority level is:

- Minimize the largest number of cases that finish at the same time

Multiple cases in the OR often finish at the same time, resulting in an inadequate number of peri-operative aides (PAs) available to turnover the operating theatres. When this occurs, the start of following elective cases scheduled in those operating theatres may be delayed. At other times, however, the same number of PAs may have nothing to do.

Hence, smoothing the number of cases that finish at the same time will help balance turnover workload throughout the day. In turn, cases will start on time, staff will be satisfied, and surgeons will be less tempted to leave the OR department during turnovers.

5.2 Input Data

For the second stage, almost all of the required input data would have already been collected in the first stage. Table 5-1 depicts the inputs needed for this stage, and is split up according to whether or not HSC already uses the data for elective OR scheduling:

Table 5-1 Input data required for stage 2

Source	Data already used by HSC	Data not currently used by HSC
Surgeons	• The cases to be scheduled, including: — Estimated case duration (including set-up and clean up)	• The cases to be scheduled, including: — Estimated recovery time in the recovery unit (i.e. PACU) — Estimated recovery time in the post-operative unit, if less than 24 hours
OR Dept.	• The operating theatres that can be used for each case • The opening and closing times for each staffed operating theatre, on each day	• The expected number of operating theatres that need to be cleaned at the beginning of the day
PACU		• The expected number of PACU beds that will be occupied at the beginning of the day
MS3	• The unit's daily closing time, where patients must be discharged	
B3	• The time when Overnight patients must be discharged in order to admit new patients	

Stage 2 requires far less data than stage 1 and uses only slightly more information than what is currently needed by HSC for elective OR scheduling. Recovery times can simply be estimated by the surgeons, or historical data can be used, if available. Management from the OR department and the PACU can easily specify the additional data required.

5.3 Assumptions

The assumptions used in stage 1 are also used in stage 2. All elective surgical patients go to the PACU for recovery after surgery. Afterwards, Same Day (SD) patients go to MS3, Overnight patients go to B3, and Inpatients and Same Day Admission (SDA) patients go to inpatient units. No other transfers take place before discharge. Many surgeons have offices nearby or at HSC, and they visit patients throughout the facility when they are not operating. Therefore, it is assumed that each surgeon is able to operate at any time on the days they stipulate they are available.

5.4 Constraints

In this stage, the following constraints need to be considered:

Table 5-2 List of constraints in stage 2

Constraint 1:	A case cannot be assigned to an operating theatre that it cannot be performed in
Constraint 2:	A case cannot be scheduled in an operating theatre before it is open
Constraint 3:	A surgeon cannot work on more than one case at a time
Constraint 4:	An operating theatre cannot have more than one case scheduled in it at a time
Constraint 5:	A SD patient must be scheduled early enough to allow for sufficient recovery and discharge by the time MS3 closes
Constraint 6:	An Overnight patient must be scheduled early enough to allow for sufficient recovery and discharge by the time B3 has to admit new patients

5.5 Mathematical Model for Stage 2

For this stage, two ways of generating daily schedules are proposed. First, the problem can be mathematically modelled and solved using lexicographic programming.

The following notations are employed in the mathematical model:

MS3 Closing time of MS3

B3 Time at which new Overnight patients must be admitted in B3

Dur_i Surgical duration of case *i*, in half hours

where $i = 1,\ldots,n$ (n = number of cases to be scheduled)

Rec_i Recovery duration of case *i*, in half hours

$Post_i$ Post-operative duration of case *i*, in half hours, if less than 24 hours; 50 otherwise

Tot_i Total duration of case *i*, including surgery, recovery, and post-operative care ($Dur_i + Rec_i + Post_i$)

Sur_{is} 1 if case *i* is under surgeon *s*; 0 otherwise

where $s = 1,\ldots,p$ (p = number of surgeons with cases)

$Open_o$ Opening time for operating theatre *o*

where $o = 1,\ldots,r$ (r = number of staffed operating theatres)

$Close_o$ Closing time for operating theatre *o*

D_o Total OR time available for operating theatre *o* ($Close_o - Open_o$)

OR_{io} 1 if case *i* can be performed in operating theatre *o*; 0 otherwise

x_i 1 if the patient of case *i* is scheduled as a SD patient; 0 otherwise

y_i 1 if the patient of case *i* is scheduled as a Overnight patient; 0 otherwise

z_i 1 if the patient of case i is scheduled as a SDA patient; 0 otherwise

M A large number

$PACU_1$ Number of PACU beds occupied at the beginning of the day (Time 1)

PA_1 Number of operating theatres that need to be cleaned at the beginning of the day (Time 1)

The following variables represent the mathematical model's solution:

b_{1a} Largest overtime experienced by any operating theatre

b_{1b} Largest deviation between OR time utilized by any two operating theatres

b_2 Largest number of PACU beds occupied at any time

b_3 Largest number of cases that finish at the same time

R_{io} 1 if operation i is assigned to operating theatre o; 0 otherwise

$Start_i$ Scheduled start time of operation i, beginning with set-up

Fin_o Scheduled finish time of operating theatre o

$Over_o$ Overtime for operating theatre o, in half hours

$Under_o$ Underutilized OR time for operating theatre o, in half hours

ξ_o Difference between available and utilized OR time for operating theatre o, in half hours

α_{ic} 1 if the recovery period of case i is scheduled to start at the beginning of period c; 0 otherwise

where $c = 1,\ldots,$ a (a = number of half hour periods during the day)

β_{ic} 1 if the recovery period of case i is scheduled to finish at the beginning of period c; 0 otherwise

PACU_c Number of PACU beds required during period c

PA_c Number of cases that finish at the beginning of period c

The following variables ensure a linear mathematical model:

θ_{id} 1 if case i starts before case d; 0 otherwise

where i, d = 1,…,n (n=number of cases to be scheduled)

b_{1b}^{-} Negative deviation between OR time utilized by operating theatres o and e

b_{1b}^{+} Positive deviation between OR time utilized by operating theatres o and e

Objective Function:

Priority Level 1: Minimize $z = 2 * b_{1a} + b_{1b}$ (1)

Priority Level 2: Minimize $z = b_2$ (2)

Priority Level 3: Minimize $z = b_3$ (3)

Subject to:

$$R_{io} \le OR_{io} \qquad \forall i \in [1..n], \forall o \in [1..r] \tag{4}$$

$$Start_i \ge Open_o - M(1 - R_{io}) \qquad \forall i \in [1..n], \forall o \in [1..r] \tag{5}$$

$$Start_d \ge (Start_i + Dur_i) - M(3 - \theta_{id} - Sur_{is} - Sur_{ds})$$

$$\forall i, d \in [1..n] / i \ne d, \forall s \in [1..p] \tag{6}$$

$$Start_d \geq (Start_i + Dur_i) - M(3 - \theta_{id} - R_{io} - R_{do})$$

$$\forall i, d \in [1..n] / i \neq d, \forall o \in [1..r] \quad (7)$$

$$x_i(Start_i + Total_i) \leq MS3 \qquad \forall i \in [1..n] \quad (8)$$

$$y_i(Start_i + Total_i) \leq B3 \qquad \forall i \in [1..n] \quad (9)$$

$$Start_d \geq Start_i - M(1 - \theta_{id}) \qquad \forall i, d \in [1..n] / i \neq d \quad (10)$$

$$Fin_o \geq (Start_i + Dur_i) - M(1 - R_{io}) \qquad \forall i \in [1..n], \forall o \in [1..r] \quad (11)$$

$$Over_o - Under_o \geq Fin_0 - Close_o \qquad \forall o \in [1..r] \quad (12)$$

$$b_{1a} \geq Over_o \qquad \forall o \in [1..r] \quad (13)$$

$$\xi_o = Close_o - Open_o - \sum_{i=1}^{n} Dur_i R_{io} \qquad \forall o \in [1..r] \quad (14)$$

$$b_{1b}^{+} - b_{1b}^{-} \geq \xi_o - \xi_e \qquad \forall o, e \in [1..r] \quad (15)$$

$$b_{1b} \geq b_{1b}^{+} + b_{1b}^{-} \quad (16)$$

$$Start_i + Dur_i = \sum_{c=1}^{a} c(\alpha_{ic}) \qquad \forall i \in [1..n] \tag{17}$$

$$Start_i + Dur_i + \mathrm{Re}\,c_i = \sum_{c=1}^{a} c(\beta_{ic}) \qquad \forall i \in [1..n] \tag{18}$$

$$PACU_c = PACU_{(c-1)} + \sum_{i=1}^{n} (\alpha_{ic} - \beta_{ic}) \qquad \forall c \in [1..a] \tag{19}$$

$$b_2 \geq PACU_c \qquad \forall c \in [1..a] \tag{20}$$

$$PA_c = \sum_{i=1}^{n} \alpha_{ic} \qquad \forall c \in [1..a] \tag{21}$$

$$b_3 \geq PA_c \qquad \forall c \in [1..a] \tag{22}$$

$$\theta_{id} + \theta_{di} = 1 \qquad \forall i,d \in [1..n] / i > d \tag{23}$$

$$\sum_{o=1}^{r} R_{io} = 1 \qquad \forall i \in [1..n] \tag{24}$$

$$\sum_{c=1}^{a} \alpha_{ic} = 1 \qquad \forall i \in [1..n] \tag{25}$$

$$\sum_{c=1}^{a} \beta_{ic} = 1 \qquad \forall i \in [1..n] \qquad (26)$$

$$R_{io}, \alpha_{ic}, \beta_{ic} \in \{0,1\} \qquad \begin{matrix} \forall i \in [1..n], \forall o \in [1..r], \\ \forall s \in [1..p], \forall c \in [1..a] \end{matrix} \qquad (27)$$

The goals in the mathematical model are split into 3 priority levels. In order to generate a schedule for stage 2 using lexicographic goal programming, the model will need to be executed 3 times. Equations (1-3) represent the objective functions for priority levels 1-3, respectively. In equation (1), b_{1a} and b_{1b} represent the two priority level one objectives. Based on how delays and elective surgical case cancellations can occur at HSC, b_{1a} was given a weight of 2 because it was considered twice as important as b_{1b}. Because both terms are in the same units and their ideal value is zero, they do not need to be normalized before b_{1a}'s weight is applied. The objectives in priority level two and three are represented by the terms b_2 and b_3 in equations (2) and (3), respectively.

Equations (4–9) correspond to Constraints 1–6 (described in Table 5-2), respectively. Equation (10) determines whether one case is scheduled before another. Equation (11) calculates the scheduled finish time of each operating theatre. Equation (12) works out the overtime or underutilized OR time for each theatre, which is then used in equation (13) to compute the value of the first objective in priority level one. Equation (14) analyzes the amount of OR time utilized in each operating theatre, which is used by equations (15–16) to establish the value of the second objective in priority level one.

Equations (17–18) calculate when the recovery period in the PACU will start and end for each case, respectively. Equation (19) determines the number of PACU beds occupied during each half hour period, which is subsequently used in equation (20) to work out the objective value in priority level two. Similarly, the number of cases that finish in the OR at the start of each period is calculated in equation (21), and those values are used in equation (22) to determine the objective value in priority level three. Finally, equations (23–27) are the integrality constraints.

When the model is run for priority level one, equations (17–22) are not included in the model because they have no effect on the solution obtained, and will only needlessly increase the number of variables and constraints that are considered. Similarly, equations (21–22) are not required when the model is run for priority level two, although the objective values previously found for priority level one must be included as constraints. When the model is run for priority level three, all equations must be included and the objective values found for priority levels one and two must be formulated as constraints.

5.5.1 Execution of the Mathematical Model for Stage 2

The first two sets of the test problems previously created and solved in stage 1 (described in section 4.5.1) were also used to test the stage 2 mathematical model. The mathematical model for stage 2 was executed using Lingo[6]. Microsoft Office Excel[7] tables were used to arrange input data that the Lingo solver could read from, in addition to providing a place for solution values to be stored.

---

[6] Copyright © LINDO Systems Inc
[7] Copyright © Microsoft Corporation

Table 5-3 on the following page depicts the computational results for the two sets of problems. The term *Z* corresponds to the objective value found after the time given under *CPU*, and *B* gives the IP bound reported by the solver. If the global optimum had not been found after four hours, the search was terminated and the best solution obtained was recorded. Solutions were manually validated to confirm that all constraints and assumptions were satisfied.

For the first set of test problems, the global optimum for all three priority levels could only be found for one particular day (i.e. Thursday) within a reasonable time frame. For the second set of test problems, the global optimum could not be found for any priority level after a reasonable amount of time. Furthermore, feasible solutions could only be found for priority level one in four out of the five problems. These results made it obvious that the mathematical model is only able to handle problems of very small sizes, and therefore cannot be applied to real-life problems at HSC.

These results prompted the decision to pursue another multi-objective optimization method to solve the stage 2 daily OR scheduling problem. The method needed to be able to handle large-sized problems, while generating good solutions in reasonable amounts of time. The Nondominated Sorting Genetic Algorithm II for Operating Room Scheduling (NSGAII-OR) was created for this purpose and is described in the following sections.

Table 5-3 Test results for stage 2, using the mathematical model

Test	Day	Priority Level	Objective Function	No. of Variables	No. of Constraints	CPU (sec)	Z	B
1	Mon	1	$2^*b_{1a} + b_{1b}$	254	959	14400	10*	2
		2	b_2	1452	1102	14400	5*	1.261
		3	b_3	1500	1198	14400	3*	0.294
	Tue	1	$2^*b_{1a} + b_{1b}$	176	573	99	4	4
		2	b_2	1086	704	14400	4*	1.433
		3	b_3	1134	800	14400	2*	0.319
	Wed	1	$2^*b_{1a} + b_{1b}$	200	691	1780	8	8
		2	b_2	1206	826	14400	4*	1.289
		3	b_3	1254	922	14400	2*	1
	Thu	1	$2^*b_{1a} + b_{1b}$	176	579	134	6	6
		2	b_2	1086	710	8525	3	3
		3	b_3	1134	806	8694	2	2
	Fri	1	$2^*b_{1a} + b_{1b}$	200	695	938	8	8
		2	b_2	1206	830	14400	4*	0.835
		3	b_3	1254	926	1979	1	1
2	Mon	1	$2^*b_{1a} + b_{1b}$	466	2220	14400	14*	0
		2	b_2	2048	2379	14400	N/A	1.761
		3	b_3	-	-	-	-	-
	Tue	1	$2^*b_{1a} + b_{1b}$	686	3682	14400	41*	0
		2	b_2	2748	3861	14400	N/A	3.196
		3	b_3	-	-	-	-	-
	Wed	1	$2^*b_{1a} + b_{1b}$	686	3676	14400	20*	0
		2	b_2	2748	3855	14400	N/A	2.087
		3	b_3	-	-	-	-	-
	Thu	1	$2^*b_{1a} + b_{1b}$	686	3680	14400	N/A	0
		2	b_2	-	-	-	-	-
		3	b_3	-	-	-	-	-
	Fri	1	$2^*b_{1a} + b_{1b}$	686	3687	14400	17*	0
		2	b_2	2748	3866	14400	N/A	1.783
		3	b_3	-	-	-	-	-

* Best, but not necessarily optimal, solution found

5.6 Nondominated Sorting Genetic Algorithm II for Operating Room Scheduling (NSGAII-OR)

The Nondominated Sorting Genetic Algorithm II for Operating Room Scheduling (NSGAII-OR) is a version of the Nondominated Sorting Genetic Algorithm II (NSGA-II) (Deb et al. 2000a, Deb et al. 2000b), specifically adapted for the second stage of the proposed elective OR scheduling system.

5.6.1 Modification of Objectives

The objectives declared in section 5.1 are the same ones addressed by NSGAII-OR. However, their priority levels are not considered in order to allow users to see the real trade-offs for each objective in each non-dominated solution. The table lists the objectives in NSGAII-OR.

Table 5-4 List of objectives in NSGA-II

Objective 1:	Minimize the largest overtime experienced by any operating theatre (f_1)
Objective 2:	Minimize the largest deviation between OR time utilized by any two operating theatres (f_2)
Objective 3:	Minimize the largest number of Post-Anaesthesia Care Unit (PACU) beds occupied at any time (f_3)
Objective 4:	Minimize the largest number of cases that finish at the same time (f_4)

$f1$, f_2, f_3, and f_4 are equal to b_{1a}, b_{1b}, b_2, and b_3 in the mathematical model, respectively.

5.6.2 Domination Definition

In NSGAII-OR, a solution x_i dominates a solution x_j if:

1. Solution x_i is feasible and solution x_j is not
2. Solutions x_i and x_j are both infeasible, but solution x_i has a smaller overall constraint violation
3. Solutions x_i and x_j are both infeasible and have the same overall constraint violation, but solution x_i dominates solution x_j with regards to the objectives
4. Solutions x_i and x_j are both feasible, but solution x_i dominates solution x_j with regards to the objectives

This definition is almost the same as the one proposed by Deb (2001), except for the addition of rule (3). This rule was included because a few of the problems addressed in this chapter have no feasible solutions, and it is possible for the algorithm's population to solely consist of solutions having the same overall constraint violation. Therefore, the infeasible solutions with better objective values are preferable, and rule (3) considers this.

5.6.3 Non-dominated Sorting

According to NSGAII-OR's domination definition, the entire algorithm's population must be sorted according to non-domination each time a new generation begins. To achieve this, the following items must be determined for each solution (Deb et al. 2000a, 2000b):

1. Domination count n_p: the number of solutions that dominate solution p
2. S_p: a set of solutions that solution p dominates

If a solution p has $n_p = 0$, it is non-dominated. For each non-dominated solution, every member in its set S_p is given a new $n_p = n_p - 1$. If this results in $n_p = 0$ for any solution q, they are put in a separate list Q. These members belong to the second non-dominated front. The procedure repeats for every member in Q in order to identify the third non-dominated front. The process continues until the front for each solution is determined.

5.6.4 Crowding Distance

In NSGAII-OR, a crowded comparison operator is used during selection and replacement to preserve diversity. First, a *crowding distance* metric is obtained for each solution. This metric estimates the density of solutions surrounding a particular solution. A solution's crowding distance d_i is equal to the average distance of the two solutions on either side of it, along each objective. For a given non-dominated set, crowding distance is calculated in the following way:

1. For each objective function m:
   a. Sort the set according to lowest order of objective value f_m
   b. If each solution's f_m is a boundary value (i.e. lowest or highest): $d_{i_m} = 0$

      Otherwise:

      $d_{i_m} = \infty$ for each solution with an f_m that is a boundary value.

      For all other solutions:

$$d_{i_m} = \frac{f_m^{(i+1)} - f_m^{(i-1)}}{f_m^{\max} - f_m^{\min}}$$

2. For each solution i, $d_i = \sum_1^m d_{i_m}$

f_m^{i-1} and f_m^{i+1} represent the solutions before and after solution i in the sorted list. $f_m^{\max}$ and $f_m^{\min}$ are the maximum and minimum values of objective function m in the population.

For example, assume that a two-objective problem (f_1, f_2) has 4 solutions (A, B, C, D) in its non-dominated set, and their objective values are: A = (6, 3); B = (8, 6); C = (4, 4); D = (12, 2). Here, $f_1^{\max}$ and $f_1^{\min}$ are 12 and 4, while $f_2^{\max}$ and $f_2^{\min}$ are 6 and 2. At step 1, we start with objective function 1 and sort the set according to lowest order of f_1. Therefore, the set is sorted as follows: C = (**4**, 4); A = (**6**, 3); B = (**8**, 6); D = (**12**, 2). For objective function 1, C and D are boundary solutions because their f_1 values are the lowest and highest in the list. If A and B were also boundary solutions (i.e. their f_1 values were also equal to 4 or 12), step 1a would occur. Since this is not the case, step 1b is carried out and d_{i_1} is calculated for each solution i. Step 1 is then repeated for objective function 2, where d_{i_2} is now calculated. This time, D and B are boundary solutions. Finally, each solution's total crowding distance value d_i is calculated by summing its individual crowding distance values d_{i_1} and d_{i_2}. Table 5-5 details these calculations.

Table 5-5 Example of crowding distance calculations

i	f_1	f_1^{i+1}	f_1^{i-1}	$d_{i_1} = \frac{f_m^{(i+1)} - f_m^{(i-1)}}{f_m^{\max} - f_m^{\min}}$	f_2	f_2^{i+1}	f_2^{i-1}	$d_{i_1} = \frac{f_m^{(i+1)} - f_m^{(i-1)}}{f_m^{\max} - f_m^{\min}}$	$d_i = d_{i_1} + d_{i_2}$
A	6	8	4	(8-4)/(12-4)=0.5	3	4	2	(4-2)/(6-2)=0.5	0.5+0.5=1
B	8	12	6	(12-6)/(12-4)=0.75	6	-	4	∞	0.75+∞=∞
C	4	6	-	∞	4	6	3	(6-3)/(6-2)=0.75	∞+0.75=∞
D	12	-	8	∞	2	3	-	∞	∞+∞=∞

This operator is almost the same as the one used in NSGA-II, except that step 1a has been included. It was found that in many problems addressed in this chapter, the values obtained for Objective 4 in a non-dominated front were often only boundary solutions. In NSGA-II's original crowding distance calculation, each of the solutions in that front would have been assigned crowding distances equal to infinity, regardless of their values for the other three objectives. Step 1a allows us to avoid this.

5.6.5 Solution Representation

Each solution in the population is represented by a two-dimensional matrix, where columns correspond to half-hour periods and rows correspond to operating theatres. Therefore, each cell depicts the case scheduled in a particular operating theatre at a given time and naturally ensures that Constraint 4 is satisfied. Figure 5-1 is an example of a simple OR schedule where eight cases have been scheduled over three operating theatres. Figure 5-2 depicts how this schedule would be represented in NSGAII-OR.

<table>
<tr><th></th><th>Time 1</th><th>Time 2</th><th>Time 3</th><th>Time 4</th><th>Time 5</th><th>Time 6</th><th>Time 7</th><th>Time 8</th><th>Time 9</th></tr>
<tr><th>OR 1</th><td colspan="3">Case 1</td><td colspan="5">Case 5</td><td></td></tr>
<tr><th>OR 2</th><td>Case 2</td><td colspan="4">Case 7</td><td colspan="4">Case 4</td></tr>
<tr><th>OR 3</th><td colspan="2">Case 3</td><td></td><td colspan="2">Case 8</td><td></td><td colspan="3">Case 6</td></tr>
</table>

Figure 5-1 Example of an OR Schedule

1	1	1	5	5	5	5	5	
2	7	7	7	7	4	4	4	4
3	3		8	8		6	6	6

Figure 5-2 Example of an NSGAII-OR solution

5.6.6 Main Loop

NSGAII-OR consists of the following steps:

1. *Initialization*: Generate a population of N solutions
2. *Selection*: From the population, create a mating pool of N solutions
3. *Crossover*: From the mating pool, produce a population of N offspring using a crossover probability p_c
4. *Mutation*: Mutate offspring solutions using a mutation probability p_m
5. *Replacement*: Form a new population of N solutions
6. Repeat steps (2–5) until the required number of generations has been reached
7. Present the solutions belonging to the non-dominated front in the final population

5.6.7 Initialization

During the initialization process, a population of solutions is generated. In most genetic algorithms, this is carried out randomly. Datta et al. (2007) found that this may make it difficult to find feasible solutions. Hence, the authors used an heuristic to assist the algorithm's search for feasible solutions. Because of the constraints in the elective OR daily scheduling problem, NSGAII-OR also employs an heuristic to generate the initial population. The heuristic utilizes the following steps:

1. For each SD case i, calculate $t_i = MS3 - Tot_i$
2. For each Overnight case i, calculate $t_i = B3 - Tot_i$
3. Schedule each SD and Overnight case according to ascending t, at the earliest possible time where Constraints 1 – 4 are not violated

4. Schedule each remaining case at the earliest possible time in a randomly chosen operating theatre where Constraints 1 – 4 are not violated

The terms *MS3* and *B3* correspond to the time when the post-operative units MS3 and B3 have to discharge SD and Overnight patients, respectively. The term Tot_i is equal to the total duration of case *i*, including surgery, recovery, and post-operative care. Therefore, the term t_i corresponds to how early an SD or Overnight case *i* must be scheduled so that Constraints 5 and 6 are not violated. For example, if an SD patient's total duration (Tot_i) is equal to 11 hours and MS3's closing time is 12 hours after the start of the schedule, then the patient's operation must be scheduled within the first hour or else the patient cannot be discharged by the time MS3 closes. Hence, the heuristic ensures that the SD and Overnight cases with the smallest scheduling windows are scheduled first.

This heuristic ensures that every solution respects Constraints 1 – 4. However, certain problems may have no feasible solutions with regards to Constraints 5 and 6. In those cases, the heuristic will only minimize the violation of Constraints 5 and 6.

Once a population has been created, the objectives values and constraint violations must be determined for each solution. The population can then be sorted according to non-domination, and each solution's crowding distance can be calculated.

5.6.8 Selection

The mating pool is formed through a tournament selection process, where each solution in the population is compared to another solution. Therefore, each solution should take part in a comparison exactly twice. For each comparison, the solution on a lower non-domination level is chosen to join the mating pool. If two solutions are on the same level, then the solution with the larger crowding distance is selected.

5.6.9 Crossover

During crossover, two parent solutions are randomly chosen from the mating pool and used to produce two offspring solutions, using the following steps:

1. The first and second offspring copy a random number of columns from the first and second parent, respectively.
2. The first offspring copies a random number of columns from the second parent, provided that Constraints 1 – 4 are not violated. Similarly, the second offspring copies a random number of columns from the first parent.
3. Each unscheduled SD and Overnight case is scheduled according to ascending *t*, at the earliest possible time where Constraints 1 – 4 are not violated.
4. Each remaining unscheduled case is scheduled at the earliest possible time in the operating theatre with the most underutilized OR time, where Constraints 1 – 4 are not violated.

For example, Figure 5-3 depicts how crossover would occur at step 1, if the second column (i.e. Time2) is randomly chosen from Parent 1 and the third and fourth columns (i.e. Time3 and Time4) are randomly chosen from Parent 2.

<table>
<tr><th colspan="5">Parent 1</th></tr>
<tr><th></th><th>Time1</th><th>Time2</th><th>Time3</th><th>Time4</th></tr>
<tr><td>OR 1</td><td>Case 1</td><td>Case 5</td><td colspan="2">Case 4</td></tr>
<tr><td>OR 2</td><td colspan="3">Case 3</td><td></td></tr>
<tr><td>OR 3</td><td colspan="2">Case 2</td><td></td><td></td></tr>
</table>

<table>
<tr><th colspan="5">Parent 2</th></tr>
<tr><th></th><th>Time1</th><th>Time2</th><th>Time3</th><th>Time4</th></tr>
<tr><td>OR 1</td><td></td><td></td><td>Case 5</td><td>Case 1</td></tr>
<tr><td>OR 2</td><td colspan="2">Case 2</td><td colspan="2">Case 4</td></tr>
<tr><td>OR 3</td><td colspan="3">Case 3</td><td></td></tr>
</table>

<table>
<tr><th colspan="5">Offspring 1</th></tr>
<tr><th></th><th>Time1</th><th>Time2</th><th>Time3</th><th>Time4</th></tr>
<tr><td>OR 1</td><td></td><td>Case 5</td><td></td><td></td></tr>
<tr><td>OR 2</td><td colspan="3">Case 3</td><td></td></tr>
<tr><td>OR 3</td><td colspan="2">Case 2</td><td></td><td></td></tr>
</table>

<table>
<tr><th colspan="5">Offspring 2</th></tr>
<tr><th></th><th>Time1</th><th>Time2</th><th>Time3</th><th>Time4</th></tr>
<tr><td>OR 1</td><td></td><td></td><td>Case 5</td><td>Case 1</td></tr>
<tr><td>OR 2</td><td></td><td></td><td colspan="2">Case 4</td></tr>
<tr><td>OR 3</td><td colspan="3">Case 3</td><td></td></tr>
</table>

Figure 5-3 Example of crossover at step 1

Figure 5-4 demonstrates how crossover would occur at step 2, if the first and third columns (i.e. Time1 and Time3) are randomly chosen from Parent 1 and the fourth column (i.e. Time4) is randomly chosen from Parent 2. Offspring 2 is only able to copy the cell position of Case 1 from Parent 1 because Cases 3 and 4 have already been scheduled, while the addition of Case 2 would violate Constraint 4. Similarly, Offspring 1 is only able to copy the cell position of Case 1 from Parent 2 because the addition of Case 4 would violate Constraint 4.

<table>
<tr><th colspan="5">Parent 1</th></tr>
<tr><th></th><th>Time1</th><th>Time2</th><th>Time3</th><th>Time4</th></tr>
<tr><th>OR 1</th><td>Case 1</td><td>Case 5</td><td colspan="2">Case 4</td></tr>
<tr><th>OR 2</th><td colspan="3">Case 3</td><td></td></tr>
<tr><th>OR 3</th><td colspan="2">Case 2</td><td></td><td></td></tr>
</table>

<table>
<tr><th colspan="5">Parent 2</th></tr>
<tr><th></th><th>Time1</th><th>Time2</th><th>Time3</th><th>Time4</th></tr>
<tr><th>OR 1</th><td></td><td></td><td>Case 5</td><td>Case 1</td></tr>
<tr><th>OR 2</th><td colspan="2">Case 2</td><td colspan="2">Case 4</td></tr>
<tr><th>OR 3</th><td colspan="3">Case 3</td><td></td></tr>
</table>

<table>
<tr><th colspan="5">Offspring 1</th></tr>
<tr><th></th><th>Time1</th><th>Time2</th><th>Time3</th><th>Time4</th></tr>
<tr><th>OR 1</th><td></td><td>Case 5</td><td></td><td>Case 1</td></tr>
<tr><th>OR 2</th><td colspan="3">Case 3</td><td></td></tr>
<tr><th>OR 3</th><td colspan="2">Case 2</td><td></td><td></td></tr>
</table>

<table>
<tr><th colspan="5">Offspring 2</th></tr>
<tr><th></th><th>Time1</th><th>Time2</th><th>Time3</th><th>Time4</th></tr>
<tr><th>OR 1</th><td>Case 1</td><td></td><td>Case 5</td><td></td></tr>
<tr><th>OR 2</th><td></td><td></td><td colspan="2">Case 4</td></tr>
<tr><th>OR 3</th><td colspan="3">Case 3</td><td></td></tr>
</table>

Figure 5-4 Example of crossover at step 2

Finally, Figure 5-5 shows how crossover would occur at steps 3 and 4, where all remaining unscheduled cases are added to each offspring.

<table>
<tr><th colspan="5">Offspring 1</th></tr>
<tr><th></th><th>Time1</th><th>Time2</th><th>Time3</th><th>Time4</th></tr>
<tr><th>OR 1</th><td></td><td>Case 5</td><td></td><td>Case 1</td></tr>
<tr><th>OR 2</th><td colspan="3">Case 3</td><td></td></tr>
<tr><th>OR 3</th><td colspan="2">Case 2</td><td colspan="2">Case 4</td></tr>
</table>

<table>
<tr><th colspan="5">Offspring 2</th></tr>
<tr><th></th><th>Time1</th><th>Time2</th><th>Time3</th><th>Time4</th></tr>
<tr><th>OR 1</th><td>Case 1</td><td></td><td>Case 5</td><td></td></tr>
<tr><th>OR 2</th><td colspan="2">Case 2</td><td colspan="2">Case 4</td></tr>
<tr><th>OR 3</th><td colspan="3">Case 3</td><td></td></tr>
</table>

Figure 5-5 Example of crossover at steps 3 and 4

This operator ensures that Constraints 1 – 4 are not violated. However, due to the characteristics of the problem at hand, there is no guarantee that Constraints 5 and 6 will not be violated although the operator does reduce this possibility.

5.6.10 Mutation

During mutation, offspring are mutated according to the following steps:

1. Randomly select one case *x*.
2. If there is another case that can switch cell positions with case *x* (i.e. start time and OR), without violating any constraints (i.e. Constraints 1 – 6), then:

   Complete step 3.

   Otherwise:

   Go to step 4.
3. Randomly select a second case *y* that can switch cell positions with case *x*, without violating any constraints. Then schedule case *x* to begin at the original starting cell position of case *y*, and vice versa.
4. If there is another case that can switch cell positions with case x without violating Constraints 1 2, 5, and 6, then:

   Go to step 6.

   Otherwise:

   Complete step 5.
5. Reschedule case *x* in a randomly chosen OR at the earliest time where no constraints are violated.
6. Randomly select a second case *y* that can switch cell positions with case *x*, without violating Constraints 1, 2, 5, and 6. Remove cases *x* and *y* from the offspring.
7. If both cases are scheduled in the same OR, then:

   Complete step 8.

Otherwise:

Complete step 9.

8. From cases *x* and *y*, schedule the case that originally had the later start time in the same OR at the earliest time where no constraints are violated. Then schedule the other case in the same OR at the earliest time where no constraints are violated.
9. Schedule case *x* in the original OR of case *y* at the earliest time where no constraints are violated. Then schedule case *y* in the original OR of case *x* at the earliest time where no constraints are violated.

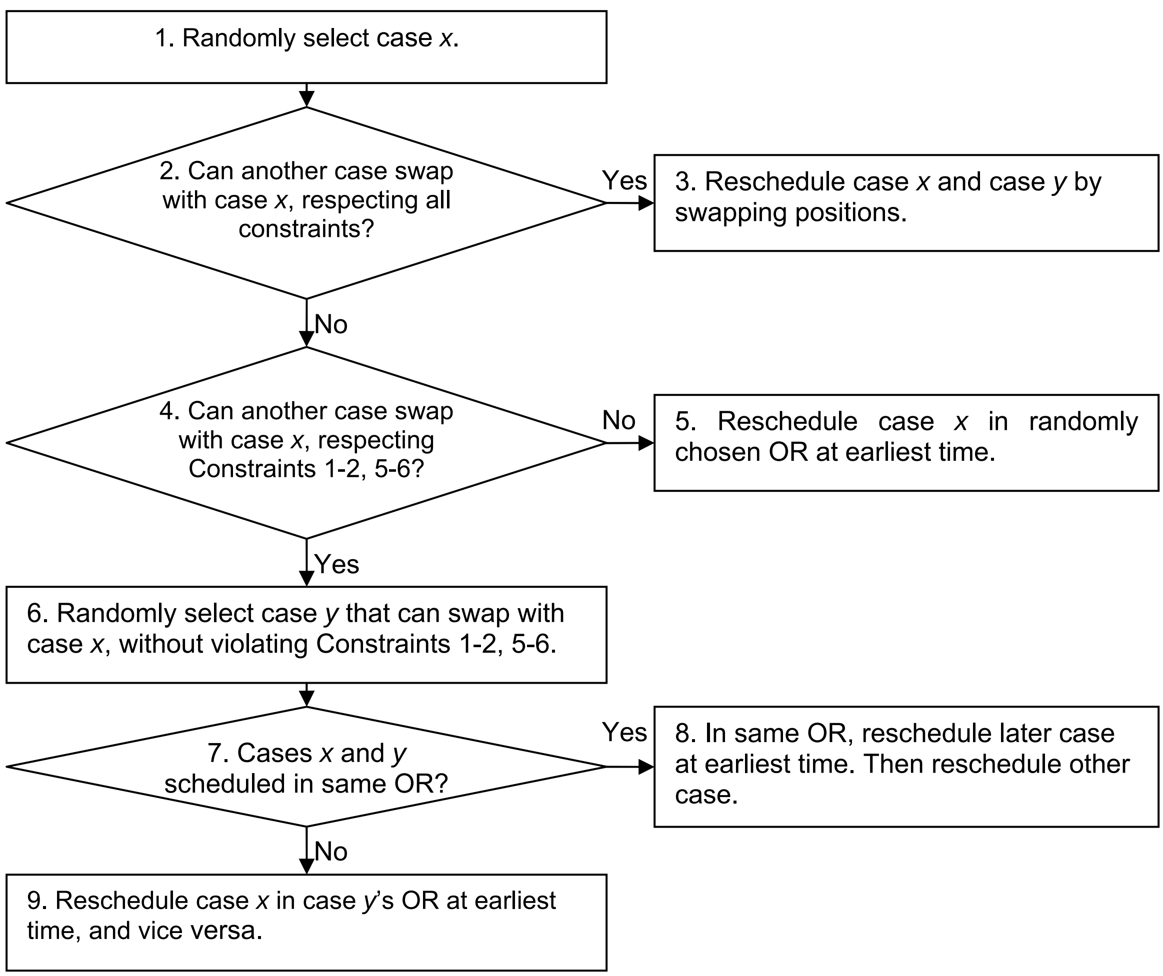

Figure 5-6 Flowchart of NSGAII-OR's mutation operator

This mutation operator will result in one of four scenarios (i.e. step 3, 5, 8, or 9) and will not degrade a solution's overall constraint violation. To depict how these 4 scenarios can occur, the offspring shown in Figure 5-7 will be used in several examples, and Table 5-5 gives information about each case.

<table>
<tr><th colspan="5">Offspring before Mutation</th></tr>
<tr><th></th><th>Time1</th><th>Time2</th><th>Time3</th><th>Time4</th></tr>
<tr><th>OR 1</th><td>Case 5</td><td></td><td>Case 6</td><td>Case 1</td></tr>
<tr><th>OR 2</th><td colspan="3">Case 3</td><td></td></tr>
<tr><th>OR 3</th><td colspan="2">Case 2</td><td colspan="2">Case 4</td></tr>
</table>

Figure 5-7 Example of an offspring before mutation

Table 5-6 Case information for the example offspring

Case	Surgeon	Possible ORs	Patient Type	Latest time case can be scheduled without violating Constraint 5	Latest time case can be scheduled without violating Constraint 6
1	A	All	SDA	N/A	N/A
2	B	2, 3	SD	Time3	N/A
3	A	All	SD	Time3	N/A
4	C	All	SD	Time3	N/A
5	C	All	Overnight	N/A	Time3
6	B	All	SD	Time3	N/A

If Case 5 is chosen as case *x* during mutation step 1, the mutation operator will check to see if there are any cases that Case 5 can switch starting cell positions with, without violating any constraints (step 2). Table 5-6 lists the outcomes of switching each case with Case 5. Since Case 4 is the only case that can switch with Case 5, without violating any constraints, mutation step 3 is carried out using Cases 4 and 5 as demonstrated in Figure 5-8.

Table 5-7 Outcome of switching Case 5 with other cases in the example offspring

Case	Constraint Violation Outcome	Constraint Violation No.
1	Case 5 would start at Time4.	6
2	Case 2 would be performed in OR 1.	1
3	Both Cases 3 and 6 would be scheduled in OR 2 at Time 3	4
4	N/A	N/A
6	Both Cases 2 and 6 would be scheduled at Time1. Both Cases 4 and 5 would be scheduled at Time3.	3 3

<table>
<tr><th colspan="5">Offspring before Mutation</th></tr>
<tr><th></th><th>Time1</th><th>Time2</th><th>Time3</th><th>Time4</th></tr>
<tr><th>OR 1</th><td>Case 5</td><td></td><td>Case 6</td><td>Case 1</td></tr>
<tr><th>OR 2</th><td colspan="3">Case 3</td><td></td></tr>
<tr><th>OR 3</th><td colspan="2">Case 2</td><td colspan="2">Case 4</td></tr>
</table>

<table>
<tr><th colspan="5">Offspring after Mutation</th></tr>
<tr><th></th><th>Time1</th><th>Time2</th><th>Time3</th><th>Time4</th></tr>
<tr><th>OR 1</th><td colspan="2">Case 4</td><td>Case 6</td><td>Case 1</td></tr>
<tr><th>OR 2</th><td colspan="3">Case 3</td><td></td></tr>
<tr><th>OR 3</th><td colspan="2">Case 2</td><td>Case 5</td><td></td></tr>
</table>

Figure 5-8 Example of mutation after step 3

Similarly, if Case 1 is chosen as case *x*, Table 5-7 lists the outcomes of switching each case with Case 1. Since there are no cases that can switch with Case 1 without violating any constraints, the mutation operator moves to step 4, where it checks if there are cases that can switch with Case 1 without violating Constraints 1, 2, 5, and 6. As shown in the table, no such case exists. Therefore, mutation step 5 is carried out, where Case 1 is rescheduled in a randomly chosen OR at the earliest possible time where no constraints are violated. Figure 5-9 demonstrates how this would occur if OR 2 is randomly chosen.

Table 5-8 Outcome of switching Case 1 with other cases in the example offspring

Case	Constraint Violation Outcome	Constraint Violation No.
2	Case 2 would be performed in OR 1 and start at Time4.	1, 5
3	Case 3 would start at Time4.	5
4	Case 4 would start at Time4.	5
5	Case 5 would start at Time4.	6
6	Case 6 would start at Time4.	5

Offspring before Mutation				
	Time1	**Time2**	**Time3**	**Time4**
OR 1	Case 5		Case 6	Case 1
OR 2	Case 3			
OR 3	Case 2		Case 4	

Offspring after Mutation				
	Time1	**Time2**	**Time3**	**Time4**
OR 1	Case 5		Case 6	
OR 2	Case 3			Case 1
OR 3	Case 2		Case 4	

Figure 5-9 Example of mutation after step 5

In Figure 5-10, Case 6 has been chosen as case *x*. Table 5-8 lists why there are no cases that can switch with Case 6 without violating constraints. However, Cases 3 – 5 can switch without violating Constraints 1, 2, 5, and 6. Therefore, one of them is randomly chosen as case *y* and removed (step 6). In our example, Case 5 has been chosen. Because Cases 5 and 6 were originally scheduled in the same OR (step 7), step 8 is performed. Since Case 6 was originally scheduled after Case 5, it is rescheduled first in the same OR at the earliest time where no constraints are violated. Case 5 is then rescheduled.

Table 5-9 Outcome of switching Case 6 with other cases in the example offspring

Case	Constraint Violation Outcome	Constraint Violation No.
1	Case 6 would start at Time4.	5
2	Case 2 would be performed in OR 1. Both Cases 1 and 2 would be scheduled in OR 1 at Time4.	1 4
3	Both Cases 2 and 6 would be scheduled at Time3. Both Cases 1 and 3 would be scheduled in OR 1 at Time4.	3 4
4	Both Cases 1 and 4 would be scheduled in OR 1 at Time4.	4
5	Both Cases 2 and 6 would be scheduled at Time1. Both Cases 4 and 5 would be scheduled at Time3.	3 3

Offspring before Mutation				
	Time1	**Time2**	**Time3**	**Time4**
OR 1	Case 5		Case 6	Case 1
OR 2	Case 3			
OR 3	Case 2		Case 4	

Offspring after Mutation				
	Time1	**Time2**	**Time3**	**Time4**
OR 1	Case 6	Case 5		Case 1
OR 2	Case 3			
OR 3	Case 2		Case 4	

Figure 5-10 Example of mutation after step 8

In Figure 5-11, Case 6 has again been chosen as case *x*. As explained in the previous example, this means that mutation step 6 will be reached. This time, Case 4 has been randomly chosen as case *y*. Because Cases 4 and 6 are not scheduled in the same OR (step 7), step 9 is carried out. Case 6 is rescheduled in Case 4's original OR at the earliest time where no constraints are violated, and vice versa.

Offspring before Mutation

	Time1	Time2	Time3	Time4
OR 1	Case 5		Case 6	Case 1
OR 2	Case 3			
OR 3	Case 2		Case 4	

Offspring after Mutation

	Time1	Time2	Time3	Time4
OR 1	Case 5	Case 4		Case 1
OR 2	Case 3			
OR 3	Case 2		Case 6	

Figure 5-11 Example of mutation after step 9

5.6.11 Replacement

The replacement process is similar to the one used in NSGA-II. First, the parent and offspring solutions are combined to form a new population of size 2*N*. Each solution must be checked for constraint violation so that the new population can be sorted according to non-domination. The solutions on the non-dominated front are chosen to become the next generation's population, and their crowding distances are calculated. If there are not enough solutions in this front to make a population of size *N*, solutions are then chosen from the second non-dominated front and their crowding distances are calculated. If there are still not enough solutions to make a complete population, solutions are progressively chosen from the next non-dominated fronts, and their crowding distances determined, until the population is full. All unused solutions are then deleted.

The solutions in the last allowed front may contain more solutions than needed. In this case, the following actions are taken:

1. Sort each solution in the last allowed front into sets according to their objective values. Solutions in the same set will have the same values for each objective, and therefore the same crowding distance.
2. In each set, delete any solution that is a duplicate of another. Therefore, the remaining solutions $n_r = 1,\ldots,N_r$, in each set $r = 1,\ldots,R$, will be unique.
3. Arrange each set according to descending crowding distance. Therefore, set $r = 1$ will have the largest distance, set $r = 2$ will have the second largest, and so on.
4. $r = 1$

   $n_r = 1$ for all $r = 1,\ldots,R$
5. For each space remaining in the next generation's population:
   a. Select solution n_r in set r to join the next generation
   b. $n_r = n_r + 1$
   c. If $\sum_{r=1}^{R} n_r = R + \sum_{r=1}^{R} N_r$, then:

      $r = 1$

      $n_r = 1$ for all $r = 1,\ldots,R$

   Otherwise:

      i. If $r = R$, then:

         $r = 1$

      Otherwise:

         $r = r + 1$

      ii. While $n_r > N_r$

Repeat step i.

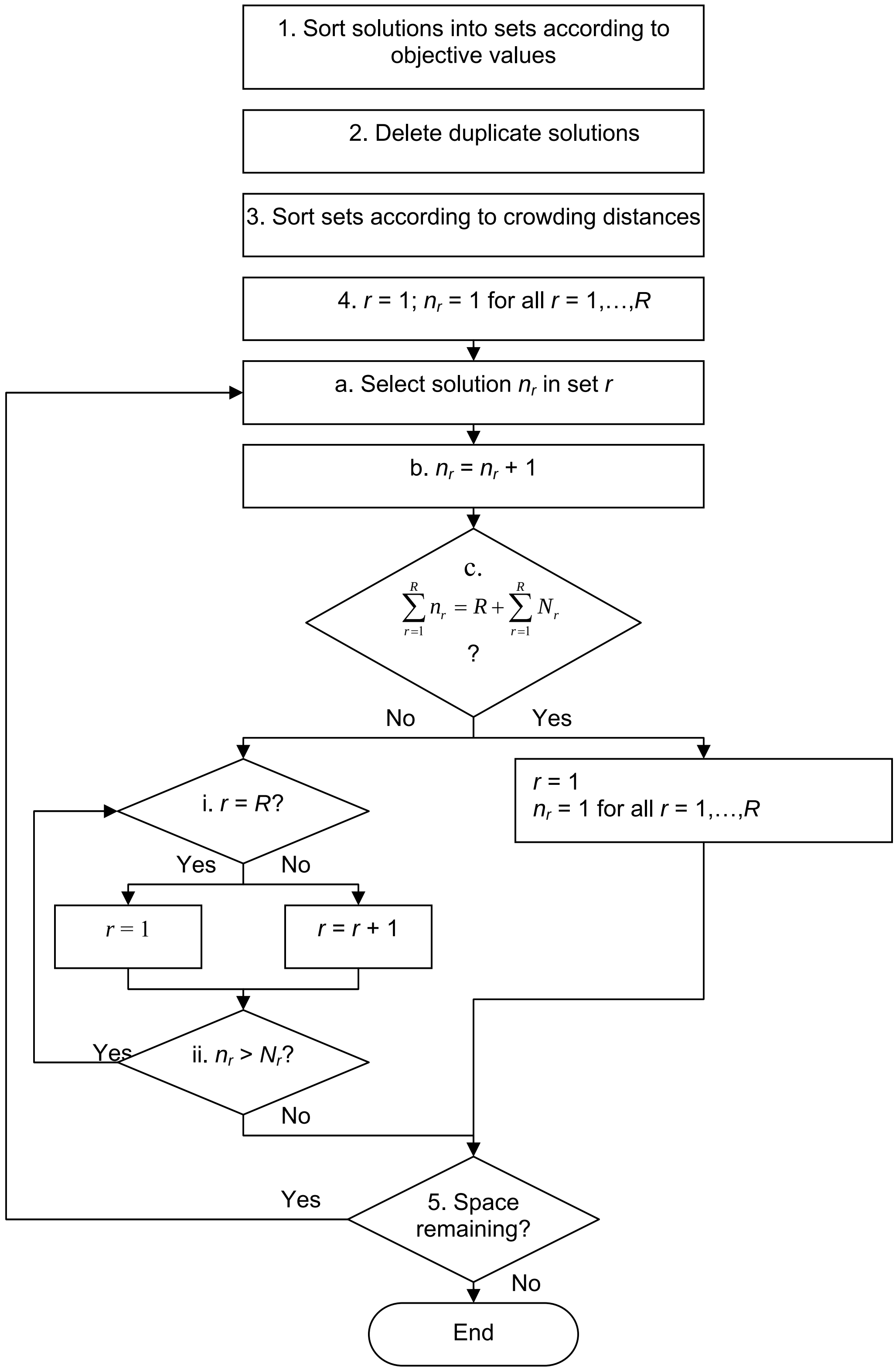

Figure 5-12 Flowchart of NSGAII-OR's replacement process for last allowed front

This replacement process for the last allowed front works by grouping solutions according to their objective values and deleting any duplicate solutions. Each group is then sorted according to descending crowding distance. Depending on how much space is remaining in the population, solutions are then chosen from these groups according to their sorted order. A different solution will be chosen each time, until all solutions have been selected. The process then starts over again with the first solution in the first group, until there is no space remaining in the new population. As an example, Figure 5-13 depicts three sorted groups (i.e. Set A, B, and C) in a last allowed front, consisting of 3 (named A1-A3), 2 (named B1-B2), and 4 solutions (named C1-C4), respectively. When solutions are chosen from these three groups, their selection order will be A1, B1, C1, A2, B2, C2, A3, C3, and C4.

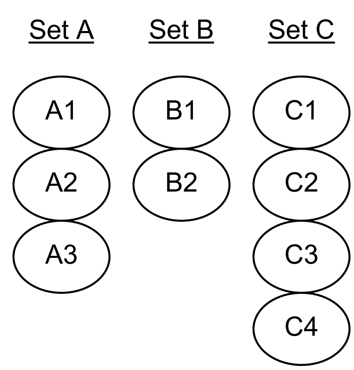

Figure 5-13 Example of solutions in last allowed front

In the original NSGA-II, the solutions having the largest crowding distances in the last allowed front are chosen to fill up the remaining space in the new population. For the problems addressed in this chapter, however, there are often a large number of unique solutions having the same objective values. If these solutions are in a low non-domination level and have very large crowding distances, they will be chosen for the next generation's population, resulting in a new population that is dominated by solutions with the same objective values. Hence, NSGA-II's replacement process was modified to select solutions in the last allowed front according to both their crowding distances and objective values, helping to further maintain diversity for the problems at hand.

5.6.12 Parameter Settings

In order to find the appropriate settings for N (population size), p_c (crossover probability), and p_m (mutation probability), NSGAII-OR was tested on one of the HSC problems (i.e. Week 1 – Monday) addressed in stage 1. A 3^5 full factorial experimental layout was used, where each parameter was set at one of the five values shown in Table 5-10, resulting in 375 combinations. Three replications were carried out for each combination by fixing the random seed generator to three different values, resulting in 1125 experiments.

Table 5-10 List of values tested for each parameter

N	p_c	p_m
100	0.9	0.1
200	0.8	0.2
300	0.7	0.3
400	0.6	0.4
500	0.5	0.5

Theoretically, high N and p_c values should increase the possibility of converging to the true Pareto-optimal front. However, this comes with increased computational time for each generation. In order to take this relationship into account, experiments with smaller N and p_c settings were allowed to run for more generations. This was achieved by having a stopping criterion of 80000 function evaluations for each experiment, resulting in an average number of generations equal to 80000/(N* p_c). Therefore, the combination with the highest N and p_c values (i.e. 500 and 0.9, respectively) had a stopping criterion of roughly 178 generations, while the combination with the lowest N and p_c values (i.e. 100 and 0.5, respectively) had a stopping criterion of approximately 1600 generations.

For each experiment, the objective value sets obtained by each solution in the final non-dominated front were recorded. To measure each experiment's performance with respect to convergence and diversity, the following two metrics were used (Khor et al. 2005):

1. Generational Distance, *GD* (a convergence metric)

$$GD = \left(\frac{1}{n} \sum_{i=1}^{n} d_i^2 \right)^{1/2}$$

2. Spacing, *S* (a diversity metric)

$$S = \left[\frac{1}{n} \sum_{i=1}^{n} \left(d_i^{'} - \overline{d^{'}} \right)^2 \right]^{1/2} / \overline{d^{'}}, \text{ where } \overline{\mathrm{d}^{'}} = \frac{1}{n} \sum_{i=1}^{n} d_i^{'}$$

Smaller values of *GD* signify better convergence, while larger values of *S* indicate better diversity. In both equations, the term n corresponds to the number of unique objective value sets in the final non-dominated front after a particular run.

In the *S* equation, the term $d_i^{'}$ is equal to the Euclidean distance between unique member i and its nearest unique member in the final non-dominated front. In the *GD* equation, the term d_i is equal to the Euclidean distance between unique objective value set member i in the final non-dominated front and its nearest unique objective value set member in the true Pareto front. Since the problem's true Pareto-optimal solutions are unknown, the most optimal objective value set was used (i.e. 0 0 0 1) in the *GD* calculations. Therefore, although the results give an indication of how well a particular parameter combination performed, they are not definitive.

To determine each parameter's effects on *GD* and *S*, a statistical analysis of variance (ANOVA) test was carried out. Table 5-11 displays the resultant p values, while the full results are contained in Appendix A. Results show that the N value has a significant effect on the *GD* metric, along with the interaction between N and p_c. Similarly, results indicate that the N value has a significant effect on the S metric, along with the interaction between p_c and p_m.

Table 5-11 ANOVA results for the generational distance (*GD*) and spacing (*S*) metrics

Source	*P* Value	
	***GD* metric**	***S* metric**
N	0.000	0.008
p_c	0.906	0.147
p_m	0.619	0.891
$N^* p_c$	0.055	0.922
$N^* p_m$	0.854	0.374
$p_c^* p_m$	0.326	0.042
$N^* p_c^* p_m$	0.918	0.352

These ANOVA results were used to determine which $N^*p_c^*p_m$ combination to use for the HSC problems addressed in this chapter. Since results showed that N and N^*p_c have a significant effect on the *GD* metric, their variations were ranked according to their average performance for the *GD* metric. Similarly, since results showed that N and $p_c^*p_m$ have a significant effect on the S metric, their variations were also ranked according to their average performance for the S metric. Table 5-12 displays these ranks.

Table 5-12 GD ranks for *N* and $N*p_c$, and S ranks for *N* and p_c*p_m

N	*GD Rank*
100	1
200	2
300	4
400	5
500	3

$N*p_c$	*GD Rank*
100*0.9	10
100*0.8	5
100*0.7	1
100*0.6	2
100*0.5	3
200*0.9	6
200*0.8	7
200*0.7	4
200*0.6	14
200*0.5	22
300*0.9	19
300*0.8	20
300*0.7	12
300*0.6	9
300*0.5	21
400*0.9	17
400*0.8	24
400*0.7	25
400*0.6	15
400*0.5	16
500*0.9	13
500*0.8	8
500*0.7	23
500*0.6	18
500*0.5	11

N	*S Rank*
100	5
200	4
300	1
400	3
500	2

p_c*p_m	*S Rank*
0.9*0.1	8
0.9*0.2	2
0.9*0.3	21
0.9*0.4	3
0.9*0.5	16
0.8*0.1	9
0.8*0.2	15
0.8*0.3	24
0.8*0.4	10
0.8*0.5	23
0.7*0.1	18
0.7*0.2	17
0.7*0.3	4
0.7*0.4	18
0.7*0.5	1
0.6*0.1	20
0.6*0.2	11
0.6*0.3	12
0.6*0.4	22
0.6*0.5	25
0.5*0.1	7
0.5*0.2	13
0.5*0.3	6
0.5*0.4	5
0.5*0.5	14

In order to determine which $N*p_c*p_m$ combination performed the best overall, each combination was assigned a *performance value* equal to the summation of its individual *GD rank* and *S rank*. Since there are 25 variations for $N*p_c$ and p_c*p_m, and only 5 variations for *N*, the *GD rank* and *S rank* for *N* were multiplied by 5 in order to give them the same weight. For example, the $N*p_c*p_m$ combination of 100*0.9*0.1 has a performance value equal to (1*5) + 10 + (5*5) + 8 = 48. Since the $N*p_c*p_m$ combination of 100*0.7*0.5 had the lowest, and therefore best, performance value, these parameters were used in NSGAII-OR for each problem addressed in this chapter.

5.6.13 Execution of NSGAII-OR for Stage 2

NSGAII-OR was developed in MATLAB[8], a high-level programming language. The stopping criterion for each run was set at 4 hours, which is a reasonable amount of computational time to find good solutions without being too long.

5.6.13.1 Test Problems

The first set of test problems solved by the mathematical model for stage 2 was also used to test NSGAII-OR. The solutions were manually validated to ensure that the schedules were sound. Table 5-13 displays the computational results for these test problems, including the different sets of objective values obtained for each problem. The term *Gen* corresponds to the number of generations reached, and *U* represents the number of unique solutions in the final non-dominated front. All non-dominated solutions were feasible. Note that because different schedules may result in the same objective values, the number of objective value sets is less than the number of unique non-dominated solutions.

[8] Copyright © The MathWorks, Inc.

Table 5-13 Test 1 results for stage 2, using NSGAII-OR

Day	*Gen*	*U*	Objective Values (f_1 f_2 f_3 f_4)					
			Set 1	Set 2	Set 3	Set 4	Set 5	Set 6
Mon	22673	100	5 2 3 1	10 2 2 1				
Tue	24844	100	2 2 4 2	3 2 3 2	3 3 4 1	4 2 2 1		
Wed	23866	42	5 0 3 2	5 0 4 1	5 2 3 1	8 0 3 1	10 3 2 1	11 0 2 1
Thu	24495	100	2 4 4 2	3 2 3 2	3 4 3 1	4 2 3 1	5 2 2 1	19 2 1 1
Fri	23600	100	3 4 3 1	4 4 2 1				

To compare the test results of NSGAII-OR with the stage 2 mathematical model, the set of objective values that best minimized the model's objective functions (i.e. Priority Levels 1, 2, and 3: Minimize $z = 2 * b_{1a} + b_{1b}$, $z = b_2$, and $z = b_3$ respectively) were selected for each problem. Table 5-14 compares the objective values obtained using the best schedules obtained by the mathematical model and NSGAII-OR's selected schedules. For Monday, Wednesday, and Friday, NSGAII-OR found solutions that neither dominated nor were dominated by the mathematical model's solutions. For Tuesday and Thursday, NSGAII-OR found slightly worse results (i.e. 1 half hour of overtime) regarding the first objective. However, keep in mind that for each problem, NSGAII-OR's computational time was only four hours while the mathematical model's computational time ranged between five and twelve hours.

Table 5-14 Comparison between the objective values obtained by the mathematical model's schedules and NSGAII-OR's selected schedules

	Objective Values (f_1 f_2 f_3 f_4)				
	Mon.	Tue.	Wed.	Thu.	Fri.
Model	4 2 5 3	1 2 4 2	4 0 4 2	2 2 3 2	2 4 4 1
NSGAII-OR	5 2 3 1	2 2 4 2	5 0 3 2	3 2 3 2	3 4 3 1

5.6.13.2 Health Sciences Centre (HSC) Problems

The HSC problems addressed in the previous chapter (for stage 1) were solved for stage 2 using NSGAII-OR. Table 5-15 displays the computational results. Again, the stopping criterion was set at four hours. The term *Gen* relates to the number of generations reached and *U* represents the number of unique solutions in the final non-dominated front. As explained in section 5.6.7, NSGAII-OR is able to ensure that Constraints 1 – 4 are not violated, but it cannot guarantee the same for Constraints 5 and 6 because there are certain problems that may not have any solutions which respect these two constraints. Therefore, for these types of cases, V_5 and V_6 give the amount of violation for Constraints 5 and 6, respectively.

Table 5-15 Results for stage 2, using NSGAII-OR

Week	Day	*Gen*	*U*	Feasible? (Yes=1,No=0)	V_5 (half hours)	V_6 (half hours)
1	Mon	9709	100	1	-	-
	Tue	10799	100	1	-	-
	Wed	13778	100	0	1	-
	Thu	10499	100	1	-	-
	Fri	9757	100	1	-	-
2	Mon	9159	100	1	-	-
	Tue	9244	100	1	-	-
	Wed	11487	100	0	-	1
	Thu	10356	100	1	-	-
	Fri	9103	100	1	-	-
3	Tue	10112	100	1	-	-
	Wed	12154	100	1	-	-
	Thu	9761	100	1	-	-
	Fri	9182	100	1	-	-
4	Mon	9100	100	1	-	-

	Tue	12091	100	1	-	-
	Wed	12133	100	0	1	-
	Thu	11306	100	1	-	-
	Fri	11682	100	1	-	-
5	Mon	10816	100	1	-	-
	Tue	10415	100	1	-	-
	Wed	12546	100	1	-	-
	Thu	12470	100	1	-	-
	Fri	11530	100	1	-	-

For the Wednesdays of weeks 1, 2, and 4, cases could not be scheduled without violating Constraints 5 and 6. However, the constraint violations were minimized to only 1 half hour. Constraints 5 and 6 were considered to some extent during the SD and Overnight assignment process in stage 1. However, the stage 1 mathematical model will need to be modified to fully consider all possible outcomes when assigning SD or Overnight status, so that feasible solutions always exist in stage 2.

Table 5-16 displays the different objective value sets obtained for each problem using NSGAII-OR. The objective values in the shaded boxes belong to the schedules selected for comparison with the actual schedules used at HSC, and were selected based on how well they minimize the mathematical model's objective functions (i.e. Priority Levels 1, 2, and 3: Minimize $z = 2 * b_{1a} + b_{1b}$, $z = b_2$, and $z = b_3$, respectively). On average, the selected NSGAII-OR schedules will result in a maximum of 1.8 hours of overtime in any operating theatre (i.e. f_1), and a maximum deviation of 1.7 hours with regards to OR time utilization between any two operating theatres (i.e. f_2). If an OR department wished to employ this scheduling system, they should generate schedules sufficiently far in advance so that staffing can be adjusted to meet the schedule, and hence avoid overtime costs.

Table 5-16 Objective values obtained for stage 2, using NSGAII-OR

Week		Objective Values (f_1 f_2 f_3 f_4)										
		Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7	Set 8	Set 9	Set 10	Set 11
1	M	4 4 7 3	5 4 7 2	6 4 6 2	12 4 5 2							
	T	6 5 7 2	6 6 6 2	7 4 6 2	8 4 5 3	8 5 5 2	10 3 6 2	11 3 5 2	23 7 4 2			
	W	3 3 5 2	4 3 4 3	6 3 4 2								
	Th	4 8 5 3	6 8 4 2									
	F	4 3 7 2	6 3 6 3	6 6 6 2	7 2 6 2	11 3 5 2	17 4 4 2	20 2 5 2	20 3 4 2			
2	M	4 3 8 2	4 4 7 2	6 3 7 3	8 2 8 3	9 2 7 3	10 2 6 2	12 3 5 2				
	T	4 2 7 3	4 6 6 3	4 6 7 2	5 5 6 3	8 4 7 2	8 6 6 2	9 3 6 3	10 2 6 3	10 2 7 2	10 3 6 2	12 2 6 2
	W	0 3 6 4	2 2 6 2	4 4 5 4	5 4 5 3	5 5 5 2	6 2 5 2					
	Th	1 4 7 3	1 4 8 2	2 2 6 2	7 6 5 2	9 3 5 2						
	F	3 4 6 3	4 3 6 2	5 3 5 2	7 3 4 2							
3	T	7 5 6 3	7 5 7 2	7 7 6 2	7 8 5 3	8 5 6 2	8 7 5 3	12 9 5 2	15 5 5 2	20 11 4 2		
	W	2 2 6 3	6 2 5 2									
	Th	1 4 8 4	5 5 6 2	6 3 5 2	7 4 4 2							
	F	3 3 6 2	4 2 6 3	5 8 5 2	6 2 6 2	6 3 5 2	7 2 5 2	10 10 4 2				
4	M	5 4 6 3	6 7 7 2	8 4 6 2	10 3 6 3	11 7 5 2	12 3 5 2					
	T	1 4 6 3	2 5 6 2	3 4 5 3	3 5 5 2	4 4 6 2	5 4 5 2	6 3 5 3	8 5 4 2	11 3 4 2		
	W	4 2 6 2	6 2 5 2	7 1 6 2	13 1 5 2							
	Th	2 3 7 3	4 4 8 2	4 5 7 2	4 6 6 3	5 3 7 2	6 7 6 2	6 9 5 2	8 3 6 2	8 5 5 2	10 3 5 2	
	F	1 3 6 3	3 4 5 3	5 4 5 2	6 3 5 2	6 4 4 2						
5	M	8 3 4 2	12 4 3 2									
	T	4 3 6 3	5 4 6 2	6 4 5 3	8 4 5 2	10 3 5 2	10 4 4 2					
	W	7 2 5 2										
	Th	3 3 7 2	3 5 6 2	5 4 6 2	5 5 5 2	8 3 6 2	8 4 4 2					
	F	5 3 5 2	5 5 4 2	7 3 4 2								

Figure 5-14 compares the largest number of PACU beds occupied at any time (i.e. f_3), based on the actual schedules employed at HSC and the selected schedules generated by NSGAII-OR. On average, NSGAII-OR reduced these numbers by 31.9%. At the busiest time in PACU, at least 15 PACU beds were needed due to HSC's schedules. The selected NSGAII-OR schedules, however, would have only required a maximum of 8 PACU beds. This demonstrates that more cases could likely be performed without increasing resources, simply by smoothing resource utilization variability.

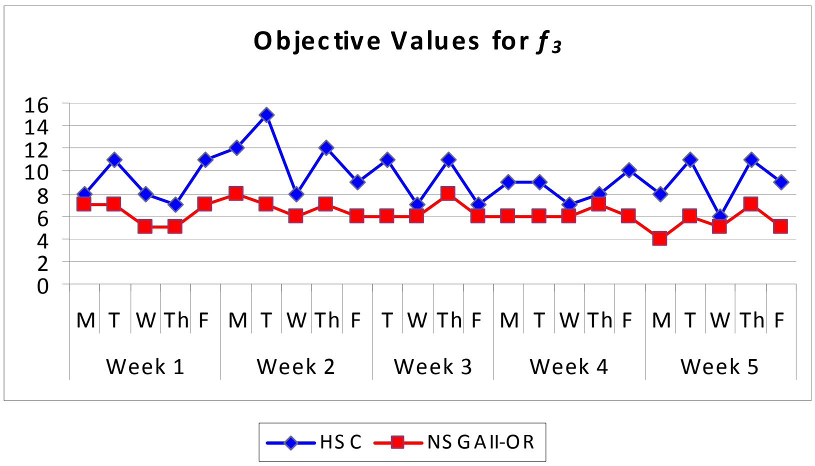

Figure 5-14 Comparison between f_3 values obtained by HSC's schedules and NSGAII-OR's selected schedules

Similarly, Figure 5-15 compares the largest number of cases that are scheduled to finish at the same time (i.e. f_4), resulting from HSC's schedules and NSGAII-OR's selected schedules. On average, NSGAII-OR's selected schedules lowered these numbers by 57.1%. According to the schedules used by HSC, up to 9 cases finished simultaneously. On the other hand, only a maximum of 4 cases would have finished at the same time if NSGAII-OR's selected schedules had been used.

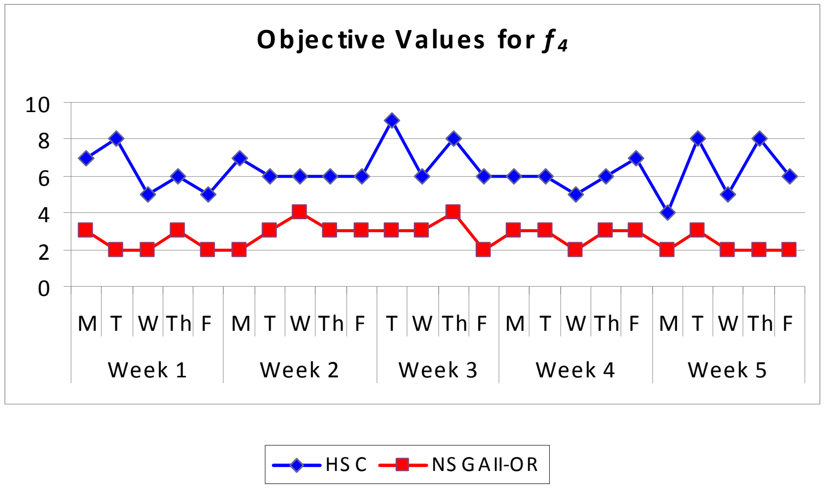

Figure 5-15 Comparison between f_4 values obtained by HSC's schedules and NSGAII-OR's selected schedules

5.7 Summary

This chapter presented the second stage of the proposed elective operating room (OR) scheduling system, based on the elective surgical patient flow at Health Sciences Centre (HSC) in Winnipeg, Manitoba. In this stage, cases are assigned to operating theatres and given start times. The resultant daily schedules reduce the artificial variability occurring within each day, based on four objectives classified into three priority levels. The first priority level minimizes overtime and balances utilized OR time among each operating theatre, in order to reduce costs and staff dissatisfaction. In the second priority level, Post-Anaesthesia Care Unit (PACU) bed occupancy is evenly distributed throughout the day in order to decrease delays caused by an insufficient number of PACU beds during peak periods. Finally, the third priority level minimizes the number of cases that finish simultaneously, so that delays caused by an insufficient number of peri-operative aides (PAs) to turnover the operating theatres during peak periods are reduced.

For this stage, the problem was first mathematically modelled and solved using lexicographic goal programming. The mathematical model was tested on two sets of test problems, where it was discovered that optimal solutions could not be obtained for the majority of even the smallest problems. Furthermore, solution times greatly increased with only slight increases in the number of cases to be scheduled. Therefore, a second optimization method was developed for solving the problems at hand.

This second method is a genetic algorithm, called Nondominated Sorting Genetic Algorithm II for Operating Room Scheduling (NSGAII-OR). Compared to the mathematical model, NSGAII-OR was able to generate comparable solutions to the test problems in shorter amounts of time. Compared to the actual schedules employed at HSC during the five week period addressed in stage 1, NSGAII-OR reduced the largest number of PACU beds occupied during any period of the day by an average of 31.9%, and the largest number of cases that finish simultaneously during any period of the day by an average of 57.1%. Therefore, NSGAII-OR was able to successfully generate schedules with less variability within each day, regarding these two objectives. With modest programming knowledge, NSGAII-OR can be tailored to meet the needs of many other facilities by adjusting, adding, or removing constraints or objectives.

Two of the constraints in this stage relate to Same Day (SD) and Overnight patients being scheduled early enough to allow for sufficient recovery by their post-operative unit's discharge deadline. For three particular problems addressed in this chapter, there was no way to schedule cases without violating one of these constraints. Although the mathematical model for the first stage did consider these constraints to some extent when assigning patients SD or Overnight status, it should be modified in future research to fully grasp all scheduling possibilities as a result of its assignments. Another alternative to addressing these constraints would be to create a single multi-objective optimization method that can schedule in just one step, rather than two. However, this will be a very complex problem that is difficult to solve. Furthermore, one will likely have to remove some objectives in order to make the problem more tractable.

Chapter 6: Conclusions and Future Research

Recent evaluations on the causes of delays in healthcare facilities have concluded that delays may be significantly reduced by improving patient flow. When patients, staff, information, and materials do not timely and efficiently flow through a hospital, staff and patient satisfaction, costs, and patient safety are all negatively affected. Based on this information, Health Sciences Centre (HSC) in Winnipeg initiated a project to analyze its adult surgical patient flow and generate possible methods of improvement. As a result, the researchers involved concluded that the largest barrier to smooth patient flow was the way in which elective surgeries were being scheduled in the operating room (OR) department. This led to the development of a new elective OR scheduling system for HSC, which is presented in this thesis.

At HSC, as in most facilities, surgical cases are scheduled through a block scheduling policy in which surgical dates are chosen by surgeons. However, resources may be better utilized when a facility books cases through an open scheduling policy, where they have control over surgical date, operating theatre, and start times assignments. Therefore, the proposed elective OR scheduling system generates weekly elective OR schedules with this control.

Before the scheduling system can be carried out, HSC must first allocate hours of elective OR time to each surgical group during a particular week. Each group must then submit a list of cases that they wish to perform during that week, provided that their allocated OR time has not been exceeded. The OR department can then use the proposed elective OR scheduling system to generate complete OR schedules.

In the proposed system, weekly elective OR schedules are generated in two stages. In the first stage, cases are assigned to the different days of the week so that day-to-day artificial variability is reduced. Because the objectives in this stage have very distinct priorities, a biased search for an optimal solution can be conducted. This thesis demonstrates that the problem can be mathematically modelled and solved using lexicographic goal programming, which is suitable for multi-objective problems where no trade-offs between objectives in different priority levels are allowed.

In the second stage of the proposed system, the cases assigned to each day are given operating theatre and start time assignments so that artificial variability occurring within a day is reduced. Like the first stage, the second stage has distinct priorities for each objective, and hence a biased search for an optimal solution can be used. However, this thesis demonstrates that lexicographic goal programming is unsuitable for the problems in this stage, due to its inability to handle the problem sizes. Alternatively, a Nondominated Sorting Genetic Algorithm II for Operating Room Scheduling (NSGAII-OR) was developed and shown to be a feasible optimization method for the second stage.

Both stages were tested using data pertaining to elective cases performed during a five week period at HSC. Compared to the actual schedules employed at HSC, the stage 1 lexicographic programming model was able to reduce the largest daily bed occupancy variability for any service by an average of 25.1%, and the bed occupancy on the days following the scheduled week by an average of 19.9%. In addition, the model reduced total daily Same Day (SD) and Same Day Admission (SDA) patient volume variability by an average of 94.3%, while minimizing daily OR utilization variability to only one hour. The stage 2 NSGAII-OR was able to reduce the largest number of Post-Anaesthesia Care Unit (PACU) beds occupied during any period of the day by an average of 31.9%, and the largest number of cases that finish simultaneously during any period of the day by an average of 57.1%. These results suggest that there should be less artificial variability if the proposed elective OR scheduling system was actually implemented at HSC. Subsequently, there should be an improvement in the facility's surgical patient flow.

6.1 Alternatives

The proposed elective OR scheduling system removes the control over choosing each case's surgical date and start time from its surgeons. In addition, surgeons and patients will not be aware of their upcoming surgical schedules until the complete OR schedule has been generated, which is dependent on when each surgical group submits their selected cases, along with when the OR department decides to employ the proposed optimization methods. These two issues directly impact surgeon satisfaction and will probably be the major barrier to implementing the proposed scheduling system at HSC. As a way of alleviating some of these issues, HSC can use the proposed system to create a few cyclic weekly OR schedules that rotate throughout the year. This will allow the surgeons, patients, and staff to be aware of the upcoming elective OR schedule, weeks or months in advance. Alternatively, the proposed system can be used for only scheduling cases that belong to specific surgical groups that are able to submit their selected cases far in advance.

If desired, HSC can choose to only use the first stage of the proposed system. After each case is assigned to a particular day, its surgeon(s) can work together with the OR department to come up with an operating theatre and start time assignments. In this way, at least the objectives in the first stage will be addressed, while some control is returned to the surgeons.

6.2 Future Research

For three particular problems addressed in the second stage, there was no way to schedule cases such that some constraints were not violated. Although the mathematical model in the first stage did consider those constraints to some extent, it will need to be modified so that feasible solutions will always exist in the second stage.

Alternatively, elective OR schedules may be created in just one step. In this way, all of the objectives and constraints will be considered simultaneously. Moreover, the objectives in the second stage will be optimized without being constrained by the results of first stage. Future research will involve the modification of NSGAII-OR to achieve this, allowing for complete schedules to be created in just a single run by using only one multi-objective optimization method. However, this will increase the complexity of the problems being solved, and some objectives may need to be removed in order to make the problems easier to handle.

Future work will also include incorporating techniques into the proposed elective OR scheduling system so that it can accommodate any unforeseen events that may occur after a schedule has been generated. This is important because unforeseen events often occur in the OR department, such as the arrival of emergency cases. One simple way of accommodating these events would be to erase existing schedules and generate entirely new ones. However, this is an impractical strategy because the process will be time consuming, the allocation of resources will be disrupted, and the strategy will likely lead to failure. Therefore, it would be better to adjust old schedules to meet new events. This process of adaptation is called *reactive scheduling*, and it is essential in making the proposed scheduling system a well-rounded success.

Appendix A

Table A-1 Results for stage 1 – Only Priority Level 1

Week	No. of Variables	No. of Constraints	CPU (sec)	Priority Level	Objective Function	Z	B
1	10337	6619	31	1	$2*b_{1a}/B1a + b_{1b}/B1b$	3	3
				2	b_2	48	
				3	b_3	6	
2	10853	7376	48	1	$2*b_{1a}/B1a + b_{1b}/B1b$	3.058	3.058
				2	b_2	38	
				3	b_3	7	
3	9843	5747	15	1	$2*b_{1a}/B1a + b_{1b}/B1b$	3.015	3.015
				2	b_2	34	
				3	b_3	1	
4	10523	6956	5	1	$2*b_{1a}/B1a + b_{1b}/B1b$	3	3
				2	b_2	44	
				3	b_3	5	
5	10499	6797	11	1	$2*b_{1a}/B1a + b_{1b}/B1b$	3	3
				2	b_2	46	
				3	b_3	6	

Table A-2 Results for stage 1 – Only Priority Level 2

Week	No. of Variables	No. of Constraints	CPU (sec)	Priority Level	Objective Function	Z	B
1	3218	5550	14400	2	b_2	25*	23.8
				1	$2*b_{1a}/B1a + b_{1b}/B1b$	3.263	
				3	b_3	11	
2	3734	6307	605	2	b_2	24	23.4
				1	$2*b_{1a}/B1a + b_{1b}/B1b$	6.722	
				3	b_3	7	
3	2724	4676	65	2	b_2	17	16.75
				1	$2*b_{1a}/B1a + b_{1b}/B1b$	5.149	
				3	b_3	1	
4	3404	5887	185	2	b_2	20	20
				1	$2*b_{1a}/B1a + b_{1b}/B1b$	3.574	
				3	b_3	3	
5	3380	5728	107	2	b_2	25	24.4
				1	$2*b_{1a}/B1a + b_{1b}/B1b$	4.589	
				3	b_3	8	

Table A-3 Results for stage 1 – Only Priority Level 3

Week	No. of Variables	No. of Constraints	CPU (sec)	Priority Level	Objective Function	Z	B
1	3258	5590	52	3	b_3	0	0
				1	$2*b_{1a}/B1a + b_{1b}/B1b$	3.471	
				2	b_2	65	
2	3774	6347	54	3	b_3	0	0
				1	$2*b_{1a}/B1a + b_{1b}/B1b$	6.026	
				2	b_2	62	
3	2764	4700	45	3	b_3	0	0
				1	$2*b_{1a}/B1a + b_{1b}/B1b$	5.179	
				2	b_2	22	
4	3444	5927	65	3	b_3	0	0
				1	$2*b_{1a}/B1a + b_{1b}/B1b$	3.165	
				2	b_2	52	
5	3420	5768	8	3	b_3	0	0
				1	$2*b_{1a}/B1a + b_{1b}/B1b$	3.956	
				2	b_2	58	

Table A-4 Results for stage 1 – 1-3-2

Week	Priority Level	Objective Function	No. of Variables	No. of Constraints	CPU (sec)	Z	B
1	1	$2*b_{1a}/B1a + b_{1b}/B1b$	10137	6609	9	3	3
	3	b_3	10381	6664	45	0	0
	2	b_2	10396	6679	775	24	23.8
2	1	$2*b_{1a}/B1a + b_{1b}/B1b$	10643	7366	27	3.058	3.058
	3	b_3	10897	7421	118	0	0
	2	b_2	10912	7436	11727	24	23.4
3	1	$2*b_{1a}/B1a + b_{1b}/B1b$	9628	5749	9	3.015	3.015
	3	b_3	9911	5781	37	0	0
	2	b_2	9902	5789	215	17	16.75
4	1	$2*b_{1a}/B1a + b_{1b}/B1b$	10308	6946	8	3	3
	3	b_3	10567	7001	101	0	0
	2	b_2	10582	7016	14400	27*	20
5	1	$2*b_{1a}/B1a + b_{1b}/B1b$	10294	6787	10	3	3
	3	b_3	10543	6842	24	0	0
	2	b_2	10558	6857	469	25	24.4

Table A-5 Results for stage 1 – 2-1-3

Week	Priority Level	Objective Function	No. of Variables	No. of Constraints	CPU (sec)	Z	B
1	2	b_2	3213	5545	14400	25	23.8*
	1	$2*b_{1a}/B1a + b_{1b}/B1b$	10342	6624	43	3	3
	3	b_3	10396	6679	297	0	0
2	2	b_2	3729	6302	725	24	23.4
	1	$2*b_{1a}/B1a + b_{1b}/B1b$	10858	7381	61	3.058	3058
	3	b_3	10912	7436	14400	1*	0
3	2	b_2	2743	4680	6	17	16.75
	1	$2*b_{1a}/B1a + b_{1b}/B1b$	9872	5759	26	3.015	3.015
	3	b_3	9902	5789	60	0	0
4	2	b_2	3399	5882	14400	22*	20
	1	$2*b_{1a}/B1a + b_{1b}/B1b$	10528	6961	14400	N/A	3
	3	b_3	-	-	-	-	-
5	2	b_2	3375	5723	24	25	24.4
	1	$2*b_{1a}/B1a + b_{1b}/B1b$	10504	6802	107	3	3
	3	b_3	10558	6857	171	0	0

Table A-6 Results for stage 1 – 2-3-1

Week	Priority Level	Objective Function	No. of Variables	No. of Constraints	CPU (sec)	Z	B
1	2	b_2	3213	5545	14400	25	23.8*
	3	b_3	3268	5600	78	0	0
	1	$2*b_{1a}/B1a + b_{1b}/B1b$	10397	6679	1804	3	3
2	2	b_2	3729	6302	725	24	23.4
	3	b_3	3784	6357	301	0	0
	1	$2*b_{1a}/B1a + b_{1b}/B1b$	10913	7436	499	3.058	3.058
3	2	b_2	2743	4680	6	17	16.75
	3	b_3	2798	4717	48	0	0
	1	$2*b_{1a}/B1a + b_{1b}/B1b$	9928	5789	200	3.015	3.15
4	2	b_2	3399	5882	14400	22*	20
	3	b_3	3454	5937	44	0	0
	1	$2*b_{1a}/B1a + b_{1b}/B1b$	10583	7016	9952	3	3
5	2	b_2	3375	5723	24	25	24.4
	3	b_3	3430	5778	6047	0	0
	1	$2*b_{1a}/B1a + b_{1b}/B1b$	10559	6857	19	3	3

Table A-7 Results for stage 1 – 3-1-2

Week	Priority Level	Objective Function	No. of Variables	No. of Constraints	CPU (sec)	Z	B
1	3	b_3	3253	5585	10	0	0
	1	$2^*b_{1a}/B1a + b_{1b}/B1b$	10382	6664	68	3	3
	2	b_2	10396	6679	775	24	23.8
2	3	b_3	3769	6342	20	0	0
	1	$2^*b_{1a}/B1a + b_{1b}/B1b$	10898	7421	161	3.058	3.058
	2	b_2	10912	7436	11727	24	23.4
3	3	b_3	2783	4702	15	0	0
	1	$2^*b_{1a}/B1a + b_{1b}/B1b$	9912	5781	16	3.015	3.015
	2	b_2	9927	5789	215	17	16.75
4	3	b_3	3439	5922	65	0	0
	1	$2^*b_{1a}/B1a + b_{1b}/B1b$	10568	7001	11	3	3
	2	b_2	10582	7016	14400	27*	20
5	3	b_3	3415	5763	50	0	0
	1	$2^*b_{1a}/B1a + b_{1b}/B1b$	10544	6842	43	3	3
	2	b_2	10558	6857	469	25	24.4

Table A-8 Results for stage 1 – 3-2-1

Week	Priority Level	Objective Function	No. of Variables	No. of Constraints	CPU (sec)	Z	B
1	3	b_3	3253	5585	10	0	0
	2	b_2	3268	5600	14400	27*	23.8
	1	$2^*b_{1a}/B1a + b_{1b}/B1b$	10397	6679	14400	N/A	3
2	3	b_3	3769	6342	20	0	0
	2	b_2	3784	6357	243	24	23.4
	1	$2^*b_{1a}/B1a + b_{1b}/B1b$	10913	7436	499	3.058	3.058
3	3	b_3	2783	4702	15	0	0
	2	b_2	2798	4717	120	17	16.75
	1	$2^*b_{1a}/B1a + b_{1b}/B1b$	9927	5789	200	3.015	3.015
4	3	b_3	3439	5922	65	0	0
	2	b_2	3454	5937	14400	22*	20
	2	b_2	10583	7016	9952	3	3
5	3	b_3	3415	5763	50	0	0
	2	b_2	3430	5778	4315	25	24.4
	1	$2^*b_{1a}/B1a + b_{1b}/B1b$	10559	6857	19	3	3

Table A-9 Analysis of Variance (ANOVA) Results for *GD* Metric

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	593.665(a)	124	4.788	1.112	.241
Intercept	76391.093	1	76391.093	17737.982	.000
N	136.156	4	34.039	7.904	.000
p_c	4.399	4	1.100	.255	.906
p_m	11.415	4	2.854	.663	.619
$N^* p_c$	114.242	16	7.140	1.658	.055
$N^* p_m$	43.770	16	2.736	.635	.854
$p_c^* p_m$	77.996	16	4.875	1.132	.326
$N^* p_c^* p_m$	205.688	64	3.214	.746	.918
Error	1076.660	250	4.307		
Total	78061.419	375			
Corrected Total	1670.325	374			

a R Squared = .355 (Adjusted R Squared = .036)

Table A-10 GD Means for *N*

			95% Confidence Interval	
N	Mean	Std. Error	Lower Bound	Upper Bound
100	13.232	.240	12.760	13.704
200	14.007	.240	13.535	14.479
300	14.693	.240	14.221	15.165
400	14.931	.240	14.459	15.403
500	14.500	.240	14.028	14.972

Table A-11 GD Means for $N*p_c$

N	p_c	Mean	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
100	.5	13.055	.536	12.000	14.110
	.6	12.900	.536	11.844	13.955
	.7	12.811	.536	11.756	13.866
	.8	13.410	.536	12.355	14.465
	.9	13.985	.536	12.929	15.040
200	.5	15.288	.536	14.233	16.344
	.6	14.211	.536	13.156	15.266
	.7	13.286	.536	12.230	14.341
	.8	13.737	.536	12.682	14.792
	.9	13.514	.536	12.459	14.570
300	.5	15.220	.536	14.165	16.276
	.6	13.952	.536	12.896	15.007
	.7	14.171	.536	13.116	15.226
	.8	15.082	.536	14.027	16.137
	.9	15.041	.536	13.986	16.097
400	.5	14.472	.536	13.416	15.527
	.6	14.451	.536	13.396	15.507
	.7	15.833	.536	14.778	16.888
	.8	15.410	.536	14.355	16.465
	.9	14.490	.536	13.434	15.545
500	.5	14.124	.536	13.069	15.179
	.6	14.965	.536	13.910	16.021
	.7	15.293	.536	14.237	16.348
	.8	13.908	.536	12.853	14.964
	.9	14.208	.536	13.152	15.263

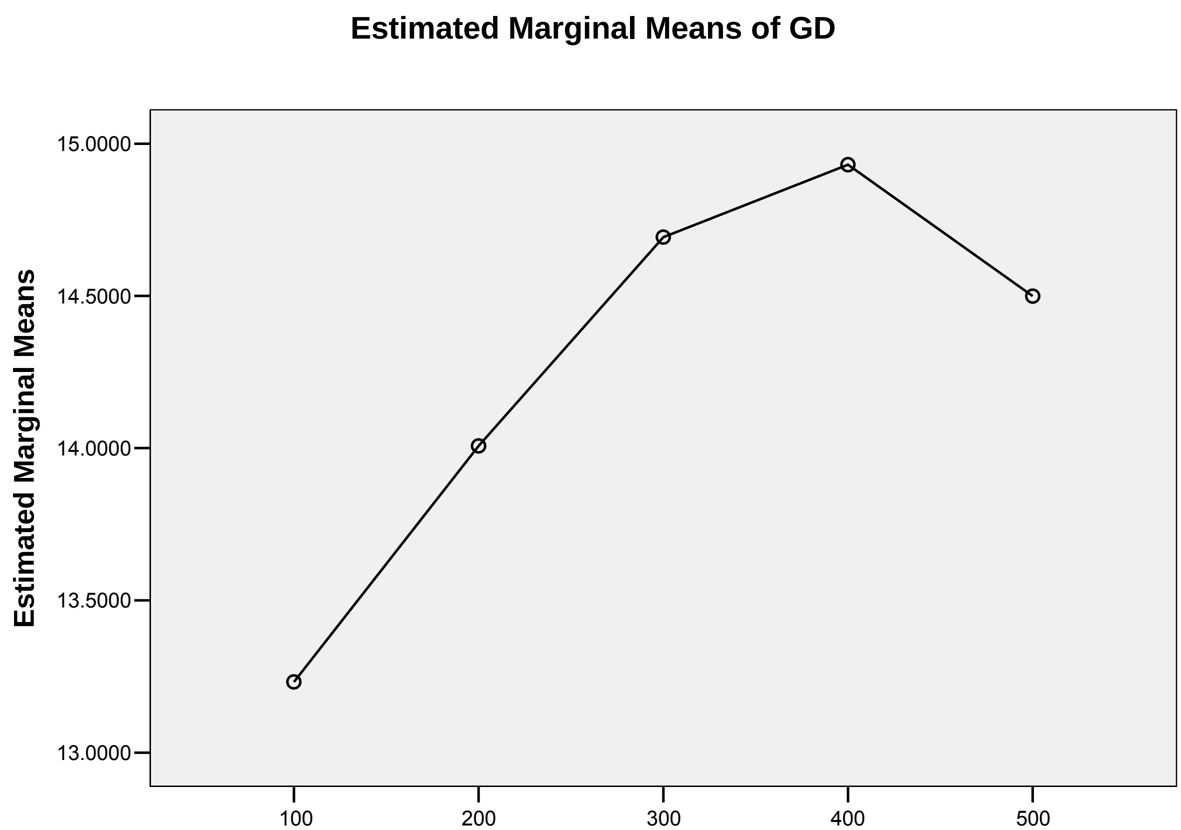

Figure A-1 GD Profile Plot for *N*

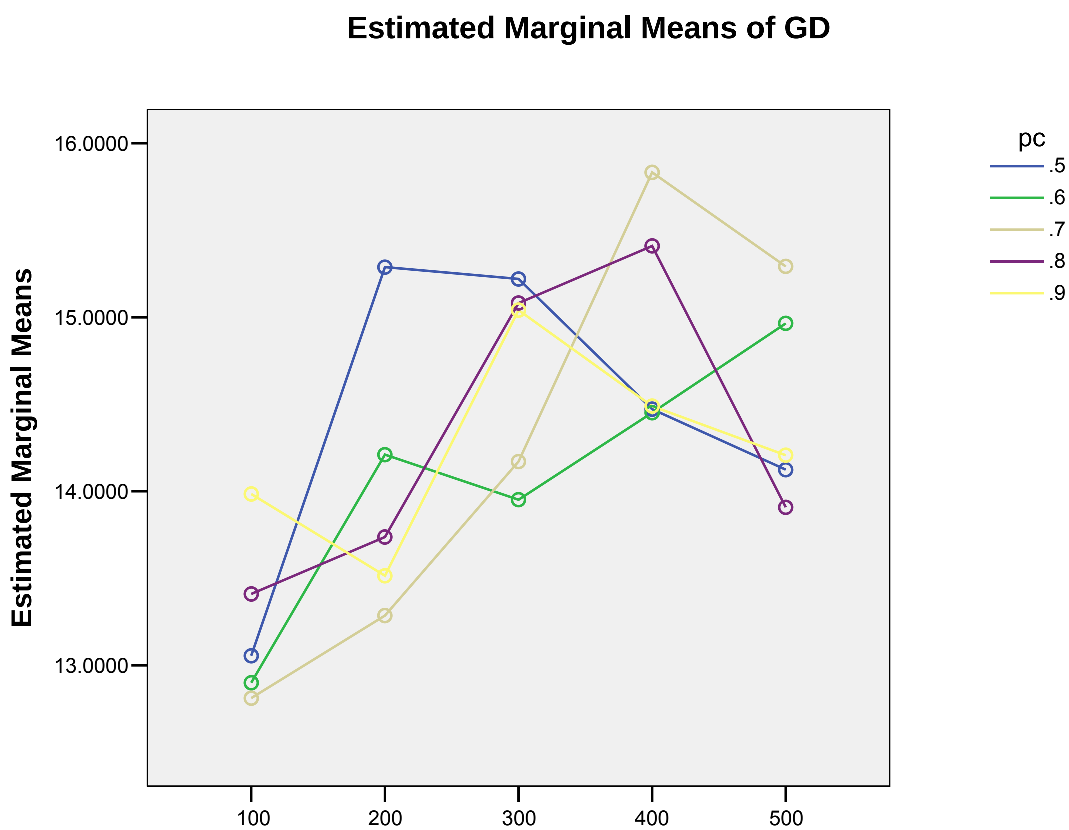

Figure A-2 GD Profile Plot for N*p_c

Table A-12 Analysis of Variance (ANOVA) Results for *S* Metric

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	11.117(a)	124	.090	1.163	.160
Intercept	80.496	1	80.496	1044.115	.000
N	1.089	4	.272	3.532	.008
p_c	.529	4	.132	1.715	.147
p_m	.086	4	.022	.279	.891
N* p_c	.670	16	.042	.543	.922
N*p_m	1.333	16	.083	1.080	.374
p_c*p_m	2.131	16	.133	1.728	.042
N* p_c*p_m	5.278	64	.082	1.070	.352
Error	19.274	250	.077		
Total	110.886	375			
Corrected Total	30.390	374			

a R Squared = .366 (Adjusted R Squared = .051)

Table A-13 S Means for *N*

N	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
100	.388	.032	.325	.451
200	.410	.032	.347	.473
300	.525	.032	.461	.588
400	.495	.032	.432	.558
500	.499	.032	.436	.562

Table A-14 S Means for p_c*p_m

p_c	p_m	Mean	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
.5	.1	.529	.072	.387	.670
	.2	.436	.072	.295	.578
	.3	.535	.072	.394	.676
	.4	.548	.072	.407	.689
	.5	.430	.072	.289	.572
.6	.1	.400	.072	.258	.541
	.2	.486	.072	.344	.627
	.3	.481	.072	.340	.622
	.4	.373	.072	.232	.515
	.5	.303	.072	.161	.444
.7	.1	.405	.072	.263	.546
	.2	.408	.072	.267	.549
	.3	.556	.072	.415	.697
	.4	.405	.072	.264	.547
	.5	.665	.072	.523	.806
.8	.1	.503	.072	.362	.644
	.2	.428	.072	.287	.570
	.3	.346	.072	.204	.487
	.4	.495	.072	.354	.637
	.5	.366	.072	.225	.508
.9	.1	.519	.072	.378	.660
	.2	.597	.072	.456	.738
	.3	.391	.072	.250	.532
	.4	.568	.072	.426	.709
	.5	.410	.072	.269	.551

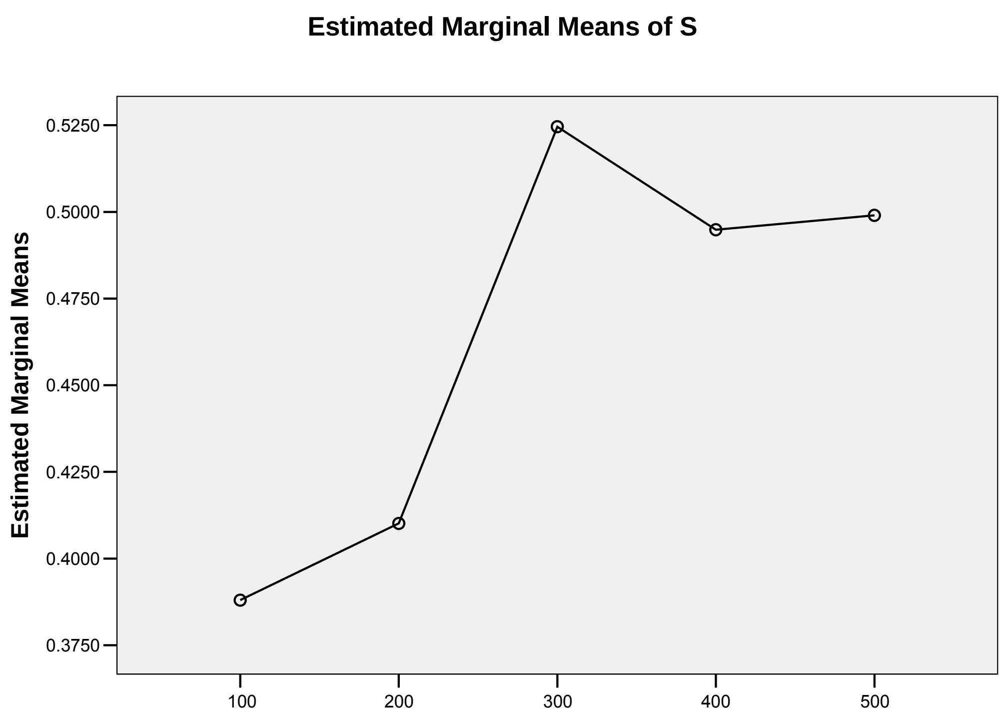

Figure A-3 S Profile Plot for *N*

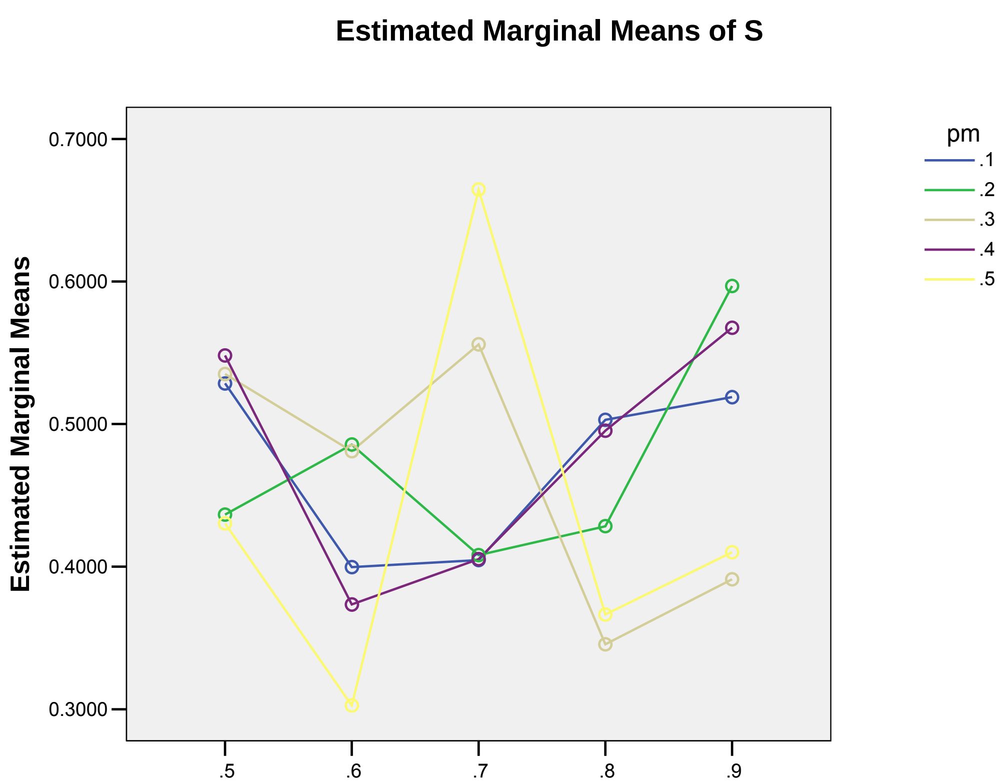

Figure A-4 S Profile Plot for p_c*p_m

Appendix B

B.1 Analysis of the Stage 1 Mathematical Model

B.1.1 Relationship between Objectives

To understand the relationship between the objectives in different priority levels, each HSC problem was solved by only considering the objectives in one priority level. The first row of Table 4-11 depicts the average objective value obtained for each priority level, when the original mathematical model was used to solve the five HSC problems. The subsequent rows display the same values for when the model only took the objectives in priority level one, two, or three into account. The computational results of each test are given in Appendix A.

Table B-1 Comparison between the results obtained by the original mathematical model and variations where the objectives in only one priority level were considered

	Objectives		
	Priority Level 1 ($2*b_{1a}/B1a + b_{1b}/B1b$)	**Priority Level 2 (b_2)**	**Priority Level 3 (b_3)**
Original Mathematical Model	3.015	22.2	0.8
Only Priority Level 1	3.015	42.0	5.0
Only Priority Level 2	4.659	22.2	6.0
Only Priority Level 3	4.359	51.8	0

From the results, it is evident that maximizing one priority level's objectives will not optimize another priority level's objectives. Hence, the problem was correctly assumed as multi-objective.

B.1.2 Relationship between Each Objective's Priority Level

The mathematical model was also tested on the five HSC problems by changing the order of priority levels. The first row of Table 4-12 depicts the average objective value obtained for each priority level in the original mathematical model. The subsequent rows display the same values for when the HSC problems were solved by using a different order of priority levels in the mathematical model. For example, 1-3-2 displays the results when the objectives in priority level one were considered first, followed by the objective in priority level three and finally priority level two. The computational results of each test are contained in Appendix A.

Table B-2 Comparison between the results obtained by the original mathematical model and variations where each objective's corresponding priority level was changed

	Objectives		
	Priority Level 1 ($2^*b_{1a}/B1a + b_{1b}/B1b$)	**Priority Level 2 (b_2)**	**Priority Level 3 (b_3)**
Original Mathematical Model	3.015	22.2	0.8
1-3-2	3.015	23.4	0
2-1-3	3.018	22.6	0.25
2-3-1	3.015	22.6	0
3-1-2	3.015	23.4	0
3-2-1	3.018	23.0	0

Interestingly, the average objective value for each priority level in each variation was not much different from the original model's value. This indicates that for those particular HSC problems, the objectives in each priority level were not greatly conflicting.

The model variations 1-3-2, 3-1-2, and 3-2-1 performed as expected. Compared to the original model (i.e. 1-2-3), 1-3-2 resulted in a better average objective value for priority level three, at the expense of priority level two. The variations 3-1-2 and 3-2-1 both predictably resulted in a better average objective value for priority level three. For 3-1-2, the average objective value for priority level one remained the same as the original model. Therefore the improvement in priority level three was offset by the deterioration in priority level two. For the variation 3-2-1, the improvement in priority level three's average objective value resulted in worse values for the other two priority levels.

The performance of the model variations 2-1-3 and 2-3-1, where the objective in priority level two was considered first, did not meet expectations. In these variations, the average objective values obtained for priority level two should have been the same or better than the original model's values. However, this was not the case because the model could not find the optimal values for Weeks 1 and 4 after four hours, at which point the solver terminated its search. This may be because the problems' feasible regions are much larger when the objective in priority level two is considered first, resulting in longer computational times to find the optimal solutions.

Appendix C

C.1 Management Questionnaire

Questionnaire (Management)

Please use the following questions as a basis of your discussion. Please answer the following questions to the best of your knowledge. If your require clarification, please do not hesitate to ask the researcher.

DEMOGRAPHIC INFORMATION

Date: ______________________________ (dd/mm/yy)
Work area: ______________________________
Comments: ______________________________

QUESTIONS

1) Have the research team documented the processes thoroughly? Yes ☐, No ☐
Comments: ______________________________

2) Are there any processes that have not been included? Yes ☐, No ☐
If yes, please explain: ______________________________

3) What is your opinion of our process redesign suggestions?
Answer: ______________________________

4) Do you think that there are other areas of process redesign that the research team has not considered? Yes ☐, No ☐
If yes, please explain: ______________________________

5) What are the barriers of implementation of the process redesign suggestions?
Answer: ____
____
____

General Comments: ____
____
____

6) Do you want to receive a summary of the study results? Yes ☐, No ☐

If yes, you can either give us your contact information or contact Dr. Tarek ElMekkawy.

Your contact information:
Name: ______________________________
Address: ______________________________
____
____
____

C.2 Nurse Questionnaire

Questionnaire (Nurse)

Please use the following questions as a basis of your discussion. Please answer the following questions to the best of your knowledge. If your require clarification, please do not hesitate to ask the researcher.

DEMOGRAPHIC INFORMATION

Date: ______________________________ (dd/mm/yy)
Work area: ______________________________
Comments: ______________________________
____

QUESTIONS

1) What are the tasks that you usually do during the day?
Answer:______________________________
____
____

2) What are the other departments with which you are communicating?
Answer:______________________________
____
____

3) How do you communicate with other departments?
Phone: ☐ Email: ☐ Hospital Information System: ☐ In-person: ☐
Fax: ☐ Other: ☐

4) What are the forms that you have to fill?
Answer:______________________________
____

5) Do you think all of the fields are appropriate? Yes ☐, No ☐
Comments: ______________________________
____
____

6) Do you have any comments on improving the information flow?
Answer: ____
____
____

7) What time of the day do you feel busier?
Answer:____
____

8) Do you have any comments on improving the processes?
Answer: ____
____
____

General Comments: ____
____
____

6) Do you want to receive a summary of the study results? Yes ☐, No ☐

If yes, you can either give us your contact information or contact Dr. Tarek ElMekkawy.

Your contact information:
Name: ______________________________
Address: ______________________________
____
____
____

C.3 Physician Questionnaire

Questionnaire (Physician)

Please use the following questions as a basis of your discussion. Please answer the following questions to the best of your knowledge. If your require clarification, please do not hesitate to ask the researcher.

DEMOGRAPHIC INFORMATION

Date: ______________________________ (dd/mm/yy)
Work area: ______________________________
Comments: ______________________________
____

QUESTIONS

1) What are the tasks that you usually do during the day?
Answer:______________________________
____
____

2) What are the other departments with which you are communicating?
Answer:______________________________
____
____

3) How do you communicate with other departments?
Phone: ☐ Email: ☐ Hospital Information System: ☐ In-person: ☐
Fax: ☐ Other: ☐

4) Do you find the equipment that you need available just in time? Yes ☐, No ☐
Comments:______________________________
____
____

5) Do you find your schedule appropriate and smooth? Yes ☐, No ☐
Comments: ______________________________
____
____

6) Do you have any comments on improving the information flow?
Answer: ____

____

____

General Comments: ____

____

____

7) Do you want to receive a summary of the study results? Yes ☐, No ☐

If yes, you can either give us your contact information or contact Dr. Tarek ElMekkawy.

Your contact information:
Name: ________________________________
Address: ________________________________
____
____
____

C.4 Patient Questionnaire

Questionnaire (Patient)

Please use the following questions as a basis of your discussion. Please answer the following questions to the best of your knowledge. If your require clarification, please do not hesitate to ask the researcher.

DEMOGRAPHIC INFORMATION

Date: ______________________________ (dd/mm/yy)
Work area: ______________________________
Comments: ______________________________
____

QUESTIONS

1) Do you have any complaints? Yes ☐, No ☐
Comments:______________________________
____
____

2) Did you have any unpleasant experiences? Yes ☐, No ☐
Comments:______________________________
____
____

3) Do you think you went through any unnecessary procedures? For example, answering the same questions twice? Yes ☐, No ☐
Comments:______________________________
____
____

4) What do you think can be improved to make your experience better?
Answer:______________________________
____
____

5) Since how long was your appointment booked? Answer:____________

6) Did you have any previous appointments that were cancelled / rescheduled?

Yes ☐, No ☐

Comments: ____

____

____

General Comments: ____

____

____

7) Do you want to receive a summary of the study results? Yes ☐, No ☐

If yes, you can either give us your contact information or contact Dr. Tarek ElMekkawy.

Your contact information:

Name: ______________________________

Address: ______________________________

____

____

____

C.5 Consent Form – Staff

Redesign of Surgery Patients Flow at Health Sciences Centre

Dr. Tarek ElMekkawy
Department of Mechanical and Manufacturing Engineering
University of Manitoba

Sponsored by Winnipeg Regional Health Authority (WRHA)

The consent form will be read to the HSC staff member by the research student. This consent form, a copy of which will be left with you for your records and reference, is only part of the process of informed consent. It should give you the basic idea of what the research is about and what your participation will involve. If you would like more detail about something mentioned here, or information not included here, you should feel free to ask. Please take the time to read this carefully and to understand any accompanying information. This document is also available in alternative formats for your convenience.

GOAL:
The objective of this research is to redesign the flow of surgery patients at Health Sciences Centre (HSC) and analyze the current practice to recommend changes that will enhance the whole system performance. The project will include the entire patient journey from pre-operative care to discharge and post-operative follow up.

In order to redesign the flow of surgery patients at HSC, the researchers first need to understand all the system processes related to this flow. In order to fully comprehend these processes, the researchers must observe and interview the people that carry out these processes. Through careful analysis, it is hoped that it will lead to suggested improvements of the current system. These improvements could be reflected in shorter waiting times and better staff and patient satisfaction.

All information collected during the course of this research will be kept confidential in the principal investigator's office for one year staring from the end of the project then all the collected information will be destroyed. No Names will be mentioned in the results of the research. All results of this study will be made public through WRHA.

Your signature on this form indicates that you have understood to your satisfaction the information regarding participation in the research project and agree to participate in interviews and be observed. In no way does this waive your legal rights nor release the researchers, sponsors, or involved institutions from their legal and professional responsibilities. Interviews will take approximately 20 minutes. You are free to withdraw from the study at any time, and /or refrain from answering any question you prefer to omit, without prejudice or consequence. Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your participation. If you have any questions or would like to receive a summary of results when the study is completed, contact Dr. Tarek ElMekkawy.

This research has been approved by the [REB:]. If you have any concerns or complaints about this project you may contact any of the above-named persons or the Human Ethics Secretariat. A copy of this consent form has been given to you to keep for your records and reference.

Participant's Name   Participant's Signature   Date (dd/mm/yy)

Researcher and/or Delegate's Signature   Date (dd/mm/yy)

Researcher and/or Delegate's Signature   Date (dd/mm/yy)

C.6 Consent Form – Patients

Redesign of Surgery Patients Flow at Health Sciences Centre

Dr. Tarek ElMekkawy
Department of Mechanical and Manufacturing Engineering
University of Manitoba

Sponsored by Winnipeg Regional Health Authority (WRHA)

The consent form will be read to the patient by the research student. This consent form, a copy of which will be left with you for your records and reference, is only part of the process of informed consent. It should give you the basic idea of what the research is about and what your participation will involve. If you would like more detail about something mentioned here, or information not included here, you should feel free to ask. Please take the time to read this carefully and to understand any accompanying information. This document is also available in alternative formats for your convenience.

GOAL:
The objective of this research is to redesign the flow of surgery patients at Health Sciences Centre (HSC) and analyze the current practice to recommend changes that will enhance the whole system performance. The project will include the entire patient journey from pre-operative care to discharge and post-operative follow up.

A part of this research project is to obtain the feedback from some patients regarding the received services. The feedback will highlight some points of weakness or strength of the current system. Through careful analysis of the patient feedbacks, it is hoped that will lead to suggested improvements of the current system. The improvement could be reflected in a shorter waiting time and better patient's satisfaction.

All information collected during the course of this research will be kept confidential in the principal investigator's office for one year staring from the end of the project then all the collected information will be destroyed. No Names will be mentioned in the results of the research. All results of this study will be made public through WRHA.

Your signature on this form indicates that you have understood to your satisfaction the information regarding participation in the research project and agree to participate as a patient. In no way does this waive your legal rights nor release the researchers, sponsors, or involved institutions from their legal and professional responsibilities. You are free to withdraw from the study at any time, and /or refrain from answering any question you prefer to omit, without prejudice or consequence. Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your participation (Dr. Tarek ElMekkawy)

This research has been approved by the [REB:]. If you have any concerns or complaints about this project you may contact any of the above-named persons or the Human Ethics Secretariat. A copy of this consent form has been given to you to keep for your records and reference.

Participant's Name	Participant's Signature	Date (dd/mm/yy)

Researcher and/or Delegate's Signature	Date (dd/mm/yy)

Researcher and/or Delegate's Signature	Date (dd/mm/yy)

References

Bagchi, T.P., *Multiobjective Scheduling by Genetic Algorithms*, Boston: Kluwer Academic Publishers, 1999.

Belien, J. and Demeulemeester, E., “Building cyclic master surgery schedules with leveled resulting bed occupancy,” *European Journal of Operational Research*, vol. 176, no. 2, pp.1185–1204, 2007.

Benayoun, R., de Montgolfier, J., Tergny, J. and Laritchev, P., “Linear programming with multiple objective function: Step method (STEM),” *Mathematical Programming*, vol. 1, no. 3, pp. 366–375, 1971.

Blake, J.T. and Carter, M.W., “A goal programming approach to strategic resource allocation in acute care hospitals,” *European Journal of Operational Research*, vol. 140, no. 3, pp. 541–561, 2002.

Blake, J.T., Dexter, F. and Donald, J., “Operating room manager’s use of integer programming for assigning block time to surgical groups: a case study,” *Anesthesia & Analgesia*, vol. 94, no. 1, pp. 143–148, 2002.

Blake, J.T. and Donald, J., “ Mount Sinai Hospital uses integer programming to allocate operating room time,” *Interfaces*, vol. 32, no. 2, pp.63–73, 2002.

Brideau, L.P., “Flow: why does it matter?,” *Frontiers of Health Services Management*, vol. 20, no. 4, pp. 47–50, 2204.

Broka, S.M., Jamart, J. and Louagie, Y.A.G., “Scheduling of elective surgical cases within allocated block-times: can the future be drawn from the experience of the past?,” *Acta Chir Belg*, vol. 103, no. 1, pp. 90–94, 2003.

Buchanan, J.T., “Multiple objective mathematical programming: a review,” *New Zealand Operational Research*, vol. 14, no. 1–27, pp. 1, 1986.

Buchanan, J.T., “A naïve approach for solving MCDM problems: the GUESS method,” *Journal of the Operational Research Society*, vol. 48, no. 2, pp. 202–206, 1997.

Calichman, M.V., “Creating an optimal operating room schedule,” *Association of Operating Room Nurses (AORN) Journal*, vol. 81, no. 3, pp. 580–588, 2005.

Cavicchio, D.J., “Adaptive Search Using Simulated Evolution.” Ph.D. thesis, University of Michigan, 1970.

Chaabane, S., Meskens, N., Guinet, A. and Laurent, M., "Comparison of two methods of operating theatre planning: application in Belgian hospital," *2006 International Conference on Service Systems and Service Management*, Troyes, France, pp. 83–90, Oct. 2006.

Changkong, V. and Haimes, Y.Y., "Optimization-based methods for multiobjective decision-making: an overview," *Large Scale Systems*, vol. 5, no. 1, pp. 1–33, 1983a.

Changkong, V. and Haimes, Y.Y., *Multiobjective Decision Making Theory and Methodology*, New York: Elsevier Science Publishing Co. Inc., 1983b.

Charnes, A., Cooper, W. and Ferguson, R., "Optimal estimation of executive compensation by linear programming," *Management Science*, vol. 1, no. 2, pp. 138–151, 1955.

Chipperfield, A. J. and Fleming, P. J., "Gas turbine engine controller design using multiobjective genetic algorithms," First IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, University of Sheffield, UK, pp. 214–219, Sep. 1995.

Cieniawski, S.E., Eheart, J. W. and Ranjithan, S., "Using genetic algorithms to solve a multiobjective groundwater monitoring problem," *Water Resources Research*, vol. 31, no. 2, pp. 399–409, 1995.

Clayton, E.R., Weber, W.E. and Taylor, B.W., "A goal programming approach to the optimization of multi-response simulation models," IIE Transactions, vol. 14, no. 4, pp. 282–287, 1982.

Coello, C.A.C., "A comprehensive survey of evolutionary-based multiobjective optimization techniques," *International Journal of Knowledge and Information Systems*, vol. 1, no. 3, pp. 269–308, 1999.

Coello, C.A.C., Aguirre, A.H. and Buckles, B.P., "Evolutionary multiobjective design of combinational logic circuits," Second NASA/DoD Workshop on Evolvable Hardware, Los Alamitos, California, pp. 161-170, Jul. 2000.

Coello, C.A.C., "A short tutorial on evolutionary multiobjective optimization," *First International Conference on Evolutionary Multi-Criterion Optimization*, Zurich, Switzerland, pp. 21-40, Mar. 2001.

Coello, C.A.C. and Christiansen, A.D., "MOSES: a multiobjective optimization tool for rngineering design," *Engineering Optimization*, vol. 31, no. 3, pp. 337–368, 1999.

Coello, C.A.C., "Recent trends in evolutionary multiobjective optimization," *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, London: Springer-Verlag, 2005, pp. 7–22.

Cohon, J.L., *Multiobjective Programming and Planning*, New York: Academic Press Inc., 1978.

Cohon, J.L., *Multicriteria Programming: Brief Review and Application, Design Optimization*, New York: Academic Press, 1985.

Collette, Y., and Siarry, P., *Multiobjective Optimization: Principles and Case Studies*, New York: Springer-Verlag, 2003.

Das, I. and Dennis, J.E., "A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems," *Structural Optimization*, vol. 14, no. 1, pp. 63–69, 1997.

Das, I. and Dennis, J.E., "Normal-boundary intersection: a new method for generating the pareto surface in nonlinear multicriteria optimization problems," *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 631–657, 1998.

Datta, D., Deb. K. and Fonseca, F., "Multi-objective evolutionary algorithm for university class timetabling problem," *Evolutionary Scheduling*, Berlin: Springer, 2007, pp. 197–236.

Davidor, Y., "A naturally occurring niche and species phenomenon: the model and first results," *Fourth International Conference on Genetic Algorithms*, San Diego, California, pp. 257–263, Jul. 1991.

Deb, K., *Multiobjective Optimization using Evolutionary Algorithms*, Chichester: John Wiley & Sons, 2001.

Deb, K., Agrawal, S., Pratab, A. and Meyarivan, T., "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," KanGAL report 200001, Indian Institute of Technology, Kanpur, India, 2000a.

Deb, K., Agrawal, S., Pratab, A., and Meyarivan, T., "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," *Sixth International Conference on Parallel Problem Solving from Nature*, Paris, France, pp. 849–858, Sep. 2000b.

Deb, K., and Goldberg, D.E., "An investigation of niche and species formation in genetic function optimization," *Third International Conference on Genetic Algorithms*, San Mateo, California, pp. 42–50, Jun. 1989.

DeJong, K.A., "An Analysis of the Behaviour of a Class of Genetic Adaptive Systems." Ph.D. thesis, University of Michigan, 1975.

Denton, B., Viapiano, J. and Vogl, A., “Optimization of surgery sequencing and scheduling decisions under uncertainty,” *Health Care Management Science*, vol. 10, no. 1, pp. 13–24, 2007.

Dexter, F. and Epstein, R.H., “Scheduling of cases in an ambulatory center,” *Anesthesiology Clinics of North America*, vol. 21, no. 2, pp. 387–402, 2003.

Dexter, F., Epstein, R.H. and Marsh, H.M., “A statistical analysis of weekday operating room anesthesia group staffing costs at nine independently managed surgical suites,” *Anesthesia & Analgesia*, vol. 92, no. 6, pp. 1493–1498, 2001.

Dexter, F., Macario, A., Lubarsky, D.A. and Burns, D.D., “Statistical method to evaluate management strategies to decrease variability in operating room utilization: Application of linear statistical modeling and Monte Carlo simulation to operating room management,” *Anesthesiology*, vol. 91, no, 1, pp. 262–274, 1999a.

Dexter, F., Macario, A., Qian, F. and Traub, R.D., “Forecasting surgical groups’ total hours of elective cases for allocation of block time: Application of time series analysis to operating room management,” *Anesthesiology*, vol. 91, no. 5, pp. 1501–1508, 1999b.

Dexter, F., Macario, A., Traub, R.D., Hopwood, M. and Lubarsky, D.A., “An operating room scheduling strategy to maximize the use of operating room block time: Computer simulation of patient scheduling and survey of patients’ preferences for surgical waiting time”. *Anesthesia & Analgesia*, vol. 89, no. 1, pp. 7–20, 1999c.

Dexter, F., Macario, A. and O’Neill, L., “Scheduling surgical cases into overflow block time – computer simulation of the effects of scheduling strategies on operating room labor costs,” *Anesthesia & Analgesia*, vol. 90, no. 4, pp. 980–988, 2000.

Dexter, F. and Traub, R.D., “Determining staffing requirements for a second shift of anesthetists by graphical analysis of data from operating room information systems,” *Journal of the American Association of Nurse Anesthetists (AANA)*, vol. 68, no. 1, pp. 31–36, 2000a.

Dexter, F. and Traub, R.D., “Sequencing cases in operating rooms: predicting whether one surgical case will last longer than another,” *Anesthesia & Analgesia*, vol. 90, no. 4, pp. 975–979, 2000b.

Dexter, F. and Traub, R.D., “How to schedule elective surgical cases into specific operating rooms to maximize the efficiency of use of operating room time,” *Anesthesia & Analgesia*, vol. 94, no. 4, pp. 933–942, 2002.

Diwekar, U.M., *Introduction to Applied Optimization*, Norwell: Kluwer Academic Publishers, 2003.

Ehrgott, M., *Multicriteria Optimization,* Berlin: Springer-Verlag, 2000.

Emmanouilidis, C., Hunter, A., and MacIntyre, J., “A multiobjective evolutionary setting for feature selection and a commonality-based crossover operator,” *2000 Congress on Evolutionary Computation*, Piscataway, New Jersey, pp. 309–316, Jul. 2000.

Eschenauer, H., Koski, J., and Osyczka, A., *Multicriteria design optimization: procedure and applications*, New York: Springer-Verlag, 1990.

Fei, H., Meskens, N. and Chu, C. “An operating theatre planning and scheduling problem in the case of a “block scheduling” strategy,” *2006 International Conference on Service Systems and Service Management*, Troyes, France, Oct. 2006.

Fonseca, C.M. and Fleming, P.J., “Multiobjective optimization and multiple constraint handling with evolutionary algorithms – Part I: a unified formulation,” *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 28, no. 1, pp. 26–37, 1998.

Fonseca, C.M., and Fleming, P.J., “Genetic algorithms for multiobjective optimization: formulation, discussion and generalization,” *Fifth International Conference on Genetic Algorithms*, San Mateo, California, pp. 416–423, Jul. 1993

Fonseca, C.M. and Fleming, P.J., “An overview of evolutionary algorithms in multiobjective optimization,” *Evolutionary Computation*, vol. 3, no. 1, pp. 1–16, 1995.

Gembicki, F.W. and Haimes, Y.Y., “Approach to performance and sensitivity multiobjective optimization: the goal attainment method,” *IEEE Transactions on Automatic Control*, vol. 20, no. 6, pp. 769–771, 1975.

Goldberg, D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Reading: Addison-Wesley Publishing Company, 1989.

Goldberg, D.E., and Richardson, J., “Genetic algorithms with sharing for multimodal function optimization,” *Second International Conference on Genetic Algorithms*, Cambridge, Massachusetts, pp. 41–49, Jul. 1987.

Goldberg, D.E. and Wang, L., 1998, “Adaptive niching via coevolutionary sharing,” *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science*, Chichester: Wiley. pp. 21–38.

Grignon, P., Wodziack, J. and Fadel, G.M., “Bi-Objective optimization of components packing using a genetic algorithm,” *NASA/AIAA/ISSMO Multidisciplinary Design and Optimization Conference*, Seattle, Washington, pp. 352–362, Sep. 1996.

Guinet, A. and Chaabane, S., “Operating theatre planning,” *International Journal of Production Economics*, vol. 85, no. 1, pp. 69–81, 2003.

Haimes, Y.Y., Lasdon, L.S. and Wismer, D.A., "On a bicriterion formulation of the problems of the integrated identification and system optimization, *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 1, no. 3, pp. 296–297, 1971.

Hajela, P. and Lin, C.Y., "Genetic algorithms in multi-criterion optimal design," *Structural Optimization*, vol. 4, no. 2, pp. 99–107, 1993.

Haraden, C. and Resar, R., "Patient flow in hospitals: understanding and controlling it better," *Frontiers of Health Services Management*, vol. 20, no. 4, pp. 3–15, 2004.

Henderson, D., Dempsey, C. and Appleby, D., "A case study of successful patient flow methods: St. John's Hospital," *Frontiers of Health Services Management*, vol. 20, no. 4, pp. 25–30, 2004.

Holland, J.H., *Adaptation in Natural and Artificial Systems*, Ann Arbor: The MIT Press, 1975.

Horn, J., Nafpliotis, N. and Goldberg, D.E., "Multiobjective optimization using niched pareto genetic algorithm," Technical report, IlliGAl Report 93005, Illinois University, Urbana, Illinois, 1993.

Horn, J., Nafpliotis, N. and Goldberg, D.E., "A niched pareto genetic algorithm for multiobjective optimization," *First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, Orlando, Florida, pp. 82–87, Jun. 1994.

Horton, S.S., "Increasing capacity while improving the bottom line," *Frontiers of Health Services Management*, vol. 20, no. 4, pp. 17–23, 2004.

Hwang, C.L. and Masud, A.S.M., "Multiple Objective Decision Making – Methods and Applications: A State-of-the-Art Survey," Berlin: Springer-Verlag, 1979.

Ignizio, J.P., *Goal Programming and Extensions*. Lexington: Lexington Books, 1976.

Ignizio, J.P., "A review of goal programming: A tool for multiobjective analysis," *Journal of Operations Research Society*, vol. 29, no. 11, pp. 1109–1119, 1978.

Institute for Healthcare Improvement (IHI), "Optimizing Patient Flow: Moving Patients Smoothly Through Acute Care Settings," *IHI Innovation Series white paper*, 2003. [Online]. Available: http://www.ihi.org. [Accessed: Aug. 13, 2007].

Jebali, A., Hadj Alouane, A.B. and Ladet, P., "Operating rooms scheduling," *International Journal of Production Economics*, vol. 99, no. 1–2, pp. 52–62, 2006.

Jiménez, F., Verdegay, J.L. and Goméz-Skarmeta, A.F., “Evolutionary techniques for constrained multiobjective optimization problems,” *1999 Genetic and Evolutionary Computation Conference*, Orlando, Florida, pp. 115–116, Jul. 1999.

Jones, D.F., Mirrazavi, S.K. and Tamiz, M., “Multi-objective meta-heuristics: an overview of the current state-of-the-art,” *European Journal of Operational Research*, 137, no. 1, pp. 1–9, 2002.

Keeney, R.L. and Raiffa, H., *Decisions with multiple objectives: preferences and value tradeoffs*, 1st ed. Cambridge: Cambridge University Press, 1976.

Keeney, R.L. and Raiffa, H., *Decisions with multiple objectives: preferences and value tradeoffs*, 2nd ed., Cambridge: Cambridge University Press, 1993.

Khor, E. F., Lee, T. H. and Tan, K. C., *Multiobjective Evolutionary Algorithms and Applications: Advanced Information and Knowledge Processing*, London: Springer, 2005. [E-book] Available: SpringerLink.

Knowles, J.D. and Corne, D.W., “Approximating the nondominated front using the pareto archived evolution strategy,” *Evolutionary Computation*, vol. 8, no. 2, pp. 149–172, 2000.

Kuhn, H. W. and Tucker, A. W., “Nonlinear Programming,” *Second Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, California, pp. 481–492, Jul. 1950.

Kuo, P.C., Schroeder, R.A., Mahaffey, S. and Bollinger, R.R., “Optimization of Operating Room Allocation Using Linear Programming Techniques,” *Journal of the American College of Surgeons*, vol. 197, no. 6, pp. 889–895, 2003.

Lambert, M., “Improvement and innovation in hospital operations: a key to organizational health,” *Frontiers of Health Services Management*, vol. 20, no. 4, pp. 39–45, 2004.

Laumanns, M., Rudolph, G. and Schwefel, H.P., “A spatial predator-prey approach to multi-objective optimization: a preliminary study,” *Fifth Internal Conference of Parallel Problem Solving from Nature*, Amsterdam, Netherlands, pp. 241–249, Sep. 1998.

Lebowitz, P., “Schedule the Short Procedure First to Improve OR Efficiency,” *Association of Operating Room Nurses (AORN) Journal*, vol. 78, no. 4, pp. 651–659, 2003.

Lichtfuss, H.K., “Evolution eines Rohrkrümmers,” Diplomarbeit, Technische Universität Berlin, Deutschland, 1965.

Lieberman, E.R., “Soviet multi-objective mathematical programming methods: an overview,” *Management Science*, vol. 7, no. 9, pp. 1147–1165, 1991.

Marcon, E., Kharraja, S. and Simonnet, G., “Minimization of the risk of no realization for the planning of the surgical interventions into the operating theatre,” Eighth *IEEE* International *Conference on Emerging Technologies and Factory Automation*, Antibes-Juan les Pins, France, pp. 675–680, Oct. 2001.

Marcon, E., Kharraja, S. and Simonnet, G., “The operating theatre planning by the follow-up of the risk of no realization,” *International Journal of Production Economics*, vol. 85, no. 1, pp. 83–90, 2003.

Marcu, T., “A multiobjective evolutionary approach to pattern recognition for robust diagnosis of process faults,” *IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes: SAFEPROCESS'97*, Kingston Upon Hull, United Kingdom, pp. 1183–1188, Aug. 1997.

Miettinen, K.M., *Nonlinear Multiobjective Optimization*, Boston: Kluwer Academic Publishers, 1999.

Montana, D., Brinn, M., Moore, S. and Bidwell, G., “Genetic Algorithms for Complex, Real-Time Scheduling,” *1998 IEEE International Conference on Systems, Man, and Cybernetics*, La Jolla, California, pp. 2213–2218, Oct. 1998.

Mulholland, M.W., Abrahamse, P. and Bahl, V, “Linear programming to optimize performance in a department of surgery,” *Journal of the American College of Surgeons*, vol. 200, no. 6, pp. 861–868, 2005.

Murata, T. and Ishibuchi, H., “MOGA: multi-objective genetic algorithms,” *1995 IEEE International Conference on Evolutionary Computation*, Perth, Australia, pp. 289–294, Nov. 1995.

Nelder, J.A., Mead, R., “A simplex method for function minimization,” *Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965.

Oei, C.K., Goldberg, D.E., and Chang, S. –J., “Tournament selection, niching, and the preservation of diversity,” IlliGAL Report No. 91011, Urbana, IL: University of Illinois at Urbanada-Champaign, 1991.

Ogulata, S.N. and Erol, R., “A hierarchical multiple criteria mathematical programming approach for scheduling general surgery operations in large hospitals,” *Journal of Medical Systems*, vol. 27, no. 3, pp. 259–270, 2003.

OR Manager, “How can ORs best manage block time for scheduling surgical cases?,” *OR Manager*, vol. 19, no. 2, pp. 1, 21–24, 2003.

Osyczka, A., “Multicriteria optimization for engineering design,” *Design Optimization*, New York: Academic Press, 1985, pp. 193–227.

Ozkarahan, I., ”Allocation of surgeries to operating rooms by goal programming,” *Journal of Medical Systems*, vol. 24, no. 6, pp. 339–378, 2000.

Pareto V., *Cours d’Economie Politique*, Libraire Droz, Genéve, 1964.

Pareto V., *Manual of Political Economy*, New York: The MacMillan Press Ltd, 1971.

Patterson, P., “What makes a well-oiled scheduling system?. *OR Manager*, vol. 12 no. 9, pp. 19–23, 1996.

Patterson, P., “A few simple rules for managing block time in the operating room,” *OR Manager*, vol. 20, no. 11, 1, pp. 9–10, 12, 2004.

Perdomo, V., Augusto, V. and Xie, X, “Operating theatre scheduling using lagrangian relaxation,” *2006 International Conference on Service Systems and Service Management*, Troyes, France, Oct. 2006.

Ray, T., Tai, K. and Seow, K.C., “An evolutionary algorithm for multiobjective optimization,” *Engineering Optimization*, vol. 33, no. 3, pp. 399–424, 2001.

Rechenberg, I., “Cybernetic solution path of an experimental problem,” Royal Aircraft Establishment, Library Translation Number 1122, Farnborough, UK, 1965.

Rohleder, T.R., Sabapathy, D. and Schorn, R., “An Operating Room Block Allocation Model to Improve Hospital Patient Flow,” *Clinical and Investigative Medicine*, vol. 28, no. 6, pp. 353–355, 2005.

Roland, B., Martinelly, C.D. and Riane, F, “Operating theatre optimization: a resource-constrained based solving approach,” *2006 International Conference on Service Systems and Service Management*, Troyes, France, Oct. 2006.

Romero, C., *Handbook of Critical Issues in Goal Programming*, Oxford: Pergamon Press, 1991.

Sandgren, E., “Multicriteria design optimization by goal programming,” *Advances in Design Optimization*, London: Chapman & Hall, London, 1994, pp. 225–265.

Sayyouth, M.H., “Goal programming: a new tool for optimization in petroleum reservoir history matching,” *Applied Mathematics Modelling*, vol. 5, no. 4, pp. 223–226, 1981.

Schaffer, J.D., “Multiple Objective Optimization with Vector Evaluated Genetic Algorithms,” *First International Conference on Genetic Algorithms*, Pittsburgh, Pennsylvania, pp. 93–100, Jul. 1985.

Schwefel, H.-P., “Projekt MHD-Staustrahlrohr: Experimentelle optimierung einer zweiphasendüse, Teil I,” Technical report, 11.034/68, 35, AEG Forschungsinstitut, Berlin, 1968.

Sier, D., Tobin, P. and McGurk, C., “Scheduling surgical procedures,” *Journal of the Operational Research Society*, vol. 48, no. 9, pp. 884–891, 1997.

Srinivas, N. and Deb, K., “Multiobjective optimization using non-dominated sorting in genetic algorithms,” Technical report, Department of Technical Engineering, Indian Institute of Technology, Kanpur, India, 1993.

Srinivas, N. and Deb, K, “Multiobjective optimization using nondominated sorting in genetic algorithms,” *Evolutionary Computation Journal*, vol. 2, no. 3, pp. 221–248, 1994.

Stadler, W. (Ed.), *Multicriteria Optimization in Engineering and in the Sciences*, New York: Plenum Press, 1988.

Steuer, R.E., *Multiple Criteria Optimization: Theory, Computation and Application*, New York: Wiley, 1986.

Strum, D. P., May, J.H. and Vargas, L.G., “Surgical procedure times are well modeled by the lognormal distribution,” *Anesthesia & Analgesia*, vol. 86, 47S, 1998.

Strum, D. P., Vargas, L.G. and May, J.H., “Surgical subspecialty block utilization and capacity planning: A minimal cost analysis model,” *Anesthesiology*, vol. 90, no. 4, pp. 1176–1185, 1999.

Testi, A., Tanfani, E. and Torre, G., “A three-phase approach for operating theatre schedules,” *Health Care Management Science*, vol. 10, no. 2, pp. 163–172, 2007.

Valenzuela, C.L., “A simple evolutionary algorithm for multi-objective optimization (SEAMO),” *Congress on Evolutionary Computation (CEC)*, Honolulu, Hawaii, pp. 717–722, May 2002.

Valenzuela, C.L., “A Simple Approach to Evolutionary Multiobjective Optimization,” *Evolutionary Multiobjective Optimization*, London: Springer-Verlag, 2005, pp. 55–79.

van Oostrum, J.M., Van Houdenhoven, M., Hurink, J.L., Hans, E.W., Wullink, G. and Kazemier, G., “A master surgical scheduling approach for cyclic scheduling in operating room departments,” *OR Spectrum*, vol. 30, no. 2, pp. 355-374, 2006.

Van Veldhuizen, D.A., “Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations.” Ph.D. thesis, Air Force Institute of Technology, 1999.

Velásquez, R., and Melo, M.T., "A Set Packing Approach for Scheduling Surgical Procedures," *Operations Research Proceedings 2005*, New York: Springer-Verlag, pp. 425–430.

Wierzbicki, A.P., "The use of reference objectives in multiobjective optimization," *Multiple Criteria Decision Making Theory and Applications*, Berlin: Springer-Verlag, 1980, pp. 468–486.

Wierzbicki, A.P., "A mathematical basis for satisficing decision making," *Mathematical Modeling*, vol. 3, no. 5, pp. 391–405, 1982.

Wilson, P. B. and Macleod, M. D, "Low implementation cost IIR digital filter design using genetic algorithms," *Second IEE/IEEE Workshop on Natural Algorithms in Signal Processing*, Chelmsford, U.K., pp. 4/1–4/8, Nov. 1993.

Wright, H.I., Kooperberg, C., Bonar, B.A. and Bashein, G., "Statistical modeling to predict elective surgery time. Comparison with a computer scheduling system and surgeon-procided estimates. *Anesthesiology*, vol. 85, no. 6, pp. 1235–1245, 1996.

Zeleny, M., *Multiple Criteria Decision Making*, New York: McGraw-Hill Inc., 1982.

Zhou, J. and Dexter, F., "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*, vol. 89, no. 5, pp. 1228–1232, 1998.

Zitzler, E. and Thiele, L., "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.