Dynamic Optimization Addressing Chemotherapy Outpatient Scheduling

by

Shoshana Hahn-Goldberg

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy

Mechanical and Industrial Engineering
University of Toronto

Dynamic Template Scheduling Addressing Chemotherapy Outpatient Scheduling

Shoshana Hahn-Goldberg

Doctor of Philosophy

Mechanical and Industrial Engineering
University of Toronto

2014

Abstract

Chemotherapy outpatient scheduling is a complex problem containing uncertainty. Chemotherapy centres are facing increasing demands and they need to increase their efficiency. However, there are very few studies looking at using optimization methods on the chemotherapy scheduling problem.

In this dissertation, the chemotherapy outpatient scheduling problem is defined within the scheduling literature. Next, we propose a methodology for choosing what information to include from the problem domain when creating a mathematical model of a real world problem. Several constraint programming models, representing different problem definitions of the deterministic chemotherapy scheduling problem, are created and evaluated for their solvability and the quality of their solutions. The chosen optimization model was tested within a dynamic framework in order to accommodate the dynamism and uncertainty inherent in chemotherapy outpatient scheduling. Termed dynamic template scheduling, this novel algorithm uses the chemotherapy centre's past records and the chosen model to create a template of open slots. As requests for appointments arrive, we use the template to schedule them. When a request arrives that does not fit the template, we update the template. To accommodate last minute additions and cancellations to the schedule, we test a shifting algorithm that moves patient start times within a predefined time limit.

We demonstrate that chemotherapy centres can use records of past appointments to inform future schedules and that integrating optimization methods into the scheduling procedures can improve

efficiency and increase throughput. This research makes a contribution to scheduling research by developing a novel technique that combines proactive and reactive scheduling to address dynamic problems with real-time uncertainty. This research also makes a contribution to health care scheduling applications by solving a case of chemotherapy outpatient scheduling, a practically important problem that has had very little treatment in the literature.

Acknowledgments

Mitacs and Bykart software for funding support.

Professors Michael Carter and Chris Beck for their valuable supervision.

Professor Daniel Frances and Dr. Maureen Trudeau for their support and feedback throughout.

Philomena Sousa, Kathy Beattie, and Thane Fitzgerald for their advice and information.

Professors Dionne Aleman and Elizabeth Jewkes for taking the time to act as appraisers.

Wen-Yang Ku for his help with the MIP modeling.

The Centre for Innovation in Complex Care for allowing me to work and do this at the same time.

My family for an incredible amount of help and support.

Moishie Goldberg for support and a large amount of assistance with the programming involved in this work.

Table of Contents

Contents

List of Tables

List of Figures

List of Algorithms

List of Appendices

Chapter 1

[redacted] Introduction

The central thesis of this dissertation is that the chemotherapy outpatient scheduling problem can be solved using mathematical modeling in the form of constraint programming combined with a novel, dynamic optimization technique. The chemotherapy outpatient scheduling problem is a complex problem with complicated resource constraints and several sources of uncertainty. This work applies optimization, through constraint programming and a dynamic scheduling technique, to the chemotherapy outpatient scheduling problem at the Odette Cancer Centre at Sunnybrook Health Sciences Centre. The solution technique is then applied to a second site, the Juravinski Cancer Centre, to examine the general applicability of the approach to chemotherapy outpatient scheduling problems.

This dissertation makes a contribution to health care scheduling applications by defining and solving the chemotherapy outpatient scheduling problem, a relevant problem that has not had much exposure to optimization. This dissertation also makes a contribution to scheduling research by developing a novel technique to address dynamic problems with real-time uncertainty and uncertainty due to last minute schedule changes. To address real-time uncertainty, we use historical data from the cancer centre and a constraint programming model that optimally solves the deterministic chemotherapy outpatient scheduling problem to produce a template that is updated dynamically as new information about the problem becomes known. To address uncertainty due to last minute changes, we incorporate a shifting algorithm to move the start times of appointments by a limited amount of time. The detailed contributions of this dissertation are summarized in Section 1.4.

1.1 Background

Requests for chemotherapy appointments for a given day at the cancer centre arrive over the course of several weeks. After a patient is seen in the clinic for the first time, the oncologist will specify the day that the patient should begin treatment and the patient's regimen (a particular set of drugs, dosages, methods of administration, and timetable for administration). At that time, the first set of appointments for that patient is scheduled and the patient is given an appointment time. After each block of treatment, the patient will be seen by the oncologist and their next block of treatment will be scheduled. Appointments for a particular day may be scheduled from several weeks to about one week in advance. However, there are emergency appointments added and appointments are cancelled up to and including the day of treatment.

On the day of treatment, patients arrive at the chemotherapy centre and follow a four stage process: (1) chart review, where they are approved for treatment; (2) pharmacy or drug preparation, where their drugs are prepared; (3) the first stage of treatment or setting up the patient, where direct nurse supervision is required; and (4) the second stage of treatment or delivery of drugs, where a nurse can monitor multiple patients. There is one nurse available who only completes the chart reviews, several pharmacy technicians available to prepare drugs, several nurses available for the treatment stages, and many chairs that are required during both treatment stages.

There are two main sources of uncertainty. First, there is real-time uncertainty, where appointments must be booked in real-time. Second, there is uncertainty due to last minute additions and cancellations to the schedule. The dynamism caused by the two sources of uncertainty results in a schedule where chairs and nurses may not be utilized efficiently. In the current practice at the clinic, appointment times are not changed after they have been provided to the patient regardless of the need to fit in last minute appointments or adjust for cancellations. The scheduler may try to move patients between chairs in order to accommodate additional appointments, but due to the time required to make such manual changes, they are not frequently attempted.

The complexity of the chemotherapy outpatient scheduling problem makes it a very interesting problem from the perspective of optimization research and there are aspects of the problem that present a challenge to traditional optimization techniques. First, there is large variability among treatments. The Odette Cancer Centre administers over 350 regimens. Treatment durations can range from 15 minutes to 8 hours for a single treatment. Combined with the dynamic arrival of appointment requests, high variability in appointment durations makes it difficult to produce an efficient schedule a priori. Second, the chemotherapy centre must provide patients with sufficient notice of their appointment time, meaning that in the current practice at the clinic, appointment times cannot be changed. Last, the chemotherapy problem at the Odette Cancer Centre is large, with around 100 patient appointments each day. Optimization techniques that take into account some of the other complex features of the chemotherapy problem are generally tested on smaller problem instances (Mehta and Uzsoy 1998, Lambrechts 2008, Adan et al/2008). Overall, the combination of the above factors makes the chemotherapy outpatient scheduling problem unique, interesting, and challenging.

The chemotherapy outpatient scheduling problem can be described as a four-stage flexible flow shop problem with the objectives of maximizing resource utilization and minimizing patient wait times. The chemotherapy application is characterized by appointment requests that are scheduled in real-time, highly variable treatment times, nurses that can be shared among patients, and an environment that is highly uncertain. Appointments are cancelled, added, and changed on a regular basis. By developing a technique to solve the chemotherapy outpatient scheduling problem, the present research makes a unique contribution to the literature on outpatient scheduling applications.

1.2 Objectives

There is considerable potential for optimization in the chemotherapy outpatient scheduling problem. The Odette Cancer Centre has developed a sophisticated booking tool, CHARM, which primarily allows staff to keep track of appointments. CHARM is able to provide the user with the earliest available appointment time for a given request based on the treatment length. When the user books the appointment, CHARM automatically books the subsequent appointments required for that regimen up to one year in advance. At the Odette Cancer Centre, as in many cancer centres, although each appointment requires a chart review nurse, a pharmacy technician, a treatment nurse, and a chair, the appointments are booked by only taking into account a single resource: the chair.

The objective of this research is to develop a more sophisticated, automated approach to scheduling to increase the efficiency of chemotherapy service delivery so that more patients can be serviced in a timely fashion.

1.3 Plan of Dissertation

This dissertation is organized as follows:

Chapter 2 is a literature review. It provides an introduction to the broader research in scheduling. Classic scheduling models, complex scheduling characteristics, and scheduling applications that are relevant to the chemotherapy problem are reviewed to highlight linkages between chemotherapy and different areas of scheduling research. Past research in chemotherapy scheduling is also reviewed and an overview of constraint programming, the optimization technique used in this research, is provided.

Chapter 3 provides a description of the chemotherapy outpatient scheduling application. Details of the Odette Cancer Centre processes and current scheduling practices are given. The chemotherapy outpatient scheduling problem is categorized with standard scheduling concepts and terms.

Chapter 4 describes the first stage of the solution process, creating a problem definition. Formulating a problem definition that can be solved in a reasonable amount of time and produces solutions that are useful in the real world is a critical step in creating a solution technique for any real-world problem. A methodology is proposed to design and evaluate different problem definitions.

Chapter 5 describes dynamic template scheduling, the proposed technique for incorporating the dynamism and uncertainty of the problem domain into the automated scheduling approach. By using the optimization model chosen in Chapter 4 and an approximation of the future, a day's expected appointments are scheduled to create an optimal template of open slots which is then used for booking appointments. When a request for an appointment arrives that does not have a matching slot in the template, the template is dynamically updated. Using data from the chemotherapy centre, statistical analysis is used to create a distribution of appointment lengths that acts as an approximation of the set of appointment requests for a given day. Experiments are conducted to test dynamic template scheduling against the optimal deterministic solution and current practice at the Odette Cancer Centre.

Chapter 6 details the algorithm used to move scheduled appointment times within a pre-determined time interval to accommodate additions and cancellations to a day that has been partially scheduled. Additions and cancellations can occur up to and including the day of treatment and are common at the cancer centre. Currently, to accommodate additions, the schedulers manually move patients from chair to chair, without changing any appointment start times, or they place the addition at the end of the day's schedule. Cancellations are left as empty slots in the schedule. Experiments are conducted to quantify the benefit of shifting, allowing small adjustments to existing appointment start times. Chapter 6 also illustrates a more complete test, a run-through of the dynamic template system, conducted at the Odette Cancer Centre. Using guidelines from the chemotherapy centre and taking into account cancellations, a realistic test of dynamic template scheduling is performed on a single test case. The schedules resulting from the run-through of dynamic template scheduling with shifting is compared to a dry run without shifting and current practice at the cancer centre.

Chapter 7 describes the application of the system at a second chemotherapy site, the Juravinski Cancer Centre in Hamilton, Ontario. Although the patients follow the same process as at the Odette Cancer Centre, the mix of patients and the process of scheduling appointments are different. Dynamic template scheduling was developed using the chemotherapy outpatient scheduling problem at the Odette Cancer Centre. By testing the system at a second chemotherapy site, the flexibility of the technique is demonstrated. A description of the process at the Juravinski Cancer Centre is provided. The process of choosing an optimization model, applying dynamic template scheduling, and shifting is conducted on data from the Juravinski Cancer Centre. Experiments, associated results, and conclusions from this study are presented.

Chapter 8 provides the conclusions and highlights the major contributions of the research. It also provides suggestions for future research.

1.4 Summary of Contributions

This dissertation examines the problem of chemotherapy outpatient scheduling. The contributions of this research are as follows:

- The chemotherapy outpatient scheduling problem is defined within the broader domain of scheduling research. Chemotherapy outpatient scheduling is a large and complex problem that has not had much attention in the scheduling literature and it has never been formally defined as a scheduling problem. By defining the chemotherapy outpatient scheduling problem as a scheduling problem, we can better understand how existing scheduling techniques may be applicable to the chemotherapy problem.

- A methodology for creating and evaluating problem definitions is proposed and used to create a model of the chemotherapy problem. This methodology is a contribution to the operations research literature because it demonstrates the importance of studying different definitions when modelling a complex, real problem. By creating an optimization model of the chemotherapy outpatient scheduling problem, a contribution is also made to health care scheduling research. By using constraint programming as the optimization technique, a contribution is made on applying constraint programming to scheduling problems.

- Dynamic template scheduling is introduced. Dynamic template scheduling is a novel technique for scheduling with uncertainty when appointments are required to be booked in real time. The creation of and experimentation with dynamic template scheduling is a contribution to scheduling research. As there are few techniques that address real-time uncertainty in complex situations and many health care applications contain real-time uncertainty, dynamic template scheduling is also a contribution to health care scheduling research.

- Dynamic template scheduling works by using an approximation of future appointments requests that will need to be scheduled on a given day. In order to create the approximation of future appointments, statistical analysis of trending appointment requests is required and is a contribution to the research on dynamic scheduling.

- To solve the chemotherapy outpatient scheduling problem, a complete scheduling system is created that addresses the real time and last minute uncertainty inherent in the problem. This method is shown to work at two centres, contributing to health care scheduling research.

Chapter 2

[redacted] Literature Review

In this chapter, an overview of scheduling research relevant to the chemotherapy outpatient scheduling problem is provided. Section 2.1 reviews classic deterministic scheduling models that are well-studied and similar in structure to the chemotherapy problem: flow shop scheduling and resource constrained project scheduling. Section 2.2 provides an overview of constraint programming, the optimization technique used in this dissertation. Section 2.3 then looks at techniques in the literature that address scheduling under uncertainty. Section 2.4 reviews other scheduling applications with similar characteristics to the chemotherapy problem: radiation therapy scheduling, operating room scheduling, and outpatient appointment scheduling. Section 2.5 examines the current state of chemotherapy scheduling research. Finally, Section 2.6 provides an overview of existing methodologies for creating mathematical models from real world, complex problems.

2.1 Classic Deterministic Scheduling Models

Scheduling problems can be classified as deterministic and static or stochastic and dynamic (Pinedo 2012). A deterministic problem has a finite number of jobs and all information pertaining to those jobs is known in advance. A stochastic problem contains some information that is not known in advance. Examples of stochastic information are machines that are subject to breakdowns, unknown demand, and jobs with stochastic processing times. The models discussed in this section are all problems. Although the arrival of chemotherapy appointment requests is stochastic and appointments must be scheduled before all problem information is known, understanding and modeling the deterministic version of the chemotherapy problem is important for this dissertation.

Much of the scheduling literature focuses on scheduling in the context of manufacturing (Graves 1981, Herrmann 2006), which includes project and machine shop scheduling. Given the large volume of research on scheduling, classification schemes have been developed to describe machine and project scheduling problem structures. Here, the classification for machine shop scheduling described by Pinedo (2012) is combined with the project scheduling classification described by Brucker, et al. (1999). The classification of problems describes the machine or resource environment, the job or activity characteristics and processing constraints, and the objective function.

The problem is assumed to consist of a number of activities or jobs that require machines or resources. A job may have a number of operations and each job has an associated processing time for each operation and each machine. A job may also have an associated release or ready time, a due date, a set of operations that must precede it, a set of operations that must come after it, and a weight.

In this section, two configurations are reviewed because they are the most relevant to the chemotherapy problem: the flow shop and resource constrained project scheduling problems.

2.1.1 Deterministic Flow Shop Problems

Consider the manufacturing problem of scheduling orders for a product. Each order is translated into a job with a due date and each job requires the production of a product, which in turn requires processing on a set of machines (Pinedo 2005). There are precedence constraints because some operations in a job must be completed before others, machine eligibility constraints as some operations can only be done on certain machines, workforce constraints because only certain workers are qualified to operate particular machines, and due dates because orders must be completed by a certain time.

The standard flow shop model has one machine (or stage) for each operation, the jobs have due dates, and the objective is to minimize the maximum tardiness which is the largest deviation between a job's completion time and its due date. In a flow shop, each job has the same machine sequence. If a flow shop has the simplification that all jobs must be processed in the same order on all machines it is called a permutation flow shop. The permutation flow shop problem with three or more machines is known to be NP hard (Framinan et al. 2004). For a formal definition of a permutation flow shop problem on m machines with the objective of minimizing makespan, see Framinan et al. (2004).

If a flow shop has at least one stage where there is more than one machine available and each job can be processed on any of the machines at that operation, it is categorized as a flexible flow shop (Ruiz and Vazquez-Rodrigez 2009). It may have two parallel machines at each operation, where each job requires processing in each operation for given lengths of time, and the objective is to minimize the makespan. A two-stage flexible flow shop with two machines at one of the stages is proven to be NP-hard by Hoogeveen, et al. (1996).

Approaches to solving the permutation and flexible flow shop include dispatching rules (Rad et al. 2006, Gupta and Tunc 1994, Dessouky et al. 1998, Brah and Wheeler 1998, Hong et al. 2007, Naderi 2010), local search (Ben-Daya and Al-Fawzan 1998, Grabowski and Wodeki 2004, Nowicki and Smutniki 1998), and exact algorithms such as mixed integer programming (Buscher and Shen 2009, Sherali et al. 1990), branch and bound (Dessouky et al. 1998, Haouri et al. 2006), and search and prune (Hong et al. 2007). Several studies compare the use of heuristics to exact methods (Dessouky et al. 1998, Hong et al. 2007) and many techniques for flexible flow shop problems are limited to two stages (Gupta and Tunc 1994, Haouri et al. 2006, Hong et al. 2007).

Exact approaches have been shown to find optimal solutions to problems from ten jobs on four machines (Buscher and Shen 2009) to 200 jobs in a two-stage flow shop where there are two machines in the first stage and four machines in the second stage (Haouri et al. 2006). These approaches are based on techniques to strengthen the lower and upper bounds used during the search. In contrast, heuristics were able to find feasible solutions to problems with up to 9000 jobs in as little as 50 seconds (Hong et al. 2007). However, the quality of these solutions is hard to evaluate. On small problems where optimal solutions are known (e.g., with eight jobs), heuristics often found feasible solutions within 5% of optimal.

As an example of a local search technique, Ben-Daya and Al-Fawzan (1998) used a tabu search to solve a simple permutation flow shop. Their paper highlights the type of technique that is necessary to solve large flow shop problems and it is helpful that they tested their proposed solutions on a standard set of benchmarks created by Taillard (1993) with many test problems ranging in size from 20 jobs on 5 machines to 500 jobs on 20 machines. Ben-Daya and Al-Fawzan (1998) created an initial sequence of jobs using the known NEH heuristic (Nawaz et al. 1983). Their tabu search had several unique features including a neighborhood created by random swapping, insertion, and block insertion, as well as a diversification and intensification scheme that used a measure of frequency, how often a job in a chosen schedule is found in a particular location. The tabu search resulted in solutions that deviated from known lower bounds by less than 9%.

Hong et al. (2007) provide a study that compares exact and heuristic techniques to solve a two stage flexible flow shop with the same number of machines at each stage. The heuristic took one machine from each stage to form machine groups that could each be modelled as a simple flow shop. Jobs were clustered into *n* groups and scheduled in *n* simple flow shops using an earliest completion time first dispatching rule. The exact algorithm tested all possible permutations. Hong et al. (2007) showed that the exact algorithm was able to find the optimal solution for problem instances with up to eight jobs. For these problems, the heuristic found solutions within 5% of the optimal value. The heuristic was also able to find feasible solutions to problems with 1000 to 9000 jobs within 50 seconds.

In conclusion, deterministic flexible flow shop problems are solved in the literature by both exact and heuristic methods. For larger problems, exact methods are not able to find optimal solutions. However, heuristics and local search techniques seem to work well as is shown by Hong et al. (2007) and Naderi et al. (2010). More detailed surveys of techniques for solving flow shop problems can be found in Faramin et al. (2004), Hejazi and Saghafian (2005), and Ruiz and Vazquez-Rodriguez (2009). In general, flexible flow shop problems with multiple stages such as the chemotherapy outpatient scheduling problem are very hard to solve and it appears that the best approach is to use heuristics or a combination of methods.

Note that the second stage of treatment aloe can be seen as a one-stage flexible flow shop as it requires patients to be partitioned amongst a set of parallel resources (the chairs) to minimize makespan. It is easy to see that this stage is equivalent to the set partition problem, a well-known NP-complete problem (Garey and Johnson 1979). Hence, the chemotherapy problem is at least as hard.

2.1.2 Deterministic Resource Constrained Project Scheduling Problems

The chemotherapy outpatient scheduling problem may also be defined as a resource constrained project scheduling problem (RCPSP) as it has complex resource constraints that are reflective of constraints found in RCPSPs (Brucker, et al. 1999). A deterministic RCPSP considers resources of limited availability and a set of activities with known durations, resource requirements, and precedence requirements (Brucker et al. 1999). The problem consists of assigning start times to each activity such that all precedence and resource requirements are fulfilled and resource availabilities are respected. For a formal definition of a RCPSP, see Fleszar and Hindi (2004).

Approaches to solving the RCPSP range from heuristic rule-based techniques (Lu and Li 2003, Basnet et al. 2001) to local search techniques (Goncalves et al. 2008, Thomas and Salhi 1998, Fleszar and Hindi 2004), and exact approaches (De Reyck and Herroelen 1998, Baptiste and Le Pape 1997, Berthold et al. 2010, Kone et al. 2011).

The exact approaches used include constraint programming (Baptiste and Le Pape 1997), mixed integer-linear programming (Kone et al. 2011), branch and bound (De Reyck and Herroelen 1998), and constraint programming and integer programming combinations (Berthold et al. 2010). Exact approaches have been shown to solve up to 96% of benchmark problem instances (Patterson 1984, Kolisch et al. 1995) with 30 or less activities in 93 seconds.

Heuristic and local search techniques include genetic algorithms (Gonclaves et al. 2008), tabu search (Thomas and Salhi 1998), the critical path method (Lu and Li 2003), and beam search (Basnet et al. 2001). These approaches found solutions within 0.69% of optimal for smaller problems and find feasible solutions to the larger problems.

A paper by De Reyck and Herroelen (1998) is a good example of how to handle varying resource requirements in RCPSPs. In order to reduce the solution space they preprocessed resource requirements and enforced precedence between activities. For every group of activities that could not be scheduled at the same time, they constructed minimal delaying modes, sets of activities that, when delayed, release enough resources for the remaining activities in the group. They evaluated each delaying mode to see if it was time-feasible and if it was, they branched on it. They generated 7200 problem instances with 10 to 100 activities, 5 to 8 resource types, and a range of resource requirements for the activities. An average of 68% solved to optimality and overall the solutions were 1.23% away from the best known solutions.

As an example of a heuristic method that worked well for large RCPSPs, Fezsar and Hindi (2004) used a variable neighbourhood search. They created several initial solutions using various heuristics, including shortest processing time first and minimum slack first. A neighbourhood was created by moving any one activity to a new position. The search also included enhanced moves, where activities in precedence or succession of chosen activities moved as well. The authors tested their technique on the set of benchmarks by Kolisch et al. (1995) with 30 activities. Their solutions deviated from the best known solutions by 0.01 to 4.08% and they were able to find feasible solutions to the largest problems in an average of 220 seconds.

In contrast to the above examples, Kone et al. (2011) presented an exact solution method. They created event-based mixed integer programming models to solve the RCPSP. Instead of the typical discrete representation of variables, where a variable will be assigned the value 1 if a certain activity is assigned to a resource at a particular time, Kone et al. (2011) use many fewer variables. One of their models, named *start and end events,* had variables representing the start and end of activities. A decision variable is one if an activity starts at event *e*. A continuous variable, *t*, is used to represent the start date. The authors tested their models using the benchmarks created by Kolisch et al. (1995) with 30 activities. Within 500 seconds, an imposed time limit, the model was not able to reach the optimal solution, but was able to find solutions that deviated an average of 0.61% from optimal.

As in flexible flow shop scheduling, the literature shows that heuristic and local search techniques work best for problems similar in size to the chemotherapy outpatient scheduling problem at the Odette Cancer Centre. Additionally, in the resource constrained project scheduling domain, the techniques used to handle varying resource requirements are important because the chemotherapy problem has varying requirements for nurses at different stages of the problem. More detailed surveys of the techniques used to solve resource constrained project scheduling problems can be found in Brucker et al. (1999) and Yang et al. (2001).

2.2 Constraint Programming

Constraint programming is described as the study of computational systems based on constraints (Rossi et al. 2006). A constraint satisfaction problem is a set of variables each with a finite domain and a set of constraints that restrict which values the variables can take. A solution is an assignment to every variable such that all constraints are satisfied. The search can be for any solution, all solutions, or for an optimal solution defined by an objective function. Standard constraint programming solving is a combination of search with propagation at each node (inferring new problem restrictions based on the current domains of each variable).

2.2.1 Search

There are different methods for systematically searching through the search space. The classic approach is chronological backtracking (Bartak 1999) that incrementally assigns values to variables. When the next variable cannot be assigned a value, the algorithm goes back one step in the tree and assigns a different value to that variable. Different search methods and heuristics have been developed to decide which variable to assign next, what value to assign to the variable in question, and how to backtrack when there is no feasible assignment for a variable (Prosser 1993, Smith and Grant 1998).

2.2.2 Propagation

As values are assigned to variables, the constraints are propagated by checking to see what values can be removed from the domains of the remaining unassigned variables. The more values that can be removed from the domains of variables, the more efficient the search process can be. A simple form of propagation is forward checking. After a variable is assigned, forward checking checks the unassigned variables directly connected to the just-assigned variable and removes values that conflict with the just-made assignment. A compromise is made between the level of propagation and the cost of performing that propagation at every node in the search (Rossi et al. 2006).

2.2.3 Global Constraints

Global constraints are constraints over multiple variables with specialized propagation algorithms that are more efficient than if the same constraint was expressed using several smaller constraints (Rossi et al. 2006). The Global Constraint Catalogue (Beldiceanu and Demassey 2010) lists several hundred global constraints representing efficient interface techniques for a variety of problem substructure that arises in combinatorial optimaization. For example, the disjunctive and cumulative constraints (Beldiceanu and Demassey 2010) are global constraints that can be used to model resource constraints. The disjunctive constraint ensures that activities allocated to the same resource do not overlap in time. The cumulative constraint generalizes disjunctive ensuring that a set of activities executing at the same time do not exceed the capacity of their common resource.

2.2.4 Constraint-Based Scheduling

Constraint-based scheduling is the term used for applying constraint programming to scheduling problems (Baptiste et al. 2001). There is significant research within constraint programming that looks at creating propagation and search algorithms for scheduling problems and constraint programming is acknowledged as a technique well suited for scheduling problems (Bartak et al. 2010, Beck and Fox 1998). Ku (2013) tested constraint programming against three different MIP models and a state-of-the-art tabu search technique on a set of flexible job shop problems, where processing times for jobs may differ based on their assigned machine. Ku found that constraint programming dominated the MIP in terms of finding the best solutions and proving optimality and was competitive against a state of the art tabu search algorithm.

Constraint programming is ideal when there are many interacting constraints, such as in scheduling problems with both temporal and resource constraints, because the constraints can be propagated and guide the search effectively (Beck and Fox 1998).

Many researchers have created and validated propagation (Baptiste and LePape 1997) and search algorithms (Godard et al. 2005, Gueret et al.2000) that are designed for modelling scheduling problems. The specially designed algorithms are complemented by the frameworks that exist for implementing constraint programming models, such as comet and CPLEX (Van Hentenryck 2008, ILOG). Comet, the framework used in this dissertation, is freely available to students, has a library of global constraints, has algorithms designed for scheduling problems built into a specific scheduling module, separates modelling and search making it easy to use, and contains a powerful language that can capture the complex structures of our scheduling problem (Van Hentenryck 2008).

2.3 Scheduling With Uncertainty

The chemotherapy outpatient scheduling problem has two main sources of uncertainty. First, appointments must be scheduled in real time as requests arrive, before the full list of requests for a given day is known. Second, closer to the day, changes to the schedule occur due to urgent appointments needing to be added to the schedule and appointments being cancelled. Even small amounts of uncertainty can make a scheduling problem much more complex. For example, a single machine problem subject to jobs with stochastic durations and an objective of minimizing lateness is shown to be NP-hard (Daniels and Carrillo 1997).

In the last fifteen years, under the heading of robust scheduling, techniques for solving problems with uncertainty have been a focus of many researchers (Herroelen and Leus 2001, Mohring and Stork 2000, Lambrechts 2008, Billaut and Roubellat 1996, Daniels and Carrillo 1997, Kouvelis et al. 2000, Mehta and Uzsoy 1998, Van Hentenryck and Bent 2006). Scheduling with uncertainty falls into three categories: (1) proactive where an initial schedule takes into account uncertainty, (2) reactive where information is revealed in real time and jobs are scheduled or rescheduled after information is revealed, and (3) predictive-reactive where a combination of the both proactive and reactive methods are used (Ouelhadj and Petrovic 2009, Bidot et al. 2009, Davenport and Beck 1999).

Techniques for proactive scheduling include adding idle time into the schedule (Davenport et al. 2001, Mehta and Uzsoy 1998, Herroelen and Leus 2001) and including a measure of problem variability in the schedule (Daniels and Carrillo 1997). The approaches used are both exact (Daniels and Carrillo 1997, Wu et al. 2009) as well as heuristic approaches (Mehta and Uzsoy 1998, Herroelen and Leus 2001).

Reactive or online scheduling involves making decisions as information is revealed dynamically. Reactive algorithms use whatever information is available at the time, along with past choices to make decisions in real time (Borodin and El-Yaniv 1998). Approaches for reactive scheduling may use rules to decide which jobs to schedule (Mohring 2000) and may include exact techniques in order to make optimal decisions (Van Hentenryck et al. 2009). Predictive-reactive approaches are also a combination of exact (Li and Ierapetrirou 2008) and heuristic methods (Bidot et al. 2009, Odonovan 1999, Billaut and Roubellat 1996, Lambrechts 2008).

As an example of proactive scheduling that uses known distributions from the problem domain, Mehta and Uzsoy (1998) created a heuristic to determine the amount and location of idle time to insert in the schedule in order to minimize the impact of machine breakdowns. The solution quality is measured by the deviation between the executed schedule and planned schedule. The authors first determined the sequence of operations using the shifting bottleneck heuristic (Adams et al. 1988) and then inserted idle time. A surrogate measure of predictability that included the expected distribution of breakdowns and repair times was used to determine the amount of idle time to insert before each job. Mehta and Uzsoy (1998) also considered a linear programming based heuristic to determine the inserted idle time. Resulting solutions had minimal deviation between executed and planned schedules with minimal sacrifice to the objective of minimizing lateness for problem instances with 10, 20 or 30 jobs and 5 or 10 machines.

As an example of a predictive-reactive approach, Lambrechts (2008) began by creating a predictive schedule for a resource constrained project scheduling problem using a heuristic that calculated the impact of each activity. A high impact activity was defined as being constrained to precede many other activities. Additionally, it was assumed that earlier in the project, activities were less prone to disruption than later on. Therefore, activities with high impact on others were scheduled earlier so that they would be less prone to disruptions. A heuristic was also used to add slack to the schedule. As the reactive piece, a priority list was made of all the activities not yet completed in order of baseline starting times. When a disruption occurred, the first activity on the list was rescheduled to be as early as possible. Lambrechts also included the option of a tabu search at this point. Experiments were conducted using the Kolisch test sets with 30 activities with uncertainty added. The priority list used for rescheduling performed 77% better than rescheduling activities in a random order.

In response to the dynamic nature of many real world problems, Van Hentenryck and Bent (2006) developed a class of algorithms that use proactive techniques to solve a variety of stochastic optimization problems. Decisions are made in real time by solving a deterministic optimization model of a sample of the future. Van Hentenryck et al. (2009) applied the technique to a real-time reservation problem where requests arrive in real time and must be dynamically allocated to limited resources to maximize profit. When a request arrived, its assignment was determined using an algorithm that made use of an approximation of future expected requests. Several algorithms were described for making that decision. One algorithm, named *Consensus*, described here because it is the most basic, uses an exact solution method at each decision point. Every time a request needed to be assigned to a resource, several samples of the future arrivals were taken and used to represent possible sets of future requests. Each sample was solved to optimality by assuming that the sample was the actual future realization of request arrivals. The solution to each sample was analyzed and a point was given to the resource that the current request was assigned. The resource with the most points over all the samples was then chosen as the assignment made to the current request.

There are many types of uncertainty in real world problems and many ways of modelling the uncertainty using proactive, reactive, and combined methods. In the literature, the combination of proactive and reactive methods appears particularly useful for handling both the real-time arrival of appointment requests and last minute uncertainty that exists in the chemotherapy problem (Lambrechts 2008, Van Hentenryck and Bent 2006). Additionally, as noted by Aytung et al. (2005), there is a need to exploit the known problem information while scheduling, as is done by Mehta and Uzsoy (1998) and Van Hentenryck and Bent (2006). At the Odette Cancer Centre, there is a feeling that there is some predictability as to what appointment requests can be expected on a given day. Such information could be used to create a stronger proactive schedule.

More detailed surveys of techniques for solving problems with uncertainty can be found in Aytung et al. (2005), Herroelen and Leus (2004), Davenport and Beck (1999), and Ouelhadj and Petrovic (2009).

2.4 Scheduling Applications in Health Care

Many health care applications have features in common. Among the most obvious, they all concern patients and need to be implemented within a health care setting. In this section, three scheduling applications within the health care domain are reviewed: (1) radiation therapy patient scheduling, (2) operating room scheduling, and (3) outpatient appointment scheduling. The common characteristics with chemotherapy outpatient scheduling are outlined in Table 2.1.

Table 2.1. Matching characteristics of the chemotherapy outpatient scheduling problem. Shared resources refers to the case where a resource is shared among multiple jobs at the same time. In the chemotherapy problem, nurses monitor several patients at the same time. In the radiation and appointment scheduling problems, all resources are dedicated to one patient at a time.

Problem Characteristic	Chemotherapy Outpatient Scheduling	Radiation Therapy Scheduling	Operating Room Scheduling	Appointment Scheduling
Multiple steps within a treatment	Y	Y	Y	N
Real time uncertainty	Y	Y	Y	Y
Uncertainty due to last minute changes	Y	Y	Y	Y
Objective to maximize utilization	Y	Y	Y	Y
Objective to minimize wait times	Y	Y	Y	Y
Large variability in treatment lengths	Y	N	Y	N
Shared Resources	Y	N	Y	N

2.4.1 Radiation Therapy

Radiation therapy gives rise to scheduling problems as patients must be booked to receive their treatment. Typically, a patient needs daily appointments for a set number of consecutive days. Each treatment appointment requires a technician and a linear accelerator, which delivers the radiation. Like the chemotherapy problem, the radiation therapy patient scheduling problem has uncertainty due to real-time arrival of appointment requests and last minute changes (Petrovic 2006). Within radiation therapy scheduling, there are a number of variations in problem definitions including assigning appointments to days and assigning start times to patients on a particular day.

Petrovic et al. (2006) modelled the radiation therapy patient scheduling problem as a job shop, where patients had to be assigned a certain number of appointments. There were two types of linear accelerators with associated processing times. There are three patient classes with associated wait time targets between the creation of a treatment plan to the start of the first treatment. Petrovic et al. (2006) tested two heuristics for booking appointments. They ordered patients on a priority list based on the time until their wait list target would be met. They then assigned each patient an earliest possible start date and latest possible start date based on their target and when they would have completed all the pre-treatment planning appointments. The first algorithm, *As Soon As Possible*, had patients start on the first possible start date in order of priority. The second, *Just in Time*, gave patients the latest possible start date. The authors tested the heuristics using real hospital data for a hospital that serviced around 160 patients each month. They found that *Just in Time* had better performance for palliative patients and *As Soon As Possible* worked better for radical and emergency patients. The authors noted the need to incorporate some form of rescheduling in their future work. Although there are many resources involved in the radiation problem such as technicians and nurses, the only resource considered was the linear accelerator. We were not able to find any future work.

Conforti and Guerrierco (2008) created an integer programming model to solve the radiation therapy patient scheduling problem with the objective of treating as many patients as possible in a day. They assumed that daily use of the linear accelerator could be modeled as a sequence of temporal blocks of the same duration. The authors tested their model using a case study of an actual clinic that had one accelerator available, with 10 15-minute slots on each day. Each patient required use of the machine for 15 minutes, for four or five treatments on consecutive days. The model was able to optimally schedule patients and achieved better results than were achieved by the real clinic. The shortcomings of their model are that uncertainty was not considered and the problem size was very small.

The problem structure and objectives of the radiation therapy patient scheduling problem are similar to the chemotherapy outpatient scheduling problem. The formulation of the radiation therapy problem as a job shop by Petrovic et al. (2006) inspired the formulation of the chemotherapy problem in this research. Both heuristic and exact approaches have been successfully used for the radiation therapy patient scheduling problem. However, in the little research that exists, uncertainty is not considered. Additionally, there are features that exist in the chemotherapy problem that are not found in the radiation therapy problem: many different treatment lengths and varying resource requirements. Therefore, it is useful to turn to radiation therapy treatment scheduling when looking to define and study the chemotherapy problem, yet it would be very difficult to directly apply any of the solution techniques.

2.4.2 Operating Room Scheduling

Operating rooms are a large cost centre and potential revenue generating area of hospitals (Denton et al. 2007). As a result, there is a large research base of operating room scheduling papers (Cardoen et al. 2009) with a number of variations in problem definitions including assigning operating room time-blocks to units or surgeons, assigning surgeries to days, sequencing the surgeries on a given day, and assigning start times to patients on a particular day. Our problem is closest to operating room problems that focus on the task of sequencing and assigning start times. Such problems involve sequencing a set of surgeries into a given number of operating rooms, taking into account resources such as surgeons, nurses, anesthetists, staff, and space in both the pre-operation and post-operation stages. Operating room scheduling is subject to significant uncertainty in surgery durations. Objectives for operating room scheduling include maximizing throughput, minimizing patient wait time, minimizing operating room idle time, and minimizing tardiness from the target schedule (Cardoen et al. 2009).

Operating room scheduling is tackled by exact (Adan et al. 2008, Denton et al. 2007) and heuristic (Marcon and Dexter 2006, Hsu et al. 2003, Denton et al. 2007) approaches. Exact solution techniques include mixed integer programming (Adan et al. 2008, Denton et al. 2007) and heuristic approaches include tabu search (Hsu et al. 2003), discrete event simulation (Marcon and Dexter 2006), and sequencing heuristics (Denton et al. 2007). Several researchers include stochastic durations when creating a proactive schedule (Adan et al. 2008, Denton et al. 2007) or in their analysis of proposed schedules (Marcon and Dexter 2006, Van Houdenhoven et al. 2007).

Adan et al. (2008) considered a thoracic surgery centre that services eight patient types. The existing system at the centre aims to optimize operating room utilization, yet the authors wanted to show that by considering the stochastic length of stays in the pre- and post-operating areas, they could achieve a better use of overall resources. Adan et al. (2008) only considered elective surgeries and assumed that a predetermined time was reserved for emergency surgeries. They created two mixed integer programming models that considered pre-operative care, use of the operating room, and post-operative care. First, they created a deterministic model that utilized the average lengths of stay. Second, they created a stochastic model that utilized an empirical distribution of lengths of stay based on retrospective data. The objective was to get as close as possible to the target utilization of each resource, while respecting capacity constraints. The solver was given 24 hours to solve, after which the stochastic model was stopped, and there remained a 12% deviation from the optimal operating room utilization of the deterministic schedule. Adan et al. (2008) created schedules with the two models and tested the schedules using historical data. Results showed that although the operating room utilization was lower than the stochastic model, overall they were able to achieve better utilization targets for the pre-operative, operating room, and post-operative areas combined.

Denton et al. (2007) created a two-phase stochastic mixed integer programming model for the operating room scheduling problem. The first phase assigned surgeries to days and the second phase sequenced the surgeries to minimize operating room idle time and patient wait time. The model was not able to find a feasible solution in a 24-hour time period, so the authors experimented with three heuristics to replace the sequencing phase of the model. Of the three heuristics, the one that sequenced surgeries in order of increasing variance performed best, achieving a deviation of approximately 4% from optimal.

Operating room scheduling is a well-established area of research, where uncertainty is often taken into account. As in other scheduling domains, exact solutions are not able to find optimal solutions for large problems, especially when uncertainty is considered. Heuristic or combinations of methods tend to be more successful as is shown in Denton et al. (2007). Operating room scheduling is similar to chemotherapy scheduling in that the objective is to minimize the makespan of a sequencing problem where there is precedence between appointment stages and a fair amount of uncertainty. However, the uncertainty is due to stochastic surgery duration or length of stay as opposed to real-time arrivals of appointment requests. Additionally, operating room scheduling rarely involves varying resource requirements. Therefore, techniques used to solve operating room scheduling problems cannot be directly applied to the chemotherapy problem.

More detailed surveys of techniques used to solve operating room scheduling problems can be found in Cardoen et al. (2009). Cardoen et al. (2009) groups papers on operating room scheduling by various criteria including patient characteristics, performance measures, decision level, type of analysis, uncertainty considerations, and applicability to implementation.

2.4.3 Outpatient Appointment Scheduling

Outpatient appointment scheduling involves scheduling patients into a primary care or outpatient clinic. The standard method is to use an appointment rule comprised of two parts: setting the appointment size (i.e., the number of patients scheduled to start at the same time) and the appointment interval (i.e., the time between appointment start times) (Cayirli and Veral 2003). Appointment rules may also specify an initial block size, the number of patients that are given the first appointment in a session, which may differ from the general block size. Patients are typically scheduled for an appointment with a single server (Cayirli and Veral 2003). Although the allotted time per patient is equal for all patients, the actual duration of an appointment is unknown. There is also uncertainty due to patients requiring urgent appointments, walk-ins, and patients not showing for their appointments (“no-shows”).

Approaches for appointment scheduling either include service time distributions in a stochastic model (Erdogen and Denton 2011, Muthuraman and Lawley 2007) or test appointment rules in various situations using simulation (Vissers and Wijngaard 1978, Cayirli et al. 2006).

Erdogen and Denton (2011) created two stochastic linear programming models with the objectives of minimizing doctor idle time and patient wait time. The first model used a list of patients to schedule, each with a probability of not showing for their appointment. The second model scheduled patients as requests for appointments arrived in a multi-stage stochastic model. Uniform and lognormal service durations were considered for a server that could service ten patients a day. Results of sensitivity analysis varying the probability of no-shows and the costs associated with doctor idle and patient wait times showed that a higher probability of no-shows caused the inter-arrival time to increase. In cases where both the no-show probability and idle time cost were high, over-booking became optimal.

Cayirli et al. (2006) used empirical data to formulate a simulation model to test six sequencing rules and seven appointment rules. They considered two types of patients, new and return, each with their own service time distribution. New patients require more time on average than return patients. The sequencing rules referred to the order in which new and return patients were seen. For example, one sequencing rule was to alternate new and return patients and another was to service all new patients at the beginning of the day. The appointment rules tested included the an appointment size of one with a fixed appointment interval set to the mean service time, an appointment size of two with increasing intervals towards the middle of the session and then decreasing, and others. Cayirli et al. (2006) also modelled six factors: (1) number of patients, (2) coefficient of variation of service times, (3) no-show and (4) walk-in probabilities, (5) punctuality of patients, and (6) service time variability. The performance measures they considered were doctor idle time, doctor over time, and patient waiting time. They found that the number of patients, and the probability of no-shows and walk-ins were the most important factors influencing patient wait time. The number of patients, the punctuality of patients, and the probability of no-shows were the most important factors influencing doctor idle time and over time. With a higher number of patients, the mean service times are smaller and the overall performance was better. Cayirli et al. (2006) also found that the sequencing rule had more of an impact than the appointment rule.

In appointment scheduling research, the majority of the analytical studies are queuing models with one stage and one server (Erdogan and Denton 2011). The solution method of using an appointment rule is well accepted and the question for each outpatient clinic is what rule to use as there is a lack of research that looks at what setup might be generalizable to many different clinics (Cayirli and Veral 2003).

On the surface, the chemotherapy problem appears to be an appointment scheduling problem that requires the optimal scheduling of outpatient appointments. Both problems are dynamic. However, the chemotherapy problem is much more complex because it has multiple stages, many servers, and known appointment durations that differ significantly. Therefore, the queuing models that work well for appointment problems would not be able to handle the complexity of the chemotherapy problem. Simulation, such as the one created by Cayirli et al. (2006), could be helpful in testing various schedules in the chemotherapy clinic.

2.5 Chemotherapy Research

Within chemotherapy outpatient scheduling, there are a number of variations in problem definitions including assigning appointments to days and assigning start times to appointments on a particular day. The variation addressed in this research is to assign start times to appointments. We consider four stages (chart review, pharmacy preparation, and two stages of treatment) and two sources of uncertainty: the real-time arrival of appointment requests and uncertainty due to last minute changes. We consider visits to radiation therapy or an oncologist only if they impact the possible start time of the chemotherapy appointment. We are not concerned with assigning appointments to days, but rather we are concerned with scheduling one day of chemotherapy appointments at a time.

Research on scheduling and sequencing chemotherapy appointments is limited. Santibanez et al. (2009) created a simulation of the ambulatory care unit at the British Columbia Cancer Agency, a chemotherapy unit similar in size to the Odette Cancer Centre. The problem was to assign an appointment date to each request rather than a start time on a known date. Their simulation encompassed several clinics as well as the radiation and medical oncology units and was used to analyze many factors, including different scheduling techniques. Results showed that many factors must be changed at once to make a significant impact on wait times.

Santibanez (2009) also investigated the use of mixed integer programming for the chemotherapy outpatient scheduling problem at the British Columbia Cancer Agency. Here, Santibanez addressed a problem similar in structure to the problem considered in this dissertation, but with different objectives. He looked at scheduling one day of appointments with the objective of finding a schedule that minimized the maximum nurse workload with an even workload among the nurses, both objectives that are not considered in this dissertation. Santibanez simplified the problem by only considering the treatment stage and not taking into account uncertainty. However, he was able to find optimal solutions to the problem.

More recently Santibanez and the team in British Columbia, Canada published the results of their complete study creating a scheduling system for a large chemotherapy clinic (Santibanez et al. 2012). The paper is the culmination of a three year study including a review of the system they developed using their simulation and mixed integer programming model described above. They achieved significant operational improvement in both the lead time of when patients are notified of their appointments, with a 57% reduction in the number of patients exceeding the target of one week, and the wait list for first appointments, with an 83% reduction in wait list size. Neither lead time of appointment notification nor wait list size is a concern in this dissertation.

Turkcan et al. (2012) created an integer programming model to solve a version of the deterministic chemotherapy outpatient scheduling problem that focused on deciding which appointments to assign to which day, but also included sequencing of appointments on a given day. The objective of the model was to balance the acuity of patients serviced by a particular nurse. Unlike in our problem, pharmacy constraints were not considered. With problems that serviced 50 patients a day and took into account four types of cancers, they found that there were over 280,000 variables and over 8000 constraints. Turkcan et al. therefore used a longest treatment time first heuristic in order to decrease the solution time.

With the complexity and potential impact of the chemotherapy outpatient scheduling problem, we would expect to see more papers tackling this issue. However, there are very few studies of the chemotherapy problem and the ones that exist either solve much smaller problems than the ones considered in this research or omit significant aspects of the problem definition such as uncertainty or the pharmacy stage.

2.6 Methodologies for Modeling Real World Problems

In constraint programming, *modelling* refers to taking a mathematically well-defined problem and developing different, yet logically equivalent, models with the goal of solving the problem more efficiently (Frisch et al. 2005). However, before modelling can happen, a mathematically well-defined problem must be created. That step, which will be referred to as *problem definition*, is deciding what simplifications can be made to the real problem so the optimization model can be solved in a reasonable amount of time and result in a solution that is useful in the real world situation.

The concept of the problem definition is present in operations research (Taha 2007, Powell and Baker 2007). However, deciding which information from the real application to include in a mathematical model does not appear to have been formally studied in the operations research literature. In other domains, the problem definition stage is more established. In this section, we look at the problem definition stage in the areas of model-based diagnosis, software engineering, and operations research.

Model-based diagnosis uses accurate mathematical models of machines and systems to diagnose faults that occur in the real world (Andersson 1998). Using a model of a correctly behaving machine in combination with given abnormal observations, a hypothesis is produced as to which faults led to that abnormal behaviour. In the design of a domain model, a level of detail must be chosen that represents the machine accurately, yet is not too difficult to represent and has an acceptable computational complexity (Andersson 1998). One way to represent complex information is to use qualitative reasoning based on non-numerical descriptions of a system (Museros and Escrig 2003, Tokuro et al. 2003). Another way to simplify definitions for model-based diagnosis is to use the single fault assumption, which requires the problem be defined in such a way that every possible fault combination is modeled as a single component that may fail (Nunes de Barros et al. 1999). In the field of model-based diagnosis there is literature on which assumptions may be necessary (Musros and Excrig 2003, Tokuro et al. 2003, Nunes de Barros et al. 1999). However, there does not appear to be research comparing different problem definitions against each other.

When creating software, we need models of required data, information and control flow, and behaviour (Pressman 2000). The system requirements are often provided in vague terms by the clients and, as a result, subsequent specifications are incomplete and exist at differing levels of detail (Pressman 2000). Software engineers have developed semi-formal descriptions of the process of designing and building software systems termed *process models*, containing detailed descriptions of how and when to collect requirements, how to analyze those requirements to decide which are the most important, and how they should be included in the requirements definition. For example, two such *process models* are the classic waterfall model (Sacchi 2001) and the newer extreme programming model (Jeffries 2001).

In the field of operations research, there are many tasks that require creating a model of a real-world domain in order to solve a problem. Taha (2007) states that creating a problem definition refers to defining the scope of the model by choosing the decision alternatives, the objective, and the limitations of how the model will be used. He also includes the problem definition stage as the first phase of an operational research study. Powell and Baker (2007) outline the steps of moving from understanding the problem context to creating a model structure. The stage of simplifying a problem is present because it is necessary in order to find solutions to almost every problem. However, there is little written in operations research literature on what one simplification does in comparison to another and what implication each simplification has to fitting the solution into the real world.

2.7 Summary

This chapter reviewed scheduling literature relevant to the chemotherapy outpatient scheduling problem. We first looked at the common scheduling frameworks of flexible flow shop scheduling and resource constrained project scheduling. From those frameworks, even without taking into account uncertainty, the literature indicates that an exact solution technique on its own is unlikely to solve a problem of our size in a reasonable amount of time.

We then described constraint programming and its use in modelling and solving scheduling problems. The present research uses constraint programming to develop an optimization model for a deterministic version of the chemotherapy outpatient scheduling problem, the complex resource constraints are modelled with guidance from resource constraint project scheduling, and the different uncertainties are addressed in subsequent stages using techniques from robust scheduling.

Next, we looked at literature that deals with scheduling while considering uncertainty. In particular, the general framework created by Van Hentenryck and Bent (2006) seemed most applicable to our problem.

We subsequently studied three application areas in the health care domain: (1) radiation therapy patient scheduling, (2) operating room scheduling, and (3) outpatient appointment scheduling. Although they all contained certain features in common with the chemotherapy problem, none were a complete match. However, using simulation to test various scenarios, as was done in some appointment scheduling literature, is a technique that we use in Chapter 4.

Following the review of health care applications, we reviewed the literature on chemotherapy outpatient scheduling. Of the few papers that exist, we were not able to find one that took into account both uncertainty and the stages preceding the actual chemotherapy treatment. The chemotherapy outpatient scheduling problem has not been widely studied, yet chemotherapy treatment delivery is a global issue with potential for improvement. The chemotherapy problem contains some unique features and we can learn from literature in other scheduling areas when developing our approach.

We finished with a brief look at literature describing how to create models of real-world problems. Although a methodology for creating a model from a real-world problem is necessary, there are few formal methodologies documented in the operations research literature. It would be helpful, when tackling a real world optimization problem, to know what sorts of simplifying assumptions are useful given the nature of the problem, as in model-based diagnosis. It would be beneficial to have processes in place to follow in order to create problem definitions that lead to solvable models with solutions that work in the real world.

Chapter 3

[redacted] Chemotherapy Outpatient Scheduling Application

The main case study used in this research is the chemotherapy outpatient scheduling problem at the Odette Cancer Centre at Sunnybrook Health Sciences Centre in Toronto, Ontario, Canada. In this chapter, details of the process and available data at the Odette Cancer Centre are presented.

3.1 The Chemotherapy Treatment Process

This analysis is based on observations and interviews conducted with the schedulers in charge of booking and the resource nurse on duty at the Odette Cancer Centre during the month of June 2009.

When a patient arrives at the centre for a chemotherapy appointment, he presents himself to reception. At reception, the patient is verified and his information is reviewed to be sure it is correct. If the patient requires blood work, he is sent to the lab. If no blood work is required, he is sent to the waiting area.

The chemotherapy treatment process at the clinic can be divided into four main stages: (1) chart review, (2) pharmacy or drug preparation, and two stages of treatment, (3) the first stage of treatment or setting up the patient and (4) the second stage of treatment or delivery of drugs.

1. Chart review – When a patient arrives at the chemotherapy unit, a dedicated nurse reviews the patient chart. Occasionally, when the blood work is deemed questionable by the nurse, the patient's oncologist may need to see the patient before treatment for a more detailed assessment. The oncologist may adjust the dosages or cancel that chemotherapy appointment.
2. Drug preparation – If the patient is approved for treatment, pharmacy is contacted and the chart is placed in a queue to have the drugs prepared. There are several pharmacy technicians mixing drugs at once.
3. Setting up the patient – When a nurse is free, the nurse will take the next chart from the queue, check to see if the drugs are ready, and take the patient to a chair or bed to begin

treatment. Setting up the patient requires direct nursing care and the time of this stage varies for each patient and regimen. Nurses may not set up more than one patient at a time.

4. Delivery of drugs – In the second part of treatment, the drug is administered while the patient is monitored by a nurse. The same nurse might not monitor the same patient for the complete duration of their treatment, especially for a regimen that has a duration that is several hours. During this stage, nurses are able to monitor up to four patients at a time. Nurses may set up a patient while monitoring others, as long as the total number of patients the nurse is responsible for does not exceed four. Consider the case of a small chemotherapy centre with one nurse and four chairs. That nurse can set up the first patient and then bring in the second while still monitoring the first. Eventually the nurse would have four patients receiving treatment at the same time. When one finishes, the nurse can set up the next patient, while still monitoring the other three. Occasionally, a patient may need more direct attention during the monitoring stage and the nurse will not set up another patient at that time.

The overall process requires different numbers of resources (i.e., staff, chairs, and beds) at different times. There is only one chart review nurse. However, there are several pharmacy technicians, several nurses available for the treatment stages, and several chairs. The number of pharmacy technicians and nurses changes from day to day. The values used in this research are five technicians, 13 or 15 nurses, depending on the day in question, and 29 chairs and beds.

At the clinic, 70 to 120 solid tumor and hematology patients are treated each day. Because the Odette Cancer Centre is part of a teaching hospital, they participate in three or four clinical trials a month. Trials are often resource intensive, requiring additional time to prepare the drugs and longer direct nursing care. Trials also get priority over regularly scheduled patients. Trials that occurred during the study period are included in the experiments presented in this research.

The chemotherapy clinic schedule often has additions and cancellations. Emergency chemotherapy treatments are rare, but emergency hydrations are not. IV hydrations for patients suffering from dehydration because of their chemotherapy are conducted in the clinic and may require the use of a chair in the unit for up to four hours. Cancellations can occur from several weeks in advance up to the day of treatment. When a patient's first treatment appointment is booked, all subsequent appointments in that patient’s regimen are also booked as they occur at regular intervals. If the patient then requires one of his appointments to be cancelled because they are not approved for treatment during the chart review stage, the remainder of the appointments in his regimen must be moved as well.

The chemotherapy outpatient scheduling problem is stochastic and has a resource environment that is a four-stage flexible flow shop (see Chapter 2). The processing characteristics include precedence constraints between stages, temporal constraints that limit the time between stages, and ready times for patients with other appointments earlier in the day. The objective of the problem is to assign start times to all the appointment stages while minimizing makespan, the end time of the day.

In this dissertation, each calendar day is a separate problem instance. Requests that need to be scheduled on a given day may arrive from several weeks before until the day of treatment. When an appointment request arrives, the appointment reservation must be assigned immediately and the patient provided with their appointment time.

3.2 Problem Data

The data used for this research is based on one year of appointment data from the current scheduling system at the Odette Cancer Centre, from April 2009 through March 2010. The list of patients serviced on each day as well as the regimens and corresponding treatment durations were extracted from the scheduling software. A list of appointments that were booked and then cancelled was also provided. A subset of the data is used in various experiments in Chapters 4 through 6 as detailed in those chapters.

In the available data, the average number of appointments that occurred per day was 90.48 with a standard deviation of 17.49. The average number of appointment minutes per day was 10909.09 minutes (the equivalent of 181.8 hours or 7.6 days) with a standard deviation of 1699.38 minutes (28.3 hours or 1.1 days). Treatment lengths ranged from 15 to 510 minutes, with the most common appointment duration being 15 minutes. The average day lasted 9.2 hours or 552 minutes.

Table 3.1. Utilization of nurses, pharmacy technicians, and chairs for the set of appointments that occurred

Resource	Number Available	Capacity (minutes)	Minutes Used	Utilization (%)
Nurses	14	37125	12511	33.70
Pharmacy	5	2760	2550	92.39
Chairs	29	16008	10996	68.69

Table 3.1 shows the average utilization of the nurses, pharmacy technicians, and chairs based on the treatments that occurred and capacity measured by multiplying the number of resources available by the average day length. We see that pharmacy technicians are highly utilized, while the other resources are not.

The average number of appointments booked per day including those appointments that were subsequently cancelled was 120.75 with a standard deviation of 17.56 appointments. The average number of treatment minutes per day for those appointments was 13665.82 minutes (the equivalent of 227.8 hours or 9.5 days) with a standard deviation of 1739.69 minutes (29 hours or 1.2 days). Table 3.2 shows the average utilization of the nurses, pharmacy technicians, and chairs based on the set of appointments booked per day including those appointments that were subsequently cancelled. Again we see that the pharmacy technicians are the most utilized resource. If all the appointments booked had occurred, the pharmacy technicians would not have had the capacity to service all the appointments.

Table 3.2. Utilization of nurses, pharmacy technicians, and chairs for the set of appointments that were booked

Resource	Number Available	Capacity (minutes)	Minutes Used	Utilization (%)
Nurses	14	37125	15480	41.70
Pharmacy	5	2760	3150	114.13
Chairs	29	16008	13665	85.36

Chapter 4

[redacted] Developing and Choosing an Optimization Model

In the next three chapters, we present dynamic template scheduling, our approach to solving the chemotherapy outpatient scheduling problem. This approach uses an optimization model of the deterministic chemotherapy problem (Chapter 4) within a dynamic algorithm to address real-time uncertainty (Chapter 5) and uncertainty due to additions and cancellations (Chapter 6).

During dynamic template scheduling, there are three sets of appointments: (1) R is the set of appointment requests that arrive in real time and must be scheduled by the user, (2) F is the set of already scheduled appointments, and (3) E is a sample of expected appointments based on a distribution created using historical data from the problem domain.

The first step in dynamic template scheduling is to create a proactive template of an expected day at the chemotherapy centre. An optimization model of the deterministic chemotherapy problem is used to optimally schedule E. The optimal schedule of E becomes open timeslots where requests for appointments can be booked.

As requests for appointments arrive in real time, the user assigns start times by matching appointment requests to open slots in the template. If a request arrives that cannot be scheduled within the template, the template is dynamically updated. Any appointments that have already been scheduled become the set F. The current request plus a new sample of remaining expected appointments E are then passed to the deterministic optimization model along with the set F to create a new template around the already scheduled appointments. The process of assigning start times to appointment requests in R and dynamically updating the template continues until all requests for a given day are scheduled.

In this chapter, we present the optimization model of the deterministic chemotherapy outpatient scheduling problem. Several constraint programming formulations are developed, each containing different simplifications, to represent possible definitions of the real-world problem. The definitions are evaluated on their solvability and quality.

This chapter is organized as follows. The next section provides a proposed method for evaluating different problem definitions. Next, we describe the constraint model formulations, a MIP model, and a simulation model that is used to compare the different problem definitions in a simulated setting. The models are followed by a description of model validation. Then, the experiments and results are presented. The final section presents a conclusion.

4.1 Method of Evaluating Problem Definitions

Including more information in the formulation of a real-world application provides a more accurate model, but can make modelling, data gathering, maintenance, and problem solving more challenging. Creating a problem definition is the act of describing the real-world domain by deciding what information to include in a mathematical model so that solvability and quality are both acceptable to the end users. *Solvability* refers to the time it takes for the model to deliver a solution. An acceptable time will depend on the dynamics of the application. For example, when scheduling in advance for a manufacturing facility, a model that takes an hour to solve may be more than fast enough, if the problem requires a schedule be produced once a day (e.g., overnight). However, when scheduling for an application that requires a schedule in real time, an acceptable time may be a few seconds or even less. *Quality* refers to how useful the solution is to the end users. Acceptable quality will again depend on the application. For example, one application may be satisfied with any feasible solution; while another may require an optimal solution.

In response to the lack of formal methods in operations research literature to guide the creation of problem definitions (see Chapter 2), we propose the following iterative method:

1. Create a *base* model that represents the application. A base model is the model we choose to start with. The creation of a truly complete model is likely not feasible and so it is necessary to omit some details. Any assumptions made should be discussed with the end users of the system and then documented. At the end of an iteration of the method, centred on what we have found through our experiments, we may generate a new base model for the next iteration, possibly including details omitted in previous iterations.

2. Determine a list of potential elements that can be added, removed, or modified in the base model. These elements can be chosen for a variety of reasons including, but not limited to, intuitions for reducing the computational effort.

3. Create models from those definitions. When a modification is applied to the base model, a new problem definition is created. Definitions should be created so that two definitions are identical except for the inclusion of a simplification in one and not the other. There should be several definitions, so that a range of models with different levels of abstraction are created. A mathematical model should then be created for each problem definition.

4. Evaluate the models on solvability, the time it takes to deliver a solution. One measure of solvability is the percentage of problem instances that can be solved within a pre-determined time limit. The problem instances should be representative of the real application, so they should be created using data from the application or be approximated so that the size and structure of the instances are a close match to the application. The goal is to find a definition that can be solved in an amount of time that is reasonable for the needs of the application.

5. Evaluate the models on quality, the usefulness of the solutions in the real world. The objective functions resultant from different definitions cannot be directly compared because the problems being solved are not mathematically equivalent. However, the solutions of the models can be compared in the real world or in a simulation. Quality measures and a definition of an acceptable solution should be determined together with the end users of the system. One application may require an optimal solution while in another the users may be satisfied with any solution that is better than what they can currently accomplish.

6. Iterate as necessary. Collect the experimental results from the previous steps and perform an audit of the problem definitions and models. At this point, there may surface a model that meets all the user requirements and performs well in both solvability and quality. In such a situation, that definition and model can be chosen to move forward with. It is also possible that a requirement is missing or that none of the models perform well in both

solvability and quality. In that case, return to step one and experiment with a new base model.

In the following sections, the above method is demonstrated, using the chemotherapy outpatient scheduling problem at the Odette Cancer Centre.

4.2 Base Model Formulation

As the first step, a base definition of the chemotherapy outpatient scheduling problem, *Base*, is defined. Through interviews with system users, site observations, and an analysis of current scheduling practice, we determine what elements might influence the scheduling of chemotherapy appointments. We then create a definition that encompasses those elements. As described above, every definition is paired with a model in order to proceed with evaluation. The following is a description of our base model.

The formulation was developed in constraint programming using COMET 2.1.0 (Dynadec Decision Technologies, Providence RI) and uses constraints and search techniques built into the software. Using constraint programming as the modelling technique made it natural to express the constraints, resulting in a model that is easy to understand. Constraint programming was also advantageous because COMET has built-in constraints and search techniques designed specifically for scheduling applications. Additionally, constraint programming makes it simple to experiment with different problem definitions because for our problem, a model for a different definition can be created through the addition or deletion of a global constraint.

The notation for the base model is as follows:

- R: set of appointment requests.
- F: set of already scheduled appointments. In many cases, we need to schedule a set of appointments in the presence of already scheduled appointments.
- $\forall i = 1,2,3,4; \forall f \in F; S_{if}^*$: an already assigned start time to stage i of already scheduled appointment f. The start times of the appointments in set F have already been fixed when they are passed into the model.

- *G*: the complete set of appointment requests and already scheduled appointments, defined as $R \cup F$. When a constraint applies to both patient requests and already scheduled appointments, it applies to each appointment *g* from the set *G*.
- *End*: a dummy appointment request constrained to be the last appointment of the day.
- S_{End}: the start time of *End.*
- *C*: set of chairs – a maximum of 29 chairs and beds. One abstraction in the model is that chairs and beds are represented as identical resources. There is currently no data available on which patients will require a chair or a bed.
- *N3*: set of nurses available for stage three, the first stage of treatment – there are 13 or 15 nurses; the total capacity is |N3|. Each test day used for experiments contains the exact number of nurses that worked in the clinic on that day.
- *N4*: set of nurses for stage four – each 0.25 of a nurse is a resource for a total of 52 or 60 units; the total capacity is |N4|. *N3* and *N4* represent the nurses in different roles, setting up and monitoring respectively. Each of the 13 or 15 nurses can monitor up to four patients during stage four.
- *T*: set of times ranging from zero through 40. Each time is a 15-minute timeslot for a maximum of ten hours of daily treatment time.
- *DR:* set of servers for drug preparation – there are five servers at all times; the total capacity is |DR|.
- $\forall i = 1,2,3,4; \forall r \in R;$ D_{ir}: duration of stage *i* for patient request *r*. Stage one represents chart review and has a duration of one time slot. Stage two represents drug preparation and has duration determined by the chemotherapy regimen. Stage three represents setting up the patient for treatment and has duration determined by the chemotherapy regimen. Stage four represents delivery of drugs and monitoring the patient during treatment. Stage four has duration of the complete treatment length determined by the regimen including both the setting up and drug delivery portions of the treatment. Although non-intuitive,

the stage four duration must span both parts of the treatment in order to ensure nursing capacity is respected (see the description of constraint 8).

- $\forall r \in R$; B_r: the ready time, when patient request r can begin the first stage. If a patient has no previous appointments on the day of treatment, the ready time is zero. If a patient does have previous appointments, his/her ready time will be the end time of the previous appointments.

- l: a constant. It represents a maximum time allowed between stages one and two, as well as between stages two and three. This time will ensure that any single patient does not wait an excessive amount of time.

For each patient request $r \in R$, we have three decision variables: the start times of the chart review, drug preparation, and treatment stages. The final stage of treatment is scheduled to start at the same time as stage three.

- $\forall i = 1,2,3;\ \forall r \in R,\ S_{ir}$: The start time of stage i for patient request r. It takes a value from 0 through |T|-1.

At the Odette Cancer Centre, nurses are entitled to 1.25 hours of breaks during the day, comprised of two 15 minute breaks and one 45 minute lunch-break. In order to accommodate these breaks in the model, the number of nurses available at different times of the day is restricted. A morning 15-minute break is accommodated by limiting the nurses available from 9 am to 11 am. We first calculate the total break time required by all nurses and then we divide the break time over the two hours and reduce the number of nurses available during that time. For example, in the case where there are 13 nurses available, they require a total of 195 minutes of a morning break. By reducing the number of nurses available during those two hours by 1.5, we are able to ensure that each nurse receives a 15-minute break. A 45-minute lunch-break is accommodated by limiting the nurses available from 11 am to 2 pm. From 3 pm to 5 pm, the number of nurses is limited to accommodate a 15-minute break in the afternoon.

The constraint programming model of the *Base* definition is as follows:

min S_{End}

Subject to:

disjunctive (S_{1g}, D_{1g}),	$\forall g \in G$	(1)
cumulative $(S_{2g}, D_{2g}, 1, \|DR\|)$,	$\forall g \in G$	(2)
cumulative $(S_{4g}, D_{4g}, 1, \|C\|)$,	$\forall g \in G$	(3)
cumulative $(S_{3g}, D_{3g}, 1, \|N3\|)$,	$\forall g \in G$	(4)
cumulative $(S_{4g}, D_{4g}, 1, \|N4\|)$,	$\forall g \in G$	(5)
$S_{1r} \geq B_r$,	$\forall r \in R$	(6)
$S_{ir} \geq S_{i-1r} + D_{i-1r}$,	$\forall r \in R, \forall i = 2,3$	(7)
$S_{4r} = S_{3r}$,	$\forall r \in R$	(8)
$S_{End} \leq \|T\|$,		(9)
$S_{4g} + D_{4g} \leq S_{End}$	$\forall g \in G$	(10)
$S_{2r} \leq S_{1r} + D_{1r} + l$,	$\forall r \in R$	(11)
$S_{3r} \leq S_{2r} + D_{2r} + l$,	$\forall r \in R$	(12)
$S_{if} = S_{if}^{*}$,	$\forall f \in F, \forall i, i = 1,2,3,4$	(13)

The objective of the model is to minimize makespan, or the start time of *End*. However, the real application has two objectives: maximizing resource utilization and minimizing patient wait times. One abstraction in this model is to use a single objective to reduce complexity. Of the two objectives of the problem, maximizing resource utilization in order to service more patients by way of minimizing the makespan is chosen. Increasing the number of patients that can be serviced is the key objective for the chemotherapy centre because their demand is growing. Also, logically, by minimizing makespan, wait times will tend to be lower, indirectly accounting for the objective of minimizing wait times. However, minimizing makespan does not guarantee a reduction in wait times, so wait times are evaluated when the definitions are tested for quality.

Constraint (1) ensures only one patient receives chart review at any time. Each stage-one activity is constrained to require one chart review nurse. The disjunctive constraint (Beldiceanu and Demassey 2010) is a global constraint that ensures activities do not overlap.

Constraints (2) through (5) are cumulative constraints (Beldiceanu and Demassey 2010), global constraints that ensure that the capacity of a discrete resource is respected. These cumulative constraints represent the pharmacy technicians, nurses, and chairs. In constraint (5) each nurse is really 0.25 of a nurse. Together with constraints (4) and (8), constraint (5) ensures that the capacity of the nurses is respected and, in particular, a nurse cannot be monitoring four patients while setting up a fifth.

Constraint (6) ensures the start time for each patient is greater than or equal to his/her ready time.

Constraint (7) ensures that stages two and three cannot start before the end time of the preceding stage.

Constraint (8) ensures that stage four starts at the same time as stage three. The stage four activity spans both parts of the same treatment and the portion of the stage four activity representing stage three must occur at the exact same time as the actual stage three activity. This somewhat peculiar modeling approach is to account for the two roles of a nurse: setting-up a patient (stage three) and monitoring (stage four). In having stages three and four start at the same time, a value of 0.25 of a nurse for the monitoring component of treatment is required throughout the entire treatment (constraint (5)) and an additional value of one nurse is required during stage three (constraint (4)). As a result, 1.25 of a nurse is required during stage three. These constraints ensure that a nurse will not be monitoring four patients, using the maximum capacity of *N4*, at the same time as setting up another, using the capacity of *N3*.

Constraint (9) ensures no patient finishes treatment after the end of the working day.

Constraint (10) ensures *End* does not start until all other activities have completed.

There is a maximum time, $l = 2$ timeslots, the equivalent of 30 minutes, between stages one and two (constraint (11)). The same limit is imposed between stages two and three (constraint (12)). These constraints ensure that patients will not have to wait for extra time between stages, even if such a delay would create a more efficient schedule.

For each already scheduled appointment, $f \in F$, there is an additional constraint added to the optimization model (constraint (13)) that ensures the start time of each stage of that appointment will remain unchanged.

4.3 Modifications and Corresponding Problem Definitions

In order to investigate multiple problem definitions, we chose three modifications to the *Base* model to experiment with. In all cases, we explore simplifying the problem by removing resources that may not be influential. Each modification results in a different problem definition that is formulated into a constraint model. All the models have the objective to minimize S_{End} and are defined explicitly in Appendix A.

1. Chart Review Modification – The chart review stage is omitted. Removing a stage should reduce computational effort. Chart review is not included in how the process is currently implemented at the hospital, so excluding it should mean the new solution is no worse than current practice. If blood work is approved, the chart review stage is negligible because it has duration of less than a timeslot. Model *CRS* (for "chart review simplification") corresponds to the problem definition that only includes the chart review modification.

2. The Pharmacy Modification – The pharmacy stage is omitted. Removing the stage should reduce computational effort. Pharmacy is not directly included in how the process is currently scheduled at the hospital. Model *CRS-PS* corresponds to the model that includes the chart review modification and the pharmacy modification.

3. The Ramp Up Modification – Rather than completely omitting the pharmacy stage, it is implicitly represented by ramping up the chair capacity. At the start of the day, there are 21 chairs available. After an hour, 24 chairs are available. After two hours, all 29 chairs are available. This modification is included to assess the value of including pharmacy in this way and because it is the current practice at the Odette Cancer Centre. Model *CRS-RUS* corresponds to the model that includes the chart review modification and the ramp up modification.

4.4 MIP Model

Researchers using constraint programming claim that it works better at solving scheduling problems than MIP and constraint programming is acknowledged as a technique well suited for scheduling problems (Bartak et al. 2010, Beck and Fox 1998). However, MIP is a more dominant and widespread technology. Traditionally, operations research based studies apply MIP to flexible flow shop or job shop problems (Demir and Kursat Isleyen 2012, Fattahi et al. 2007). Therefore, it makes sense to evaluate the solvability of the constraint model compared to a MIP model.

In this section, mixed integer programming is used to model the chemotherapy outpatient scheduling problem. The MIP model created has the chart review stage omitted and so is similar, but not equivalent (see below), to the constraint model *CRS*.

Based on the sensitivity analysis (see Section 4.8.3) showing that nurse are not constraining resources and the difficulties in modeling the MIP, we make the assumption that the second stage of treatment does not require a nurse. We assume that there will be enough nurses to monitor patients and only constrain the first part of treatment to require a nurse for set up. Nurses can monitor up to four patients at a time and as such have a capacity of 29397 minutes each day. The largest amount of treatment minutes in our data is 11190, which is 38% of the nurse capacity (see Table 4.1 in Section 4.7 for problem data).

The MIP is a time-indexed model containing the binary decision variable x_{irt}, which is set to one if the start time of stage *i* of request *r* is time *t*. The model can be written as follows:

$$\min S_{End} \quad (14)$$

subject to:

$$\sum_{t \in T} x_{irt} = 1, \quad \forall i = 2,3,4;\ \forall r \in R \quad (15)$$

$$\sum_{t \in T} (t + D_{4r}) \cdot x_{4rt} \leq S_{End}, \quad \forall r \in R \quad (16)$$

$$\sum_{r \in R} \sum_{t' \in T_{2rt}} x_{2rt'} \leq |DR|, \quad \forall t \in T, where\ T_{2rt} = \{t - D_{2r} + 1, \ldots, t\} \quad (17)$$

$$\sum_{r \in R} \sum_{t' \in T_{3rt}} x_{3rt'} \leq |N3|, \quad \forall t \in T, where\ T_{3rt} = \{t - D_{3r} + 1, \ldots, t\} \quad (18)$$

$$\sum_{r \in R} \sum_{t' \in T_{4rt}} x_{4rt'} \leq |C|, \quad \forall t \in T, where\ T_{4rt} = \{t - D_{4r} + 1, \ldots, t\} \quad (19)$$

$$\sum_{t \in T} (t + D_{2r}) \cdot x_{2rt} \leq \sum_{t \in T} t \cdot x_{3rt}, \quad \forall r \in R \quad (20)$$

$$\sum_{t \in T} t \cdot x_{3rt} = \sum_{t \in T} t \cdot x_{4rt}, \quad \forall r \in R \quad (21)$$

$$\sum_{t \in T} t \cdot x_{1rt} \geq Br, \quad \forall r \in R \quad (22)$$

$$\sum_{t \in T} (t + D_{2r}) \cdot x_{2rt} + l \geq \sum_{t \in T} t \cdot x_{3rt}, \quad \forall r \in R \quad (23)$$

$$x_{irt} \in \{0,1\} \quad \forall i = 2,3,4;\ \forall r \in R, \forall t \in T \quad (24)$$

The objective function is to minimize the makespan. Constraint (15) ensures that each activity starts exactly once. Constraint (16) ensures that the makespan is at least the latest end time of the final activity. Constraints (17), (18), and (19) respectively ensure that the capacity of the pharmacy technicians (stage two), nurses (stage three), and chairs (stage 4) is respected.

Constraint (20) is the precedence constraint between the second and third stages. Constraint (21) says that the start time of the second stage of treatment must equal the start time of the first stage of treatment. Constraint (22) says that no appointment can start before its ready time. Constraint (23) says that there is a maximum time, l, allowed between the start times of the drug preparation and treatment stages. Constraint (24) ensures x_{irt} is a binary variable.

4.5 Simulation Model

The solutions to different problem definitions cannot be directly compared because they are not mathematically equivalent: for example the makespan of the *Base* model is not comparable to the chart simplification model, *CRS*, because the latter has fewer activities. When the schedules from the different problems are tested in a simulation of the world, the results can be directly compared because all activities are included.

The simulation model was developed using SIMUL8 2009 Student Edition (Simul8 Corporation, Boston) discrete event simulation software. It encompasses patient flow from arrival through completion of treatment. The constraint models provide as output a schedule of start times for each stage of the chemotherapy process included in that model and the simulation uses those start times to generate patient arrivals. For models that only take into account the treatment stages, *CRS-PS* and *CRS-RUS*, patients arrive to the simulation 60 minutes prior to the assigned start of treatment. For models that take into account additional stages of the process, patients arrive at the earliest start time assigned to them by the model. For *Base*, patients arrive at the scheduled start time of the chart review stage. For *CRS*, patients arrive at the scheduled start time of the drug preparation stage. Each patient's durations are known based on the regimen being treated.

In the simulation, queuing follows a first-come-first-serve rule, as is done at the Odette Cancer Centre. This set up is different from, and more realistic than, the constraint models, where all appointments occur at their scheduled start time. Patients are seen in the order that they enter the unit and are taken for treatment in the order that their drugs become available. Nurses are not assigned to specific patients. Whichever nurse is available services the first patient in the queue. The simulated chemotherapy unit operates from 8:00 A.M. Each operating day is considered independently. The simulation was run until all the appointments scheduled for a given day were completed. The quality metrics used are average patient wait time and maximum patient wait time. Wait time is measured between the start of the chart review and the completion of treatment. Lower wait times equates to higher quality.

4.6 Model Validation

The constraint, MIP, and simulation models presented above were validated following the process presented by Macal (2005). Input and output data were validated by checking the agreement of the data with the original data sources and ensuring all the appointments that occurred at the Odette Cancer Centre also occurred in the generated schedules and the durations of the various stages matched the durations in the original data. Additionally, output data were cross-checked with a third party report conducted at the Odette Cancer Centre by Michael Gannon in 2007. Requirements validation was conducted by several parties independently to ensure all the constraints worked as expected. Face validation was conducted by subject matter experts at the Odette Cancer Centre to ensure the output of the models was realistic given the known assumptions present in the models in question.

In addition to the above validation processes, process validation was also conducted on the simulation model. Since the data provided by the Odette Cancer Centre only included start times for treatments that occurred, exact patient wait times were unknown. Therefore, students were recruited to collect time stamp data for patient arrivals and wait times for one week. The simulation model was then validated using two instances of data collected by the students by comparing the model wait times to the collected data. Although the wait times of individual appointments did not always match, the average wait time of the model (36.5 minutes) was within 15% of the actual average wait time (42 minutes) and was deemed acceptable by the subject matter experts at the Odette Cancer Centre.

4.7 Experimental Design

To choose a problem definition and study the impact of the resources and problem characteristics, four sets of experiments were conducted: (1) solvability experiments, (2) quality experiments, (3) constraint programming compared to MIP, and (4) sensitivity analysis.

Table 4.1. Experimental data

Test Set	Number of Appointments	Total Treatment Minutes	Nurses Available	Estimated Average Patient Wait Time (min)	Estimated Maximum Patient Wait Time (min)
Day 1	63	6210	13	60	180
Day 2	83	9435	13	60	180
Day 3	81	11760	15	60	180
Day 4	99	10725	15	60	180
Day 5	90	11190	13	60	180

The experiments are run using five problem instances, each corresponding to an actual day at the Odette Cancer Centre. The five sets used are March 1st through 5th, 2010. The data is provided in Appendix B. The direct nursing requirement and drug preparation time was determined by matching each patient's regimen to a regimen database provided by Cancer Care Ontario. The duration of the chart review stage was assumed to be 15 minutes, one timeslot, for every patient. Table 4.1 provides the number of appointments, total treatment minutes, nurses available, estimated average patient wait time, and estimated maximum patient wait time for each data set. Wait times were estimated by subject matter experts at the cancer centre.

To evaluate solvability, each data set is solved using each constraint model and the time to solve and percentage of instances solved to optimality are measured. The intention is that the chosen model will be incorporated into the current scheduling software and will be used in real time. When an appointment request arrives, the model may need to find an optimal placement for that appointment. The model must be able to find a solution relatively quickly. To test the models in that regard, the time to solve, up to a maximum of 3600 seconds is measured. If the model has not found and proved an optimal solution within 3600 seconds, the best solution found is taken as the resultant solution. Within the dynamic framework, the makespan resulting from the optimization model affects the overall solution. To test how well the models will perform in that regard, the percentage of instances solved to optimality is measured.

To evaluate quality, the schedules resulting from the constraint models were run through the simulation and the average and maximum wait times are measured. Although minimizing makespan will generally result in lower wait time for patients and a more efficient schedule, we have to check how well the solutions work in a more detailed representation of the real world as embodied in the simulation model. The patient wait time provides a measure of how well the model solutions work in the real world because if the model works well and minimizes makespan, patient wait times will also be lower. Wait time is defined as the time that a patient is in the system (between the start of the chart review stage and the completion of treatment), but not being serviced at any of the four stages.

Based on the results of the experiments on solvability and quality, one of the constraint models will be chosen and compared to the MIP model. First, the five problem instances will be solved by both the chosen constraint model and the MIP model. Then, six new test sets will be solved using both the models. Assuming the same number of resources, the number of patient appointments was reduced from 63 to 50, 40, 30, 20, 10, and 5. These problem sets are referred to as mip_50, mip_40, mip_30, mip_20, mip_10, and mip_5. Each model was given a maximum run-time of 3600 seconds.

The final set of experiments is a sensitivity analysis. We adjust various parameters of the model to assess the effect on model performance. Four parameters are studied: (1) problem size, (2) pharmacy technician capacity, (3) nurse capacity, and (4) chair capacity. Using the chosen constraint model and Test Day 1, we study the effect of makespan and CPU time when we adjust the various parameters. As above, the maximum run-time is 3600 seconds. If the model cannot find a feasible solution in that time, it will fail. The maximum makespan allowed is 40 because this corresponds to a 10-hour day, the maximum time the centre is willing to work, including overtime.

All the experiments were conducted on a Pentium Dual-Core CPU computer E5200 @ 2.50 GHz and 3.25 GB of RAM running 32 bit Windows XP.

4.8 Experimental Results and Discussion

In this section, we summarize the experimental results from the four sets of experiments. Please refer to Appendix B for the complete set of experimental results.

4.8.1 Solvability and Quality Experiments

Since the intention is for the model to be used in real time, it is important to note that for the majority of the runs, a feasible solution is found very quickly and if an optimal solution is found, it is found early on the process. A feasible solution was found 100% of the time, and a feasible solution was found within one second 96.67 % of the time.

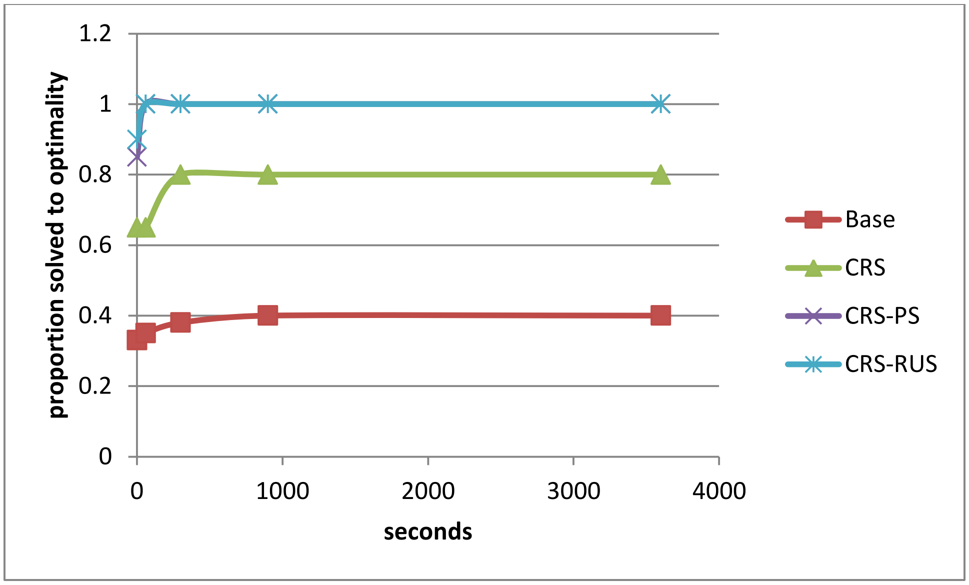


Figure 4.1. The proportion of instances solved to optimality versus time (s)

Figure 4.1 shows the proportion of instances of all models that solved to optimality at different time intervals.

Table 4.2 summarizes the results. The average results are used to create a rating scale that can be used to qualitatively compare the models as presented in Table 4.3. The description refers to the percentage relative to the results of all models. For example, a time to solve value that is worse than 75% of all the time to solve values warrants a rating of poor. In our experiments, 75% of all the time to solve values are better (lower) than 446.41 seconds. Any model that takes longer than 446.41 seconds to solve is given a rating of poor for that metric. In the table, "time to solve" is the CPU time in seconds, "optimality" is the percent of instances that are solved to optimality, "average" and "maximum wait" are the number of minutes between the start of chart review and the completion of treatment that he/she is not involved in any of the four process stages.

Table 4.2. Average solvability and quality results across all models

Measure	**Value**
Average time to solve (s)	864.76
Percent solved to optimality (%)	76.67
Average wait (min)	78.19
Maximum wait (min)	138.22

Table 4.3. Rating scale values

Rating	Description	Time to Solve Value (s)	Optimality Value (%)	Average Wait (min)	Maximum Wait (min)
Poor	Worse than 75%	> 446.41	< 50	> 200	> 309
Average	25 - 75%	0.063 - 446.41	50 – 70	11 – 200	23 – 309
Very Good	Better than 25%	< 0.063	> 70	< 11	< 23

Additional ratings of *Fair* and *Good* are used where the two solvability or quality measures fall into different rating groups. *Fair* refers to a mix of *Poor* and *Average* and *Good* refers to a mix of *Average* and *Very Good*. A rating of *Average* is given if there is a mix of *Poor* and *Very Good*.

Table 4.4 shows the results for testing performed on four different definitions of the same real-world problem. The values are the results for the particular model averaged over all five problem instances. The tests show the trade-off between models that have high solvability, which can be solved in a reasonable amount of time, and models that have high quality, yielding solutions that work well in the simulation.

Based on the values and ratings, *CRS* is the best model. The most complete definition, *Base*, was hard to solve. *CRS-PS*, the definition easiest to solve, resulted in low quality solutions. *CRS* rated well in both solvability and quality.

Table 4.4. Model comparison

	Solvability			**Quality**		
Model	**CPU time (s)**	**Optimality (%)**	**Rating**	**Average Wait (min)**	**Max Wait (min)**	**Rating**
Base	2162.28	40	Poor	11.3	31	Average
CRS	720.13	80	Average	9.0	22	Very Good
CRS-RUS	89.44	100	Good	173.8	304	Average
CRS-PS	0.03	100	Very good	201.2	310	Poor

4.8.2 Comparison with MIP

The MIP model was used to solve the five test sets presented in Section 4.7 with a maximum run-time of 3600 seconds. It was not able to find any solutions, even feasible solutions, in the allotted time. Therefore, we present the results to the six additional problem sets developed, mip_5 through mip_50. Table 4.5 presents the comparison between the MIP model and the constraint model *CRS*. If no solution was found within 3600 seconds, the CPU time is denoted by the notation '-'.

The constraint model performs considerably better than the MIP model. This result is consistent with the results found in Ku (2013) for the flexible job shop problem. A likely explanation for these results is the known weakness of the time-indexed model that requires a binary variable for each activity and time-point on each resource. Such models are known to scale poorly (Ku, 2013) however, to our knowledge, there are no other MIP models that can represent non-unary capacity resources such as the nurses and chairs. Moving to a MIP model that represented individual nurses and chairs and so would admit a disjunctive or position-based MIP model would substantially increase the dimension (and symmetry) of the problem as patients would have to be assigned to individual nurses and chairs.

These results indicate that the constraint models work better than a standard MIP model and that we are justified in using constraint programming in this research.

Table 4.5. Comparison of MIP and *CRS* models

Test Set	MIP		CRS	
	Time to first feasible solution (s)	**Time to optimal solution (s)**	**Time to first feasible solution (s)**	**Time to optimal solution (s)**
mip_5	0.36	1.45	0.04	0.06
mip_10	21.03	39.39	0.12	9.00
mip_20	123.86	238.60	11.43	20.87
mip_30	1729.42	3305.04	33.47	49.20
mip_40	3138.80	-	45.92	59.64
mip_50	-	-	46.00	78.20

4.8.3 Sensitivity Analysis

When *CRS* is used to solve Test Day 1, it achieves a makespan of 24 15-minute slots in 0.047 seconds of CPU time. With each parameter we study, we examine the effect on the makespan and CPU time as we adjust the parameters.

Test Day 1 has 63 appointments and a total of 6210 appointment minutes. We gradually increase the number of appointments by 5 appointments, each 60 minutes in length. Figure 4.2 shows the effect of problem size. Until we reach 78 appointments, there is no effect on the makespan or the time to find and prove optimality. From 78 to 113 appointments the makespan linearly increases and the model takes the full 3600 seconds to be solved. Once we reach 113 appointments, the model is quickly able to prove that there is no feasible solution.

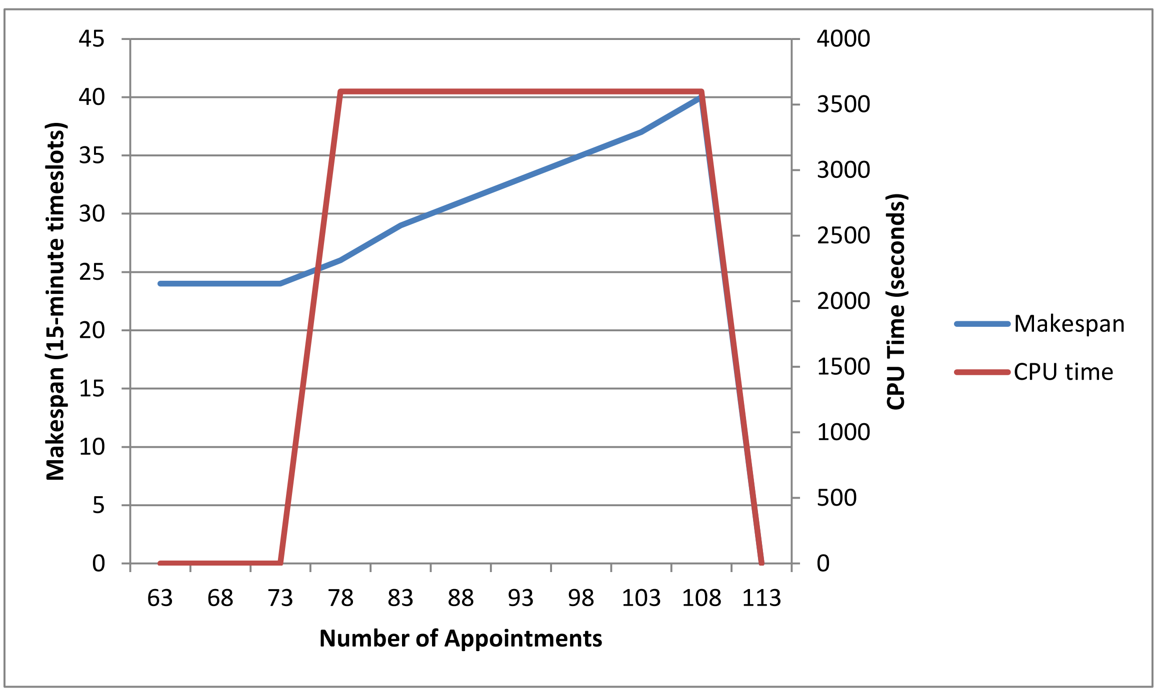


Figure 4.2. The effect on makespan and CPU time as the number of appointments is increased

Test Day 1 has five pharmacy technicians. Figure 4.3 shows the effect of gradually increasing or decreasing the number of technicians in step sizes of one. When the pharmacy technicians are reduced to four, although an optimal solution is found very quickly, we see an increase in the makespan, showing that the number of pharmacy servers is a constraining resource. When the number of pharmacy technicians is further reduced to three, the model takes the full 3600 seconds to solve and the makespan increases again. When the number of pharmacy technicians is reduced to two, the model is quickly able to prove that there is no feasible solution. On the other hand, the makespan and time to solve cannot be improved by adding pharmacy servers. By examining the problem set, we see that the makespan for Test Day 1 is the minimum possible makespan because there is a treatment that requires 24 timeslots.

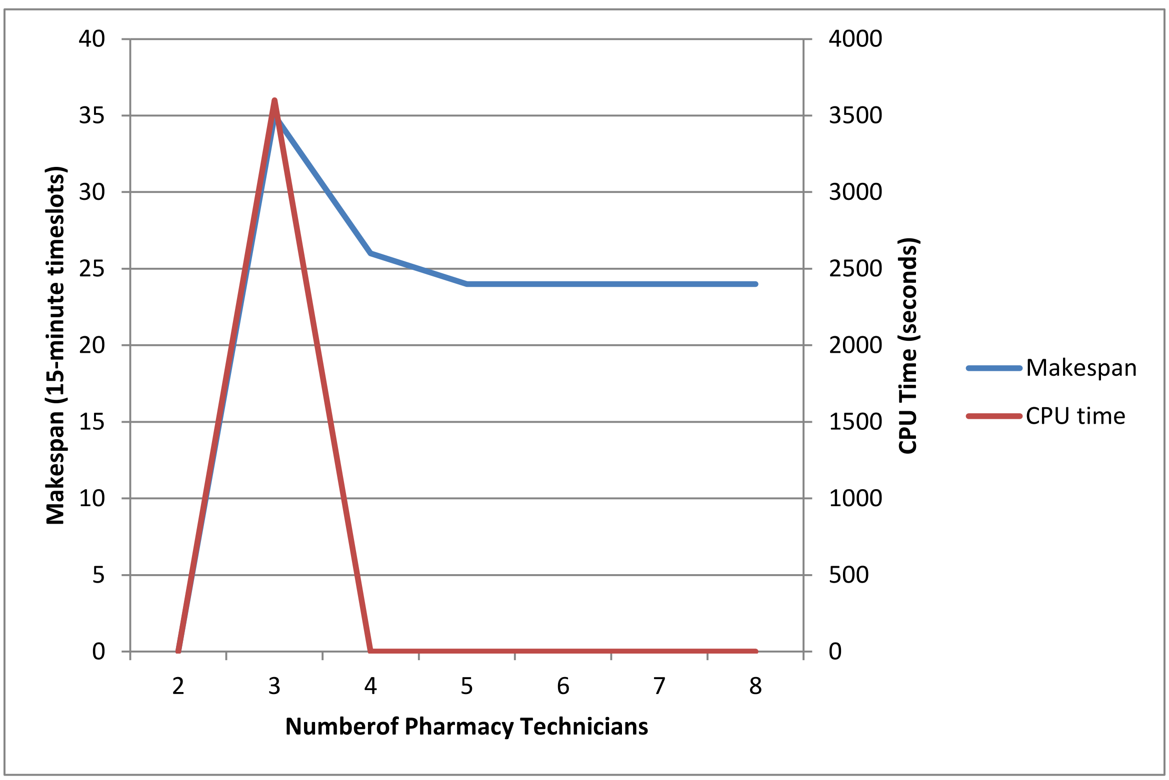


Figure 4.3. The effect on makespan and CPU time as the number of pharmacy technicians is increased and decreased

Test Day 1 has 13 nurses available. Figure 4.4 shows the effect of nurse capacity by decreasing the number of nurses available one by one. There is no effect in the makespan or time to solve until the number of nurses is reduced to nine. At nine nurses, the model is still able to find and prove optimality very quickly, but the makespan begins to increase. At seven nurses, the model takes the full 3600 seconds and the makespan continues to increase. Once we reach six nurses, the model is quickly able to prove that there is no feasible solution.

Test Day 1 has 29 chairs available. Figure 4.5 shows the effect of decreasing the number of chairs in steps of one. There is no effect on the time to solve or the makespan until the chair capacity is reduced to 18. The model can still find and prove optimality very quickly, but the makespan begins to increase. When the number of chairs is reduced to 16, the model takes the full 3600 seconds to solve and the makespan continues to increase. When the number of chairs is reduced to 12, the model can quickly prove that there is no feasible solution.

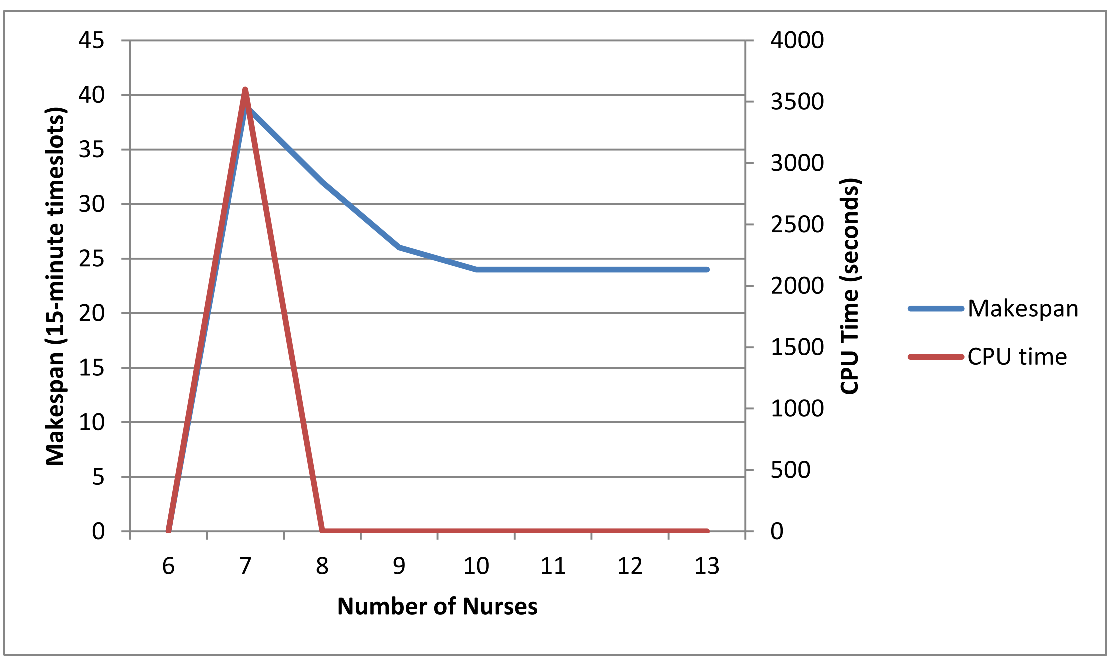


Figure 4.4. The effect on makespan and CPU time as the number of nurses is decreased

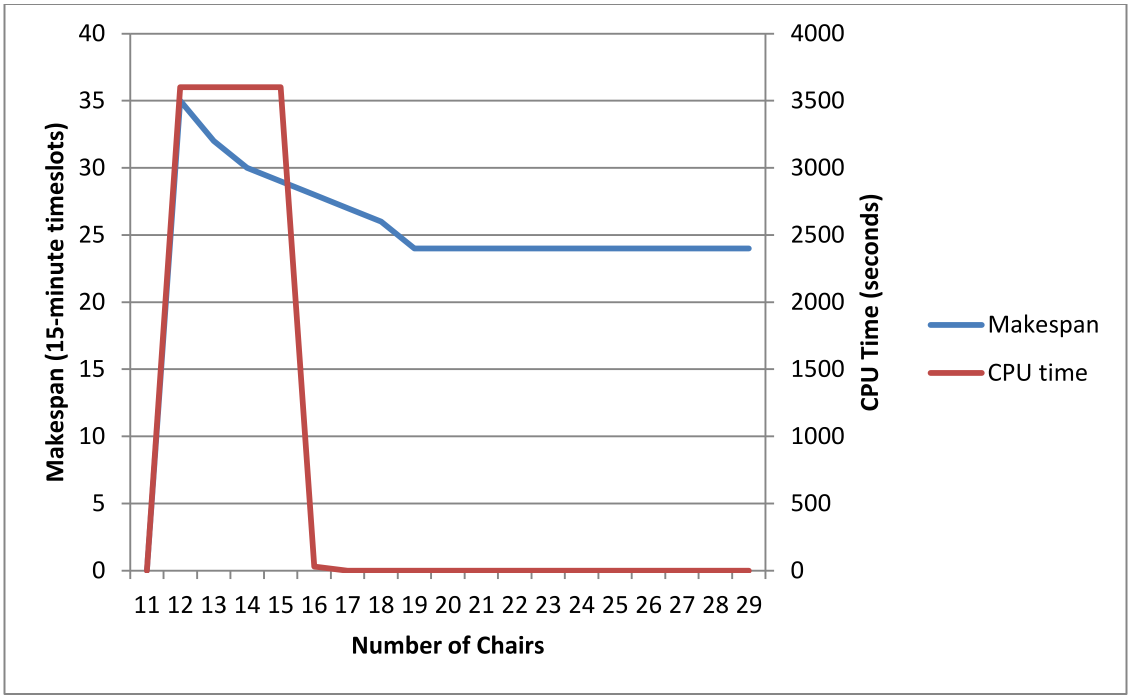


Figure 4.5. The effect on makespan and CPU time as the number of chairs is decreased

Of the parameters studied, the number of pharmacy servers is the most constraining resource. If the number is reduced by one, there is an effect on the makespan of the problem. This is not true for the number of appointments, nurses, or chairs.

It is obvious from the problem structure and the results of the sensitivity analysis that the drug preparation stage and the capacity of pharmacy technicians are bottlenecks in the chemotherapy process at the Odette Cancer Centre. However, current scheduling practice does not directly consider the drug preparation stage or the capacity of the pharmacy technicians. This result also explains why schedules resulting from *CRS-PS* and *CRS-RUS*, the models that do not contain constraints representing the drug preparation stage and the pharmacy technicians, have considerably worse performance when they are evaluated for quality using the simulation model (see Table 4.4).

4.9 Conclusion

This chapter highlighted the need for in-depth study of the problem definition stage when creating applications and provided a methodology for choosing and evaluating problem definitions of complex real-world problems. In the chemotherapy problem there was a trade-off between solvability and quality and testing various problem definitions allowed us to study the impact of representing the different problem resources.

The methodology investigated here is a contribution to operations research. The process of creating and evaluating mathematical models of the real world is not typically present in operations research literature. Although this methodology was only tested on one case, future research could test it in different problem domains or compare it to alternative methodologies.

This chapter also makes a contribution by applying constraint programming to scheduling problems and to health care scheduling applications by studying the deterministic chemotherapy outpatient scheduling problem.

Chapter 5

[redacted]Dynamic Template Scheduling

The chemotherapy outpatient scheduling problem suffers from two main sources of uncertainty: (1) real-time uncertainty, which results from having to schedule appointments as they arrive, without full knowledge of what appointment requests will arrive in the future and (2) uncertainty due to cancellations and last minute additions to the schedule.

In this chapter, dynamic template scheduling, a method for combining proactive and reactive optimization for complex scheduling problems, is described and demonstrated on the chemotherapy outpatient scheduling problem at the Odette Cancer Centre. Using historical data of the problem domain, an empirical distribution of appointment types is produced that provides the expected number of appointments of each length in a day. *CRS*, the constraint model for the deterministic chemotherapy problem (see Chapter 4), is then used to create a solution that optimizes the makespan for the expected number and mix of appointments. We call this solution a *proactive template*. It forms the predictive schedule that is then used and revised to solve the real-time problem.

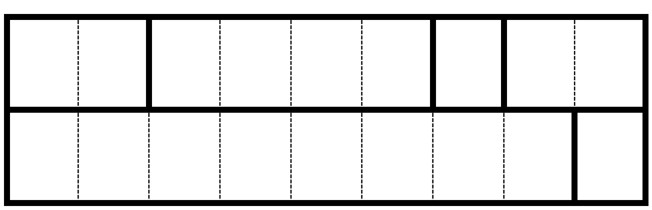
As stated in Chapter 4, there are three sets of appointments: (1) *R* is the set of appointment requests that arrive in real time and must be scheduled, (2) *F* is the set of already scheduled appointments, and (3) *E* is the sample of expected appointments taken from a distribution created using historical appointment patterns as described in detail in Section 5.2. When *CRS* is given inputs *E* and *F*, it produces an optimal template, *TS*, that ensures that any already scheduled appointments do not change. The appointments in set *E* become open timeslots that will be used to schedule subsequent appointments requests from the set *R* as they arrive. When a conflict occurs, i.e., there is no open timeslot that can be used to meet a request, the template is dynamically updated using the deterministic optimization model to accommodate the request.

The remainder of Chapter 5 is organized as follows. Section 5.1 presents an example of dynamic template scheduling. Section 5.2 describes the empirical distribution of appointments and the statistical analysis conducted. Section 5.3 describes the stochastic chemotherapy outpatient scheduling problem and the dynamic template scheduling algorithm designed to solve the problem where appointments are scheduled in real time. Section 5.4 provides a description of the experimental design. Section 5.5 gives the experimental results and discussion section 5.6 concludes.

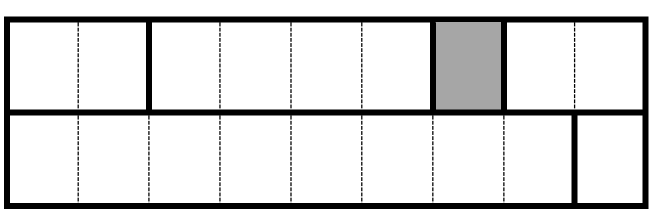
5.1 An Example

The following example illustrates the dynamic template scheduling algorithm. In this example, there are two chairs available to treat patients and the expected number of appointments, q, is six. First, a sample of size six is taken from the empirical distribution to get E, the set of expected appointments used to create the template. Note that E is not the set of real appointment requests but a sample of possible appointments. For this example, $E = \langle e_1=15,\ e_2=15,\ e_3=30,\ e_4=30,\ e_5=60,\ e_6=120 \rangle$, where treatment durations for each appointment in E is given in minutes. That set of appointment requests is then passed to the optimization model, *CRS*, which schedules these appointments optimally. The schedule is then interpreted as a set of open timeslots that can be used to schedule the actual requests that arrive.

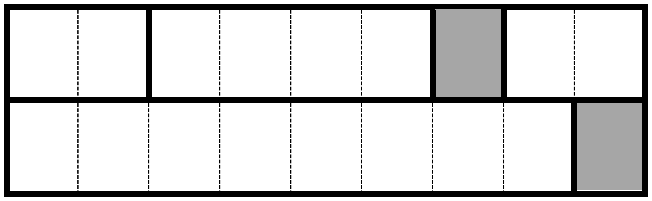
In this simple example, the optimal set of timeslots to minimize makespan $TS = [\{30, 60, 15, 30\}, \{120, 15\}]$, where the first chair has open slots of 30 minutes, 60 minutes, 15 minutes, and 30 minutes, and the second chair has open slots of 120 minutes and 15 minutes. The makespan is 135 minutes. Using bold lines outline the timeslots and normal lines to separate each 15-minute time period, we can represent *TS* as follows



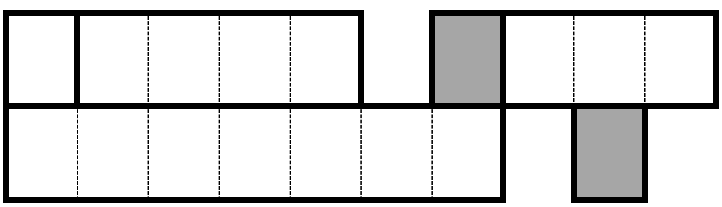
The actual requests for appointments from the set *R* of requests arrive one at a time and we schedule them into the template as they arrive. When an appointment is scheduled into *TS*, we shade the corresponding timeslot in our diagram. The first request to arrive, r_1, has a duration of 15 minutes. *TS* becomes



The assigned appointment, r_1, is added to *F* and becomes f_1. The number of appointments expected is reduced by one, so $q = 5$. The next request to arrive, r_2, is also 15 minutes. *TS* becomes

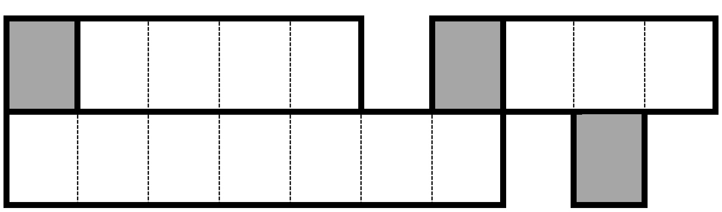


The assigned appointment is added also to *F*. The third request to arrive, r_3, is also 15 minutes. There is no open 15-minute timeslot in the template, so the template must be updated. Since two appointments have already been scheduled and one has arrived to be scheduled, $q = 3$, i.e., its original value minus the requests that have arrived (6 - 3). A new sample of expected appointments of size three is taken from the distribution and added to the to-be-scheduled request to get $E = < e_1 = r_3 = 15, e_2 = 45, e_3 = 60, e_4 = 105 >$. This time, the requests have to be scheduled around the already scheduled appointments, *F*. The new optimal schedule is *TS* = [{15, 60, **15,** 45},{105, **15**}], where the bold numbers are already scheduled appointments from *F* and the other numbers are open timeslots in the template. *TS* can be represented by

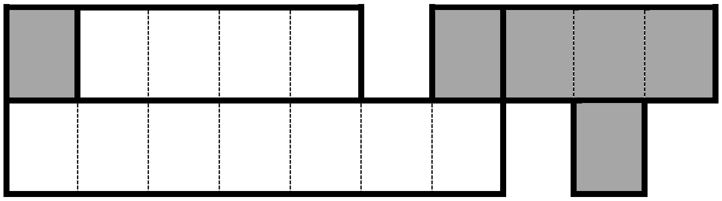


Due to the already scheduled appointments and the expected appointment durations sampled, the resulting template has empty places, such as a 15-minute "hole" left in the schedule after the 15-minute and 60-minute timeslots on the first chair. No currently existing or expected appointments can be scheduled into these times. The optimization model only creates slots for appointments in the sample. If an appointment request arrives that does not fit into the template, these holes may become filled through the generation of a new template.

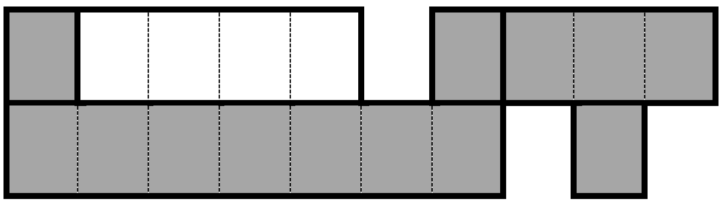
We return to the third request, r_3, of 15-minutes and schedule it into *TS*



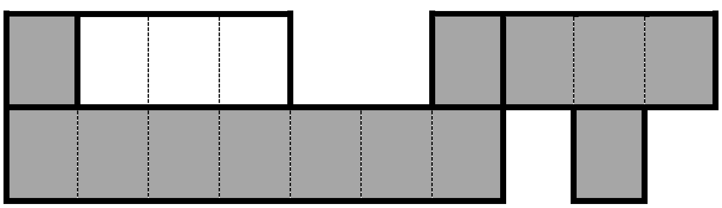
The assigned appointment is added to *F*. The fourth request, r_4, is a 45-minute appointment. It is scheduled into *TS*



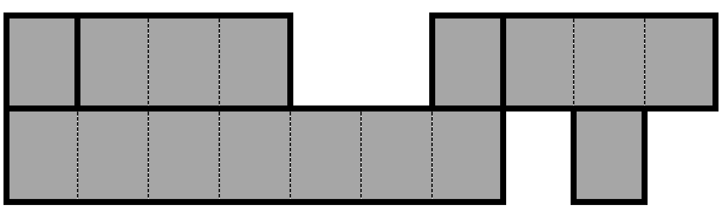
The assigned appointment is added to *F*. The fifth request, r_5, is for a 105-minute appointment and it is scheduled into *TS*



The last request, r_6, is for 45 minutes. Again, we need to update the template. This time, we are only looking for a spot for one appointment, the appointment that caused the conflict. Therefore, E=<r_6=e_1= 45> and the new optimal template *TS* = [{**15**, **15**, 45,**15**},{**105**, **15**}], where the bold numbers are already scheduled appointments. *TS* can be represented by



We schedule the final appointment request into *TS* and we get our final schedule.



As shown in the example, the scheduling procedure may leave empty slots in the schedule due to the need for real-time scheduling. When each request from *R* arrives and must be scheduled, we do not know the durations of the later requests and the actual set of requests is not likely to match exactly with what was expected. Furthermore, we assume here that once an appointment is scheduled, it cannot be moved. In Chapter 6, we explore the ability to move scheduled appointments in order to use resources more efficiently.

5.2 Empirical Distribution of Appointments

In this section, we present statistical analysis conducted on one year of data from the Odette Cancer Centre to analyze trends in appointment data in order to determine how much data to use when creating the empirical distribution. There are 350 chemotherapy regimens that can be grouped into buckets of appointments ranging in length from 15 to 510 minutes, in increments of 15 minutes. Research on new drugs is ongoing and the mix of treatment regimens continuously evolves. Using data from too far in the past could result in an inaccurate approximation of the future. Statistical analysis was performed to produce accurate predictions of the expected regimen mix.

Following the analysis, we investigated how to use the data. The empirical distribution is meant to provide a prediction of the number of appointments of each duration. Three possible data measures were considered for the creation of the empirical distribution: (1) the average number of appointments of each duration; (2) the mode of appointments of each duration; and (3) the median number of appointments of each duration.

5.2.1 Statistical Analysis – How Much Data to Use

Non-parametrical, Kruskal-Wallis testing using OriginPro 8.6 (OriginLab, Northampton, MA, 2011) can be used to determine if different sets of data are statistically likely to be drawn from the same distribution (Siegel 1956). This test can be used to evaluate whether the mix of regimens in one set of data is significantly different from another set by testing for significant differences between group medians. The null hypothesis is always that the groups of data are from the same distribution. All tests used a critical value of 0.05. If the test statistic is less than 0.05, we fail to support the null hypothesis.

The data was divided into 12 sets, one for each month. Next, Kruskal-Wallis was conducted to test the hypothesis that all the data could come from the same distribution. With a test statistic of 5.48×10^{-13}, the hypothesis was not supported. Then, tests were performed on two sets of data at a time. Table 5.1 provides a sample of the results of those tests.

Table 5.1. A selection of Kruskal-Wallis results

Comparison	Test Statistic	Decision
April versus May	0.0734	Failure to reject the null hypothesis
April versus June	0.0679	Failure to reject the null hypothesis
April versus July	0.0132	Reject the null hypothesis
September versus October	0.3242	Failure to reject the null hypothesis
September versus November	0.1098	Failure to reject the null hypothesis
September versus December	0.0392	Reject the null hypothesis

As the months from which data is drawn become more widely separated, the likelihood of their appointments being drawn from the same distribution falls. April is statistically from the same distribution as May and June, but not July. September is statistically from the same distribution as October and November, but not December. Figure 5.1 shows a grid of the year. Months for which the null hypothesis cannot be rejected are shaded.

Our conclusion is that three months of data is appropriate for creating empirical distributions. For the majority of the data, three months is the timeframe where statistically the months come from the same distribution, with the exception of three outliers. May is statistically from the same distribution as the following four months, June is statistically from the same distribution as November, and December is not statistically from the same distribution as February. Further analysis is warranted by the cancer centre to explore seasonal trends. For example, we might speculate that December differs from February due to the holidays.

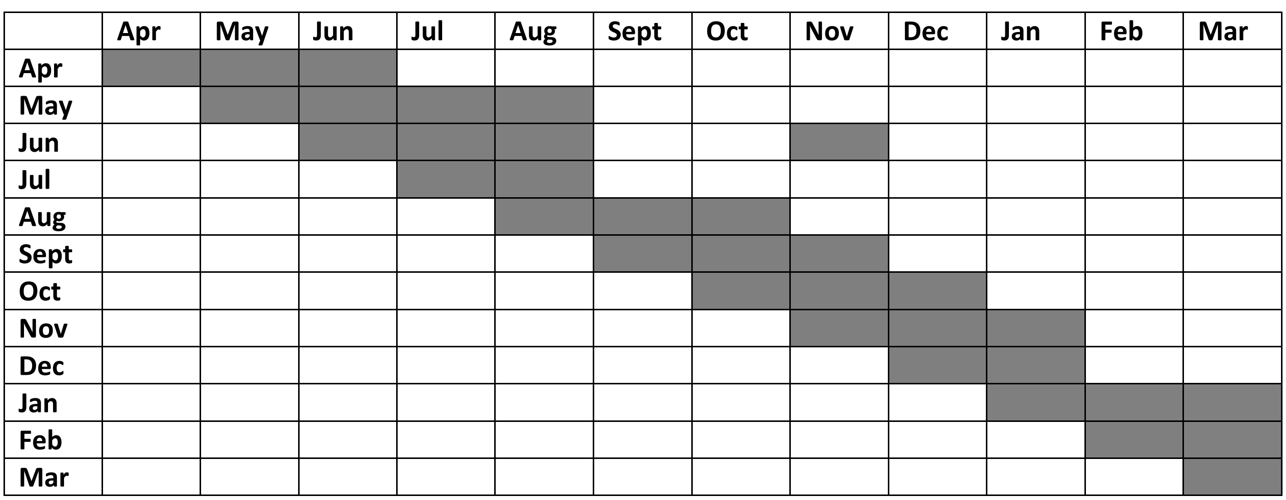


Figure 5.1. Months for which our statistical analysis indicates the appointment mix is drawn from the same underlying distribution

Table 5.2. Non-parametric tests - days of the week

Comparison	Test Statistic	Decision
Mondays vs. Tuesdays	0.91627	Failure to reject the null hypothesis
Mondays vs. Wednesdays	0.10525	Failure to reject the null hypothesis
Mondays vs. Thursdays	0.45542	Failure to reject the null hypothesis
Mondays vs. Fridays	0.38337	Failure to reject the null hypothesis
Tuesdays vs. Wednesdays	0.07325	Failure to reject the null hypothesis
Tuesdays vs. Thursdays	0.37457	Failure to reject the null hypothesis
Tuesdays vs. Fridays	0.15544	Failure to reject the null hypothesis
Wednesdays vs. Thursdays	0.33173	Failure to reject the null hypothesis
Wednesdays vs. Fridays	0.70897	Failure to reject the null hypothesis
Thursdays vs. Fridays	0.47324	Failure to reject the null hypothesis

Further Kruskal-Wallis tests were conducted to determine if there is a statistical difference between days of the week. Based on the analysis, we failed to reject the null hypothesis for any pair of days. Table 5.2 provides the results of these Kruskal-Wallis tests.

Following the non-parametric testing, forecasting models were applied to the full set of data to further analyze the historical data's ability to approximate the future. Moving average and exponential smoothing were implemented on a year's worth of data to predict the number of 15-minute appointments that are expected. The moving average was conducted with three different window sizes. Table 5.3 provides the average errors. Note that the average error of each technique is less than the standard deviation of 15-minute appointments.

Table 5.3. Average error in forecasting the number of 15-minute appointments

Forecasting Technique	Average Number of 15-minute Appointments	Standard Deviation	Average Error (number of appointments)
Moving Average with window size 20	13.22	5.02	3.95
Moving Average with window size 10	13.62	4.88	3.87
Moving Average with window size 5	13.04	4.43	3.45
Exponential Smoothing	11.46	5.69	4.51
Actual Data	12.59	4.97	N/A

Forecasting is a means of using past data to predict future data. Since the average error is less than the standard deviation that exists in the data, the error observed could be explained by variation we would expect to see in the data anyway. This result indicates that the historical data can be used to make predictions of future data.

5.2.2 How to Use the Data

In Section 5.2.1 it was determined that three months of data is appropriate for approximating the future. In this section, the question of how to use that data is addressed. We created three templates corresponding to the three ways to generate the expected number of appointments of each duration: (1) the *average* number of appointments of each duration; (2) the *mode* of appointments of each duration; and (3) the *median* number of appointments of each duration.

Each template was then used to schedule as many appointments as possible for each day for the three month period from January 1st through March 31st 2010 and we calculated the percentage of appointments that actually occurred over the full three months that could be scheduled into the given template. We also calculated the percentage of days in the three month period where at least 75% of the day's appointments could be scheduled into the template. Table 5.4 provides this information. Figure 5.2 is a visualization of the three options.

The template created using the average number of appointments of each duration could accommodate over 73% of appointments. Furthermore, 56% of the days considered had at least 75% of their appointments accommodated using the average template.

Table 5.4. Template information

Template	**Treatment Minutes**	**Number of Appointments**	**Appointments Accommodated by the Template (%)**	**Days where at least 75% of appointments can be accommodated (%)**
Average	11340	101	73.5	56.8
Mode	6915	88	61.3	47.7
Median	8962	92	68	54.5

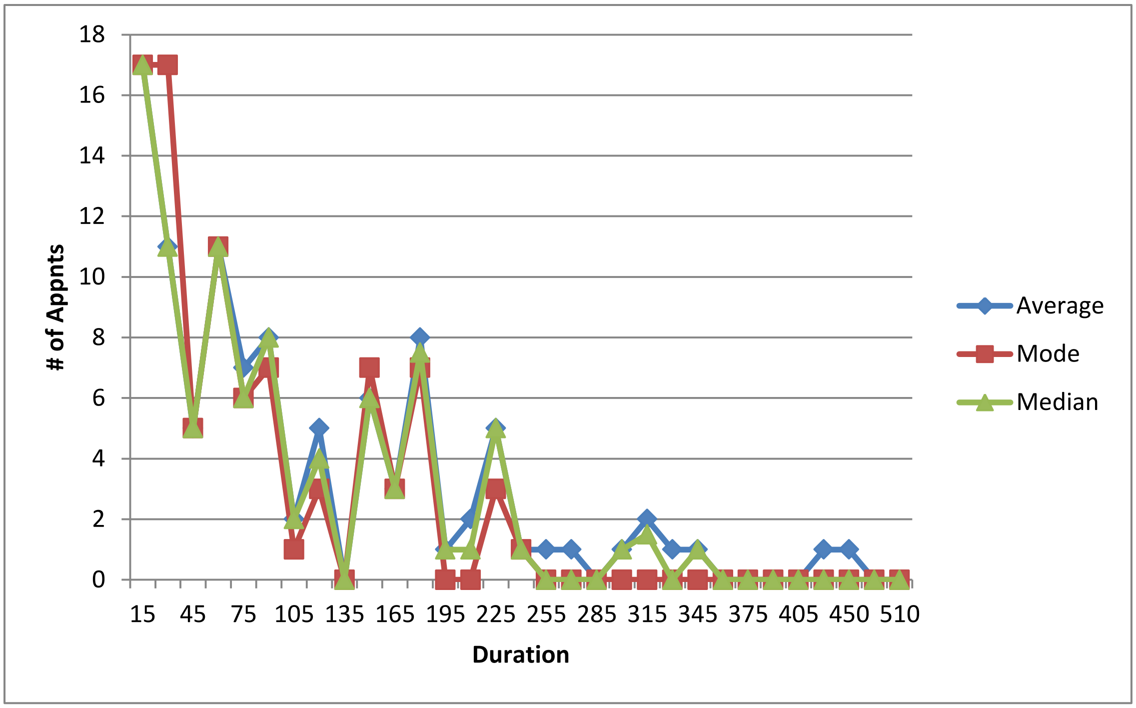


Figure 5.2. The number of predicted appointments of varying durations in the three possible templates

Based on the analysis, we determined that the empirical distribution should be created using the average number of appointments of each duration over a three month period. The empirical distribution used is provided in Appendix C.

5.3 The Stochastic Problem and Dynamic Algorithm

The dynamic template algorithm is formalized in Algorithm 5.1. As inputs to the algorithm, we provide *q*, the expected number of appointment requests for a given day, and *R*, the set of actual appointment requests that arrive in real time. In line 1, a sample of size *q* is taken from the empirical distribution and becomes the initial set of expected appointments *E*. In line 2, the set of scheduled appointments, *F*, is initialized as an empty set. In line 3, *CRS* with inputs *E* and *F* provides as output the initial template *TS*.

The appointment requests in set *R* arrive one request at a time. For each request (line 4), the current template is checked to see if there is a matching slot available for the request (line 5). In line 6, if there are multiple slots available, the appointment is assigned to the earliest one. In line 7, that newly assigned appointment request becomes part of the set of scheduled appointments *F* and in line 8, *q* is reduced by one. If there is no matching slot available, the template must be updated. In line 10, the empirical distribution is used to generate a new list of expected appointments, *E*. The appointment that caused the conflict is added into the sample to ensure that there will be a matching timeslot in the new template (line 11). In line 12, the optimization model creates a new template. The dynamic template scheduling algorithm continues assigning requests to open timeslots in the updated template until it receives another request that causes a conflict.

If more than *q* requests are received, each additional request after the q^{th} request will be treated as a conflict and a new template will be made with a spot for the additional request. Once all the requests from *R* are assigned start times, the day is considered complete.

DYNAMIC TEMPLATE SCHEDULING (*q*,*R*)

q – sample size representing the maximum number of appointments expected on a typical day
R – set of appointment requests that arrives in real time

1 Take a sample of size *q* from the empirical distribution, *E*
2 Initialize the set of already scheduled appointments *F* to an empty set
3 Use *CRS* to schedule *E* and create template *TS*
4 **while** there exists an appointment *r* in *R* **do**
5   **if** there is a matching slot in *TS* **then**
6   Book *r* into the earliest slot of matching duration
7   Add *r* to *F*
8   Reduce *q* by 1
9   **else**
10   Take a sample *E* of size *q – 1*
11   Add *r* to *E*
12   Update *TS* using *CRS* to schedule the new set *E* and current set *F*
13 **return** the complete assignment of *R*

Algorithm 5.1. Dynamic template scheduling pseudocode

5.4 Experimental Design

Three sets of experiments were conducted to test the dynamic template scheduling framework: (1) Comparison of dynamic template schedules with the optimal deterministic schedules and the actual schedules at the Odette Cancer Centre; (2) Comparison of dynamic template schedules to schedules using a proactive template; and (3) Sensitivity analysis on the effect of problem variability on the performance of dynamic template scheduling.

The experiments are run using 30 problem instances. Each instance is a list of appointments that actually occurred on a day at the Odette Cancer Centre during the months of January through March, 2010.

The test sets are solved using the algorithm described in Section 5.3. Each time the template has to be updated, the model is given 90 seconds to solve and the best solution at that time is used. During the execution of the algorithm, samples are randomly drawn from the empirical distribution of appointments. Since random numbers are used, every time the experiment runs, the solution is slightly different. Therefore, each test set is solved five times, for a total of 150 problem runs.

In the first set of experiments, we compare the performance of dynamic template scheduling to the optimal deterministic solution and the actual day that occurred at the Odette Cancer Centre to determine mean loss and mean improvement. The former is the schedule produced when it is assumed that all appointments for the day are given a priori i.e., they do not arrive in real time. The actual day is the schedule that occurred at the centre on the day in question.

The schedule produced by the dynamic template algorithm includes several key assumptions. The drug preparation time is assumed to be 30 minutes for all appointments, when in reality there are many that are shorter or longer. Second, the direct nursing requirement is assumed to be 15 minutes for all appointments, when in reality this duration fluctuates depending on the regimen. Therefore, the evaluation of the mean loss and mean improvement only provides an indication of the potential performance of the centre using dynamic template scheduling.

In the second set of experiments, we use the empirical distribution to create a static template that cannot be updated and compare the results using that template to the results using the dynamic template. As appointment requests arrive in real time, they are scheduled into the static template. If an appointment arrives that does not have a matching slot in the template, the request is scheduled into a longer slot that is closest in size. If there is no such slot available, the request is scheduled at the end of whichever chair is set to end first.

In the third set of experiments, we test the performance of dynamic template scheduling as we manipulate the variance of appointment durations. We compare the optimal deterministic makespan to the dynamic makespan using dynamic template scheduling and the static makespan using the static template.

5.5 Experimental Results and Discussion

In this section, we present the experimental results from the four sets of experiments described above.

5.5.1 Comparison with Deterministic and Actual Schedules

Table 5.5 provides the results of the experiments comparing the makespan resulting from dynamic template scheduling to the optimal deterministic makespan using *CRS* and the actual makespan that occurred at the Odette Cancer Centre. As expected, the dynamic template does not perform as well as the optimal solution and performs better than the current practice. There is an average difference between the optimal deterministic makespan and the dynamic makespan of just over 48 treatment minutes, indicating a reduction in quality (i.e., an increase in makespan) compared to the optimal deterministic makespan of 13%. However, the results show a large potential for improvement as there is an average difference between the dynamic makespan and the actual makespan of just over 85 treatment minutes, indicating an improvement over the actual schedule of 15%.

The schedules using dynamic template scheduling result in average chair utilization of 87.26% compared to the average optimal chair utilization of the deterministic problem of 91.04% and the average actual chair utilization of 68.69%. The first comparison in particular demonstrates that the dynamic template scheduling approach is achieving a utilization that is very close to the best that can be achieved.

Table 5.5. Comparison of dynamic, optimal, and actual makespans. Makespan is measured in 15-minute timeslots. The values in the table represent statistics of the difference between the dynamic makespan and the optimal and actual makespan. A positive difference means that the dynamic makespan was shorter and thus represents an improvement.

Test Set	Compared to Optimal	Compared to Actual
Average (timeslots)	-3.23	5.72
Standard Deviation (timeslots)	2.54	3.16
Median (timeslots)	3	5
Minimum (timeslots)	-11	-1
Maximum (timeslots)	0	15
95% Confidence Interval (timeslots)	(-8,0)	(1,12)

5.5.2 Comparison to a Static Template

In these experiments, we use the same five test days that are described in Section 4.7. There are a maximum of 40 15-minute slots available. If the appointment requests cannot be scheduled within the 40 timeslots, the static template fails. Table 5.6 compares the makespan using dynamic template scheduling to the makespan using the static template, called the static makespan.

In every case using the proactive template, results in idle resources (i.e., empty slots) in the middle of the day. For example, in several cases there is a chair with a large empty slot starting at time 0 followed by a filled slot. On that particular day, there is no appointment request corresponding to the long slot and so it is left empty, making the day unnecessarily long. In contrast, the dynamic template is able to adjust the template to better match the appointment requests arriving in real time.

Table 5.6. Static template comparison

Test Set	Static Makespan (timeslots)	Dynamic Makespan (timeslots)
1	40	26.8
2	40	31.6
3	40	33.9
4	Fail	34.1
5	40	35.9

5.5.3 Sensitivity Analysis

Using EasyFit 5.5 Professional (Mathwave Technologies, USA), we found the Generalized Pareto distribution with a mean of 46.5 minutes and standard deviation 125.6 to be the closest match to the empirical appointment duration data. Figure 5.4 shows a histogram of the historical data of appointment durations, overlaid with the Generalized Pareto probability density function. The x-axis is the appointment duration in minutes and the y-axis is the probability of an appointment of a given duration being requested.

Using the Generalized Pareto distribution, we can adjust the variance and test the performance of dynamic template scheduling for different levels of variance. In the actual chemotherapy data, the standard deviation of appointment duration is 92.847, slightly lower than the closest matching theoretical distribution shown below.

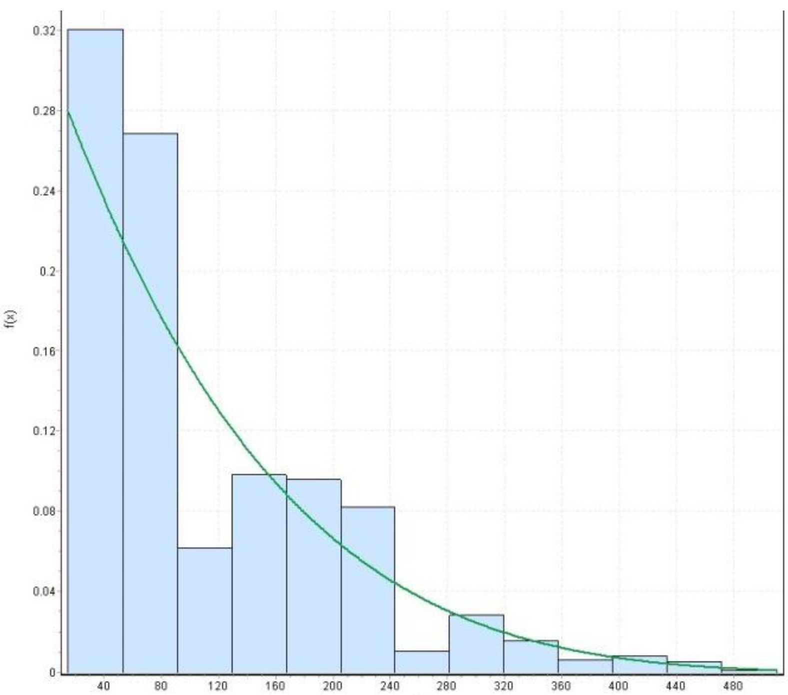


Figure 5.3. Probability density function of generalized pareto, μ = 46.5, σ = 125.6

Table 5.7. The comparison of dynamic, optimal, and static makespans at increasing levels of variance.

Standard Deviation	**Optimal makespan (timeslots)**	**Dynamic makespan (timeslots)**	**Static Makespan (timeslots)**	**Mean Loss (%)**	**Mean Improvement (%)**	**Number of Reschedules**
1	26	26	26	0	0	0
50	26	27.7	36	6.5	23.1	6
100	26	29.3	40	12.7	26.7	21
150	26	31.6	Fail	21.5	-	22
200	26	34.0	Fail	30.8	-	27
250	26	36.4	Fail	40.0	-	33

In these experiments, we used Generalized Pareto distributions with varying standard deviations: 1, 50, 100, 150, 200, and 250. We generated a data set of 50 appointments from each distribution and scheduled them assuming perfect knowledge of appointments using *CRS*, and assuming real-time appointment arrival using dynamic template scheduling and the static template. Table 5.7 shows the comparison of makespans. The mean percentage loss reflects the difference between the optimal makespan and the dynamic makespan and the mean percentage improvement reflects the difference between the dynamic and static makespans. The number of reschedules is how many times the dynamic template had to be updated. The maximum number of reschedules possible is 50, one for each appointment request.

As the standard deviation of the distribution is increased, the initial expected data set is further from the set of randomly generated appointments from the same distribution. As a result, more reschedules are required and the dynamic makespan is further off from the optimal. We observe that the number of reschedules increases significantly as the standard deviation changes from 50 to 100, but then levels off. The mean percentage loss with a standard deviation of 100 is very close to that of the real data with a standard deviation of 92.8. This result suggests, unsurprisingly, that the standard deviation of the problem data is an important determinant of the performance of dynamic template scheduling. We conclude that as the standard deviation increases, it is more important to dynamically reschedule. In the case of the standard deviation being equal to zero, the static template is as good as the dynamic one. However, even with a standard deviation of 50, the dynamic template provides a large improvement over the static template. Additionally, when the standard deviation increases to 150, the static template is not able to find a solution at all.

5.6 Conclusion

Dynamic template scheduling, a method for combining proactive and reactive optimization to address uncertainty in scheduling problems, was presented and illustrated on the chemotherapy outpatient scheduling problem. The method was tested against the optimal deterministic schedule and the current scheduling practice at the Odette Cancer Centre. The results show promise in applying optimization techniques to the chemotherapy outpatient scheduling problem. From the improvement in makespan, we conclude that there is potential to improve efficiency and to treat more patients with the current resources. The experiments show that there is value in doing future research on this problem and implementing the solution in a chemotherapy centre. In addition, we analyzed trends in appointment requests over time and were able to determine a timeframe for using historical data to create a distribution of expected appointments.

Dynamic template scheduling can be easily incorporated into existing scheduling systems, resulting in minor changes for users. The system would provide them with a set of optimal (or close-to-optimal) options for appointment times. The users can apply their own knowledge and judgment to choose from the given options.

Besides chemotherapy scheduling, there are many other applications that exhibit real-time uncertainty; e.g., health care applications where appointments are booked in advance, a factory where stock is made to order, a courier service where requests arrive at various times and decisions must be made on when to service them. Dynamic template scheduling can be applied to any problem where requests need to be assigned a start time before all requests are known and there is good historical data from which accurate expected request distributions can be drawn.

This chapter makes a contribution to scheduling research by developing a novel technique that combines proactive and reactive scheduling to address dynamic problems with real-time uncertainty.

Chapter 6

[redacted] Accommodating Last Minute Schedule Adjustments

At the Odette Cancer Centre, there is currently no formal process to accommodate additions and cancellations to the schedule. When possible, scheduling clerks manually reassign chairs to free up time, but they do not move appointment start times. Usually, additions to the schedule are assigned at the end of the day for a given chair, often resulting in overtime for staff. Cancellations are common and currently nothing is done to make use of the freed time. The result is unnecessary idle time where resources are not used effectively. In this chapter, we investigate reacting to unexpected schedule changes by *shifting*, moving appointment start times to an earlier or later time within specified limits.

In the dynamic template scheduling algorithm described in Chapter 5, once an appointment was scheduled, its start time was fixed. Now, start times of appointments are permitted to move by a predetermined maximum amount. In the chemotherapy problem, extra work would occur if patients required notification of changed appointment times and patients would be unhappy to have their appointment times changed at the last minute. Therefore, movement of an appointment's start time by as little as possible is preferred. In this chapter, we present the modifications to the model described in Chapter 5 to enable shifting of appointments and the experiments that test the effect of shifting on the makespan.

Following two sets of experiments to measure shifting, a run-through of the dynamic template system is described. In the run-through, many of the assumptions from previous experiments are removed to enable a more realistic test of the system, where we include cancelled appointments, shifting, and current booking guidelines at the Odette Cancer Centre.

6.1 Modifications to Accommodate Shifting

In the dynamic template scheduling algorithm described in Chapter 5, each time an appointment request arrives that does not have a matching slot in the template, the already scheduled appointments are saved and the remainder of the template is recreated using the optimization model. An additional constraint in the optimization model was added for every scheduled appointment, $f \in F$, which ensured the start times of those appointments would remain unchanged. Now, this constraint is altered and for each scheduled appointment, the constraint ensures that the new start time remains within a certain deviation from the original start time. Two constants, k_e and k_l, are the number of timeslots by which appointments can be shifted: k_e is the time by which the appointment can be made earlier and k_l is the time by which the appointment can be made later. In cases where k_e and k_l are equal, they are referred to collectively as k. Where S^* is the already scheduled start time and S is the variable representing the new start time, the altered constraint is

$$S_{if}^* - k_e \leq S_{if} \leq S_{if}^* + k_l, \qquad \forall f \in F, \forall i, i = 1,2,3 \qquad (13a)$$

Already scheduled appointments, instead of being fixed, are now allowed to move within specified limits.

6.2 Experiments

In this section we describe two sets of experiments designed to test the effect of shifting the start times of already scheduled appointments: (1) tests using shifting to accommodate added appointments, and (2) tests using shifting to adjust the schedule after cancellations occur. Full results for all the experiments can be found in Appendix D.

6.2.1 Accommodating Additional Appointments

The first set of experiments was done to test how shifting affects the makespan when appointments of various lengths are added to the schedule. The experiments contain three sets of experimental conditions:

1. The starting schedule. Experiments were conducted using two sets of completed schedules: (a) dynamic template schedules, the schedules resultant from using the dynamic template scheduling system and (b) real schedules, the schedules used by the Odette Cancer Centre.
2. The number and length of added appointments. Four sets of added appointments are used: (a) one appointment with duration of 120 minutes, (b) four appointments with durations of 120 minutes each, (c) one appointment with duration of four hours, and (d) two appointments with durations of four hours each.
3. The permitted start-time change, k. The guidelines at the Odette Cancer Centre state that patients only need notification if their appointment time has moved by more than a half

hour, or two timeslots. We therefore experiment with four shifting times, three of which are thirty minutes or less: (a) 15 minutes (1 timeslot), (b) 30 minutes (2 timeslots), (c) 60 minutes (4 timeslots), and (d) the case where k_e is 30 minutes and k_l is 15 minutes (2 and 1 timeslot respectively). The final combination of values was chosen with the intuition that majority of patients would not mind as much if their appointment was pushed earlier.

In this set of experiments we use the five data sets described in Section 4.7. A problem instance, defined by a completed schedule and a set of added appointments to be added is solved twice by the optimization model: once with shifting prohibited and once with shifting permitted. In our experiments we start with a completed schedule represented by a set of already scheduled appointments, *F*. We then call the optimization model to schedule the new appointments that are to be added. The optimization model adds the new appointments with the goal of minimizing makespan. When shifting is prohibited, *k* is set to 0 and when shifting is permitted, *k* is set to the corresponding limit.

By shifting appointments, the makespan of the day can be made shorter by at most the permitted time gap, *k*, since the last appointment in the starting schedule can only be moved by at most *k* time units. In our experiments, the average makespan is calculated for each starting schedule and allowed shifting time, over all cases of added in appointments. A positive difference in makespan corresponds to an improvement by allowing shifting.

Table 6.1 provides the average difference in makespan when shifting is permitted compared to when shifting is not permitted. The entries in the table are the differences in makespan given in the number of 15-minute timeslots. When starting from the real schedule, the makespan improves by *k* even though appointments are being added. To a lesser degree, there is also improvement when we start with a dynamic schedule.

Table 6.1. The average difference in makespan (in number of timeslots) when shifting is allowed when adding appointments

Starting Model	**Time Gap k (timeslots)**			
	1	**2**	**4**	**2/1**
Dynamic Template	0.7	0.9	1.2	0.8
Real	1.0	2.0	4.0	1.9

By permitting shifting of appointment start times, the centre could accommodate additional appointments. The real schedules contain more time where resources are unnecessarily idle and as a result more improvement can be made by shifting compared to starting from a template schedule.

Permitting shifting does not increase the run time of the optimization model or the run time of the dynamic template scheduling algorithm, as verified during the experiments. The optimization model is solved the same number of times with an equal number of constraints.

6.2.2 Adjusting after Cancellations

The data used in this set of experiments is from the same five days as above, but it now includes appointments that were booked and then cancelled. Appointments are cancelled for many reasons, including patients who have blood work results that are not acceptable for continuing with treatment and patients whose protocols change. When a patient is booked for his first appointment, all the future appointments within his regimen are also booked. If at any point, the patient's treatment plan changes, all his future appointments are cancelled. On average, 20% of appointments are cancelled at the Odette Cancer Centre.

Table 6.2 provides the data pertaining to cancelled appointments from January through March, 2010 as well as the data specific to our test sets. For January through March, the percentage of the total number of appointments cancelled is provided, as well as the mean and standard deviation of appointment minutes cancelled per day. The same information is provided pertaining to the complete week of test sets.

Table 6.2. Cancellation data

Sample	**Mean Appointments Cancelled (%)**	**Mean Appointments Cancelled per Day (min)**	**Standard Deviation (min)**
Jan- Mar 2010	20	2669	916.04
Test Week	19	2811	680.86
Test Set Day 1	19	1650	N/A
Test Set Day 2	19	3015	N/A
Test Set Day 3	16	3270	N/A
Test Set Day 4	20	3315	N/A
Test Set Day 5	20	2805	N/A

The experiments to test the effect of shifting to accommodate cancellations are conducted as follows:

1. The complete set of appointments is scheduled using the optimization model. The makespan of the full schedule is recorded as the full makespan.
2. The appointments that were cancelled are removed from the schedule. In this set of experiments all the appointments are cancelled at once, after the complete day is scheduled. In reality cancellations occur over time. The specific timing of cancelled appointments is included in the run-through described below in Section 6.3. The makespan of the schedule with the cancelled patients removed is recorded as the makespan after cancellations.
3. The start times of appointments are rescheduled using the optimization model and the start times are permitted to be shifted by 30 minutes. The makespan is recorded as the makespan after shifting.
4. The makespan after shifting is compared to the makespan before shifting.

Table 6.3 provides the results, given in 15-minute timeslots, of the experiments testing the effect of shifting after cancellations. By shifting appointments, the centre can achieve a better schedule after cancellations occur.

Note that days two through five have the maximum reduction in makespan as each appointment can only move by at most 2 timeslots. When appointments are cancelled and no shifting is done, in order for the makespan to change, an appointment would need to be cancelled from the last slot of each chair that ends at the makespan. In our data, there was no such collection of cancellations.

Table 6.3. Cancellation experiments

Test Set	**Full Makespan (timeslots)**	**Makespan after Cancellations (timeslots)**	**Makespan after Shifting (timeslots)**	**Improvement in Makespan (timeslots)**
Day 1	24	24	23	1
Day 2	34	34	32	2
Day 3	39	39	37	2
Day 4	37	37	35	2
Day 5	37	37	35	2

6.2.3 Discussion

Shifting is a controversial topic at the Odette Cancer Centre. There is fear that patients will be unhappy if their appointment times get moved. Currently, appointment start times are never changed once they are given to the patient. However, the reality is that appointments never happen on time as there is a lot of uncertainty in practice. Patients are assigned an appointment time to arrive at the centre; they are not told the expected time to start treatment. Shifting treatment times by a small amount, therefore would not negatively impact the patient. In this chapter we showed that shifting a small amount can improve the makespan when appointments are added or cancelled. Additionally, the algorithm permits different values for shifting early and shifting late to allow for the case where a patient does not mind if his/her appointment gets pushed earlier, yet does mind if their appointment gets pushed late or vice versa. The model allows for different k values to be specified for each already scheduled appointment. The opinion of this researcher is that shifting should be added to the dynamic template system as an optional feature. The users should be able to enter the time limit and then shift only if they feel it is necessary.

6.3 Run-Through

In this section, we present a more realistic run-through of the dynamic template scheduling algorithm with and without shifting. The following is a list of changes that makes the run-through more realistic.

1. Previously, the experiments testing dynamic template scheduling assigned appointment requests to the earliest matching timeslots. In reality, a user chooses a timeslot for reasons such as the ability of a patient to access the centre at a certain time or the length of drug preparation. During the run-through, in cases where there is more than one matching timeslot, the timeslot is chosen randomly to represent more realistic behaviour.
2. Previously, all pharmacy preparation was assumed to be 30 minutes in length. In reality, there are some drugs that can be prepared more quickly and some that take longer. During the run-through, the following guidelines provided by the Odette Cancer Centre are used: (a) any patient with an appointment earlier in the same week (this information was provided by users at the cancer centre) can be assigned at the beginning of the day as their drugs can be completely pre-mixed and (b) regimens on a provided list have drugs

that take longer than 30 minutes to prepare and a patient on one of those regimens should not be scheduled at the beginning of the day, unless the patient has earlier appointments in the same week

3. Previously, only the appointments that occurred on a particular day were scheduled using the dynamic template algorithm. In reality, cancelled appointments affect the scheduling process. During the run-through, the complete list of appointments including those appointments that were subsequently cancelled is used.
4. Previously, the direct nursing portion of the treatment was assumed to be 15 minutes for all appointments. In the run-through, durations specific to the regimen being scheduled are used.

6.3.1 Experimental Set Up and Results

The run-through was conducted using Test Set 1 from the data used for previous experiments. The following steps outline the procedure used when conducting the run-through.

Step 1: Create a list of appointment requests, *R*, which need to either be booked or cancelled. To create the list, we use data from the Odette Cancer Centre and randomize the order of serviced appointments and intermittently insert cancelled appointments to be booked and, later in the list, cancelled. Cancelled appointments are inserted randomly because we do not know the order in which they were booked. The appointments are cancelled in the actual order they were cancelled in the data. The appointment list used in this experiment is provided in Appendix D.

Step 2: Create an optimal template using the empirical distribution. This template is an original template of open slots in which to book appointments.

Step 3: While there is still at least one appointment in *R*, schedule it using Algorithm 6.1. The guidelines from the Odette Cancer Centre are listed above. This algorithm can be used with or without shifting and we experiment with both conditions below. If shifting is permitted, shifting with *k* set to 30 minutes is allowed any time the optimization model is called.

Since there is randomness in dynamic template schedule in both the order of arrival of serviced appointment requests in *R* and the sample taken from the empirical distribution to create the template, we conducted the run-through 10 times. We then present a range of possible values that the makespan can take when using dynamic template scheduling.

```
while there exists an appointment in R do
Take an item from the top of the created appointment list R
Check if the item is an appointment to be booked or cancelled
if the appointment is to be booked then
    Check for a matching slot in the template
    if there is more than one matching slot then
        Use guidelines from the Odette Cancer Centre to choose and book
    else if there is no matching slot then
        Use the optimization model to recreate the remainder of the template
else if the appointment is to be cancelled then
    Cancel the appointment
return the complete assignment of R
```

Algorithm 6.1. Run-through algorithm

During the run-through, dynamic template scheduling was able to achieve an average makespan of 31 15-minute slots with a standard deviation of 1.9, corresponding to 3:45 in the afternoon. Dynamic template scheduling with shifting was able to achieve a makespan of 29 slots with a standard deviation of 1.3, corresponding to 3:15 in the afternoon. These times can be compared with 6:45 in the afternoon, the end time of the actual schedule for the corresponding day. These times can also be compared with 2:42 in the afternoon, the end time resultant from the original experiments using dynamic template scheduling, described in Chapter 5. Although the end time of the run-through is later than the end time of the previous experiments, there is still a sizeable improvement over current practice. Table 6.4 provides a comparison of the average makespan and chair utilization for the run-through without shifting, the run-through with shifting, the actual schedule that was used at the Odette Cancer Centre, the dynamic template makespan from the experiments presented in Chapter 5, and the optimal deterministic makespan using *CRS*. The makespan is provided as the number of 15-minute slots. Chair utilization is defined as the percentage of time that the chair is in use from the time when the earliest patient begins treatment until the last patient on that chair finishes. The result is integrated over all the chairs.

Table 6.4. Run-through comparison

Schedule	Makespan (timeslots)	Chair Utilization (%)
Run-through	31	79.61
Run-through with Shifting	29	80.23
Actual Schedule	36	64.09
Dynamic Schedule	27	86.35
Optimal Schedule	24	92.78

The run-through results are not as impressive as the optimal solution with perfect information (makespan of 24) or as the dynamic template solution (makespan of 26.8). However, the run-through represents a substantially more realistic scheduling process and the results highlight the potential advantage of dynamic template scheduling over current practice. Without shifting, the improvement in makespan and chair utilization is significant. Including the ability to shift appointments by 30 minutes, improves the makespan by 30 minutes, the maximum possible. Figure 6.1 shows the range of values for the makespan for the run-through, the run-through including shifting, the dynamic template experiments from Chapter 5, the optimal schedule with known information, and the actual schedule at the Odette Cancer Centre.

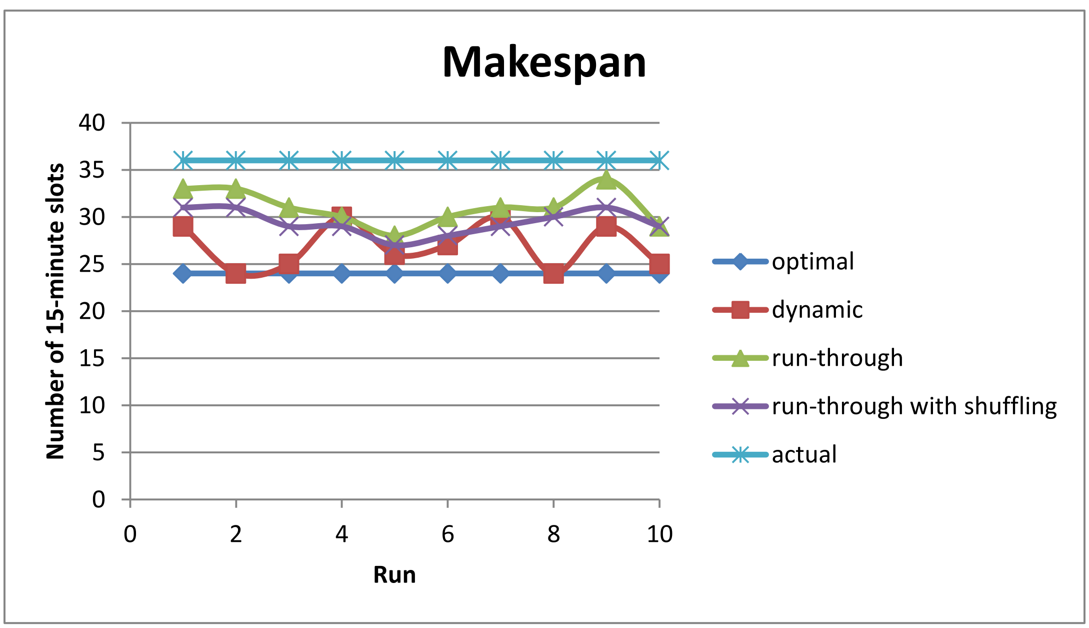


Figure 6.1. Comparison of makespan values.

6.3.2 Discussion

The run-through described in this chapter was a relatively realistic test of the dynamic template system. It resulted in a makespan 1.25 hours shorter than the actual schedule, sufficient time to service additional patients.

Although every attempt was made to create a run-through that was as realistic as possible and to reflect the procedure of using dynamic template scheduling in practice, there still remain the following aspects of the run-through that are not realistic.

1. Patients' preferences on treatment start times are only taken into account indirectly, by placing appointments randomly when there are multiple matching slots.
2. Durations of all stages are assumed to be deterministic. Although there are relatively accurate durations for pharmacy and both stages of treatment, there can still be treatments that take longer than expected due to complications or adverse reactions to the drug.
3. Last minute ("same-day") additions are not taken into account. Although rare, it can happen and if it does, the schedule will be affected.
4. Patients are assumed to arrive on time for their appointments, which is not always the case.
5. Nurses are assumed to be ready to treat patients if they are not at full capacity. In reality, nurses conduct many tasks besides administering treatment to patients.

6.4 Conclusion

In this chapter, we addressed the uncertainty in the chemotherapy problem stemming from additions and cancellations to the schedule, an aspect of the problem that has not been studied in the literature, yet has a large impact on the real world scheduling process. Additionally, we conducted a run-through of dynamic template scheduling that represented a substantially more realistic test of the scheduling approach.

In order to accommodate changes to the schedule from additions and cancellations, we proposed shifting, moving the start times of scheduled appointments by at most a pre-determined amount. Experimental results showed that shifting, even by a small amount, can improve the makespan and resource utilization, so fewer timeslots are left in the schedule where resources are unnecessarily idle.

During the run-through of dynamic template scheduling, many previous assumptions were removed, guidelines from the cancer centre were utilized, and cancellations were included in the scheduling process. The results still showed that using dynamic template scheduling with or without shifting can result in better utilization of resources and a shorter makespan than current practice.

Chapter 7

[redacted]Validation at a Second, Independent Cancer Clinic

This chapter applies the work developed in Chapters 4 through 6 to a second cancer clinic and acts as a validation of dynamic template scheduling for solving the chemotherapy outpatient scheduling problem. In particular, we apply dynamic template scheduling and the processes outlined in the previous chapters to the chemotherapy outpatient scheduling problem at the Juravinski Cancer Centre in Hamilton, Ontario, Canada.

In this chapter, the chemotherapy application at the Juravinski Cancer Centre is described. Then, the processes of choosing an optimization model, applying dynamic template scheduling, and experimenting with shifting are used to solve the scheduling problem at the Juravinski Cancer Centre. Modifications made to the model of the Odette Cancer Centre and subsequent experiments are documented.

7.1 Application Description

In this section, we describe details of the process for the specific case study of the chemotherapy outpatient scheduling problem at the Juravinski Cancer Centre in Hamilton, Ontario, Canada. The following analysis comes from observations and interviews conducted with unit clerks in charge of booking and the resource nurse on duty at the Juravinski Cancer Centre on May 12, 2009.

At the Juravinski Cancer Centre, up to 150 chemotherapy appointments take place each day, resulting in 30,000 systemic treatment visits in a year. In combination with three affiliated community clinics in the same local health integration network, the Juravinski Cancer Centre forms the largest provider of cancer care in the province.

The Juravinski Cancer Centre books appointments with an electronic tool. The centre has guidelines for the maximum number of patients that can be booked based on the number of available nurses and an online database is used to define the details of each regimen. As opposed to the Odette Cancer Centre, where CHARM has this information built in, here the booking clerk must manually check the database.

The chemotherapy outpatient scheduling problem at the Juravinski Cancer Centre is very similar to the Odette Cancer Centre as described in Section 3.2. The main difference lies in the number of resources. As the number of pharmacy servers and nurses change from day to day, we use the following values based on the average resource capacity during the study period: four pharmacy servers to conduct chart review and prepare drugs, 23 nurses, six of which are part time, and a total of 17 beds, 21 chairs, and two isolation rooms. At Juravinski, the chart review stage is conducted by a pharmacy technician instead of a nurse dedicated to the task. The chart review stage is still separate from the drug preparation stage, but uses the same set of resources. In this chapter, we again solve the chemotherapy outpatient scheduling problem with a horizon of one day and each calendar day is a separate problem instance.

Scheduled appointment data was provided by the Juravinski Cancer Centre for a three month period, April through June 2011. In the available data, the average number of total appointments that occurred per day was 89.85 with a standard deviation of 16.25 appointments. The average number of total appointment minutes per day was 11469.57 with a standard deviation of 846.73 minutes. Table 7.1 shows the average utilization of the nurses, pharmacy technicians, and chairs. We see that pharmacy technicians are highly utilized, while the other resources are not.

At the Juravinski Cancer Centre, although each treatment requires the use of many resources, the current process of booking appointments assumes that nurses are the bottleneck and there is sufficient capacity of chairs and pharmacy technicians to service the appointments booked. In our analysis, we believe that pharmacy technicians are the constraining resource.

Table 7.1. Utilization of nurses, pharmacy technicians, and chairs – Juravinski Cancer Centre

Resource	**Number Available**	**Capacity (minutes)**	**Minutes Used**	**Utilization (%)**
Nurses	20	67050	12820	19.12
Pharmacy	3	2052	2100	100.02
Chairs	40	27360	11470	41.90

The distributions of appointment data at the two cancer centres are statistically different. When tested using the non-parametric Kruskal-Wallis test with the null hypothesis that the two centres have data from the same distribution, we fail to accept the null hypothesis with a critical value of 0.05. A comparison of baseline statistics of appointment length data from the two cancer centres is provided in Table 7.2 and Figure 7.1 shows a comparison of the probability density functions at the two centres. There is an observable difference between the two graphs, providing visual confirmation of our statistical test.

Both the Odette and Juravinski Cancer Centres have large chemotherapy clinics and treat around 100 patients each day. However, the two centres have statistically different appointment distributions and a different mix of resources. By successfully applying dynamic template scheduling as developed at Odette to the chemotherapy outpatient scheduling problem at Juravinski, we demonstrate the algorithm’s flexibility and support the claim that it may be applicable at other chemotherapy clinics as well.

Table 7.2. Comparison of appointment length statistics from the two problem distributions

Statistic	**Odette**	**Juravinski**
Mean	110.30	138.25
Standard Error	1.29	1.59
Median	75.00	90.00
Mode	15.00	15.00
Standard Deviation	93.49	109.39
Sample Variance	8740.51	11966.78
Kurtosis	1.29	-0.99
Skewness	1.23	0.56
Range	495.00	345.00
Minimum	15.00	15.00
Maximum	510.00	360.00

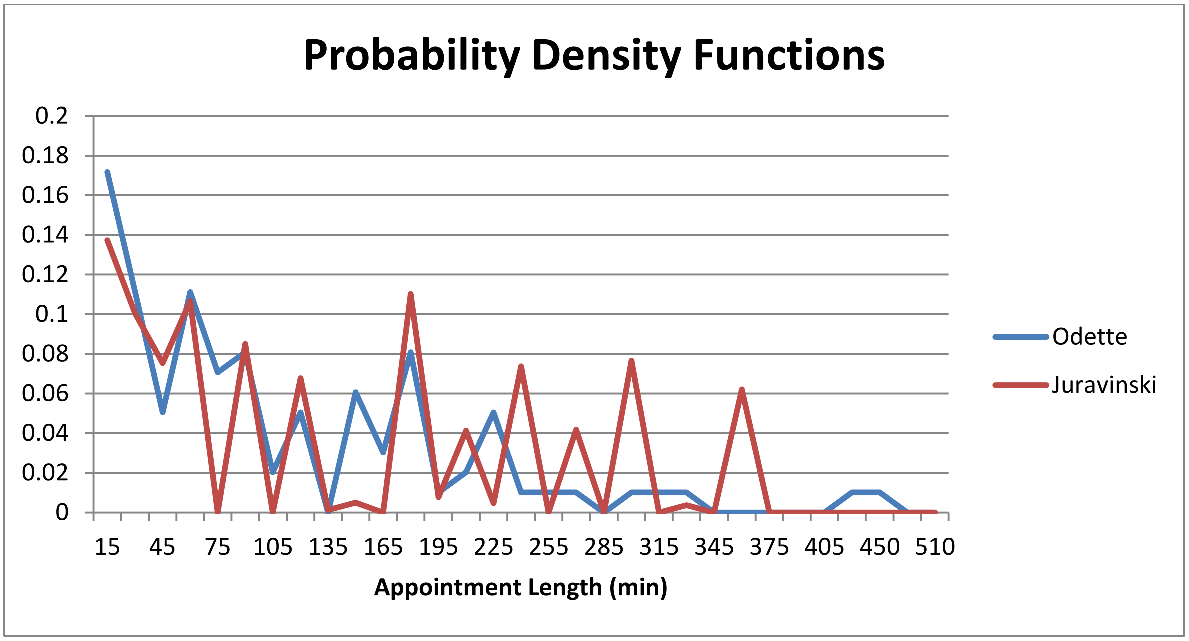


Figure 7.1. Probability density functions at the two centres.

The experiments described in this chapter are performed on one week of data from June 13th through June 17th, 2011. Table 7.3 provides information about the data. The estimated average and maximum wait times are provided by subject experts at the cancer centre based on their observations.

Table 7.3. Experimental data – Juravinski Cancer Centre

Test Set	Number of Appointments	Total Treatment Minutes	Estimated Average Wait Time (min)	Estimated Maximum Wait Time (min)
Day 1	64	10860	60	240
Day 2	87	11235	60	240
Day 3	76	11490	60	240
Day 4	87	12315	60	240
Day 5	72	11445	60	240

7.2 Choosing an Optimization Model

This section presents the methods and experiments for choosing an optimization model for the chemotherapy outpatient scheduling problem at the Juravinski Cancer Centre. The methods and experiments are analogous to the work presented in Chapter 4 for the Odette Cancer Centre. The goal is to determine which optimization model rates well in both solvability and quality. Navigating through this process at a second site tests the method for choosing a model as well as the model itself.

7.2.1 Methods

Based on the statistics at the Juravinski Cancer Centre, pharmacy technicians appear to be the most constraining resource. We therefore add an additional constraint model, the *Pharmacy Only* model (*PO*), to the models evaluated in Chapter 4. This modification does not take into account nurses or chairs, but rather assumes that the only resource is pharmacy technicians. The modification was chosen because pharmacy is thought to be the constraining stage in the process at the Juravinski Cancer Centre and by eliminating the resources of nurses and chairs, the problem is simplified. The output of the model is the optimal assignment of start times for the pharmacy stage of each appointment assuming that there will be enough resource capacity for the treatment stage of each patient once the pharmacy stage is completed. The expectation is that *PO* will be equal in quality to *CRS*, but will be easier to solve due to it being a simpler model.

7.2.2 Experimental Results

When solving the problem instances, a feasible solution was found 80% of the time, with a feasible solution being found within one second 77.14 % of the time across all models. Figure 7.2 shows the percentage of instances solved to optimality at different time intervals. With the exclusion of the base model, that could not solve any of the problem instances to optimality, the remainder of the models found optimal solutions to at least 20% of the problem instances within the maximum run-time of 3600 seconds.

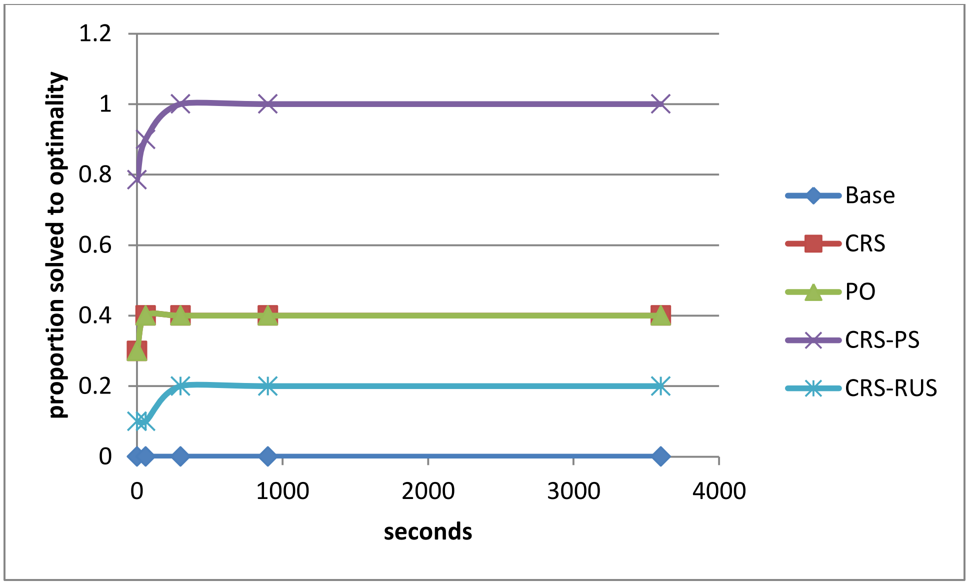


Figure 7.2. Proportion of instances solved to optimality versus time – the Juravinski Cancer Centre.

Table 7.4 presents the average results for all models and test days at the Juravinski Cancer Centre. These results are used to make a rating scale in Table 7.5 which is then used to rate the different definitions. The description refers to the percentage relative to the distribution of the measures over all models and test sets. For example, 75% of the models had a wait time better (shorter) than 178 minutes. Therefore, any model with a longer average wait is assigned a rating of poor for that metric.

Additional ratings of *Fair* and *Good* are used where the two solvability or quality measures fall into different rating groups. *Fair* refers to a mix of *Poor* and *Average* and *Good* refers to a mix of *Average* and *Very Good*. A rating of *Average* is given if there is a mix of *Poor* and *Very Good*.

Table 7.4. Average solvability and quality - the Juravinski Cancer Centre

Measure	**Value**
Average time to solve (s)	2263.46
Percent solved to optimality (%)	37.14
Average wait (min)	61.22
Maximum wait (min)	121.36

Table 7.5. Rating scale values – the Juravinski Cancer Centre

Rating	Description	Time to Solve Value (s)	Optimality Value (%)	Average Wait (min)	Maximum Wait (min)
Poor	Worse than 75%	> 3599	< 20	> 178	> 314
Average	25-75%	0.16 – 3599	20 – 30	14 – 178	31 – 314
Very Good	Better than 25%	< 0.16	> 30	< 14	< 31

Table 7.6 shows the results for testing done on seven different models of the same problem. The tests show the trade-off between models that have high solvability and models that have high quality, which result in solutions that work well in the simulation model. The conclusion based on the values and ratings is that *CRS* is the model of choice. *PO* is almost identical to *CRS*, but the average wait time is slightly longer.

At both the Odette and Juravinski Cancer Centres, *CRS* was the preferred model. Figure 7.3 shows the order in which the models performed in solvability and quality at each centre. The value given for each model is the combined score for both solvability and quality. In the graph a higher number correlates to a better score. An overall rating of *Poor* is equal to a one, while a rating of *Very Good* is a 5. Although the ratings of the models were different at each cancer centre, the model that performed the best (*CRS*), was the best at both centres.

Table 7.6. Model comparison – the Juravinski Cancer Centre

	Solvability			**Quality**		
Model	**CPU time (s)**	**Optimality (%)**	**Rating**	**Average Wait (min)**	**Max Wait (min)**	**Rating**
Base	3600.77	0	Poor	15.94	30	Good
CRS	2160.02	40	Good	8.67	30	Very Good
PO	2160.02	40	Good	11.48	30	Very Good
CRS-RUS	2882.34	20	Average	179.30	315	Poor
CRS-PS	0.07	100	Very good	179.30	315	Poor

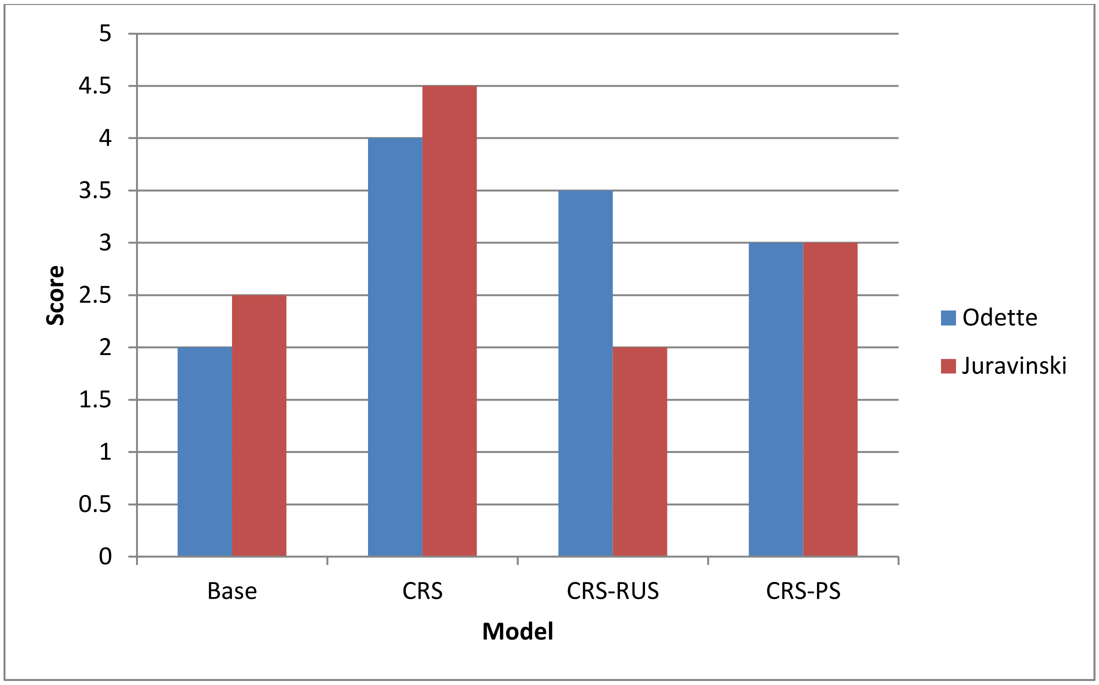


Figure 7.3. Comparison of model performance at the Odette and Juravinski Cancer Centres

Overall, the Odette Cancer Centre problem sets were more solvable than the Juravinski Cancer sets, but the quality of the solutions were very similar. The comparison of the chosen results, from *CRS*, for both centres is provided in Table 7.7. The time to solve was considerably higher with the Juravinski Cancer Centre problem sets and the percent of instances that were solved to optimality was 50% lower highlighting the difference in solvability. The Juravinski problem is more constrained due to the lower capacity of pharmacy technicians, a higher number of longer appointments, and more daily appointment minutes. These differences may explain the lower level of solvability compared to the Odette Cancer Centre problems.

Table 7.7. *CRS* results at Odette and Juravinski Cancer Centres

	Time to Solve (s)	**Percent Optimality (%)**	**Average Wait (min)**	**Maximum Wait (min)**
Odette	720.13	80	9.00	22
Juravinski	2160.02	40	8.67	30

7.2.3 Conclusions

The chosen model, *CRS* is the same model chosen at the Odette Cancer Centre. Similar trade-offs were shown between solvability and quality in the set of problem definitions considered. The work described in this section validated the proposed method for choosing a problem definition and the specific model chosen using a second chemotherapy outpatient scheduling problem application.

7.3 Dynamic Template Scheduling

This section presents the experiments using dynamic template scheduling to solve the chemotherapy outpatient scheduling problem at the Juravinski Cancer Centre. The methods and experiments are analogous to the work presented in Chapter 5 for the Odette Cancer Centre. The goal is to show that dynamic template scheduling can be applied to another instance of the problem, thereby validating the method as a valuable technique for solving the chemotherapy outpatient scheduling problem.

The Juravinski Cancer Centre provided three months of appointment data, April through June 2011, which was used to create an empirical distribution of expected appointments. As was done in Chapter 5 for the Odette Cancer Centre, the average number of appointments of each length was used to create the distribution.

7.3.1 Experimental Results

Table 7.8 provides the results of the experiments comparing the dynamic makespan resulting from dynamic template scheduling to the optimal (full knowledge) makespan and the actual makespan from the corresponding day in the clinic. The mean loss is defined as the mean percentage difference in makespan between the dynamic template solution and the optimal deterministic solution. The mean improvement is defined as the mean percentage difference in makespan between the actual schedule and the dynamic template solution.

Table 7.8. Dynamic template scheduling versus the optimal solution and the actual schedule – the Juravinski Cancer Centre

Test Set	**Optimal Deterministic Makespan (timeslots)**	**Dynamic Makespan (timeslots)**	**Actual Makespan (timeslots)**	**Mean Loss (%)**	**Mean Improvement (%)**
1	31	33.2	41	7.10	19.02
2	38	38.7	43	1.84	10.00
3	33	39.0	47	18.18	17.02
4	38	38.3	51	0.79	24.90
5	33	38.4	46	16.36	16.52

As expected, the dynamic template does not perform as well as the optimal solution and performs better than the current practice. There is an average difference between the dynamic and optimal makespan of close to 44 treatment minutes, representing an average loss of 8.85%. However, the results show potential for improvement in makespan at the cancer centre. There is an average difference between the dynamic makespan and the actual makespan of just over 121 treatment minutes, representing an average improvement of 17.49%.

Figure 7.4 provides a comparison of the impact of applying dynamic template scheduling at both centres. The average percentage loss when compared to the optimal solution and the average percentage improvement when compared to the actual schedule is shown. The impact of applying dynamic template scheduling at both cancer centres is very similar in magnitude.

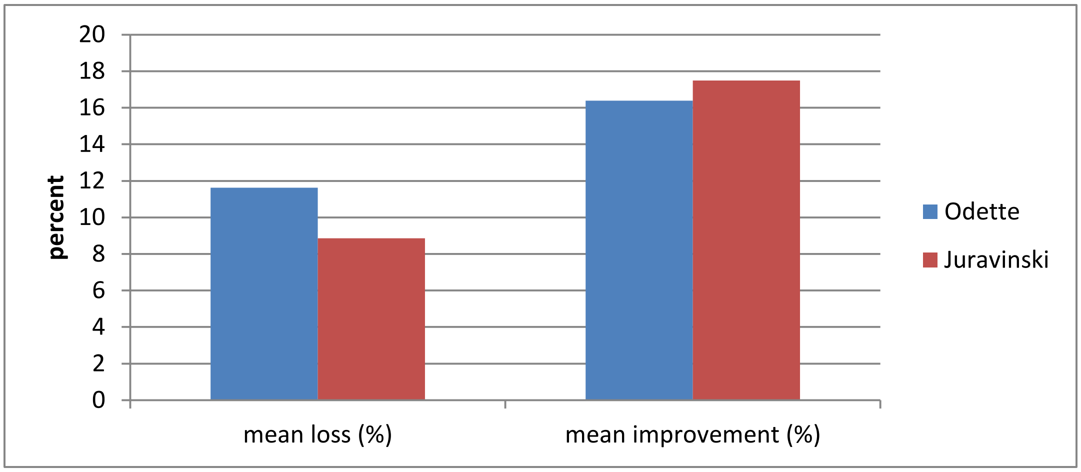


Figure 7.4. The impact of dynamic template scheduling at the Odette and Juravinski Cancer Centres

7.3.2 Conclusions

The dynamic template scheduling method results in improved makespan over current practice at the Juravinski Cancer Centre, just as at the Odette Cancer Centre. Similar improvements are shown on the two problems despite the difference in the distributions of appointments.

7.4 Accommodating Last Minute Schedule Adjustments

This section presents experiments aimed at evaluating the value of shifting start times of already scheduled appointments. The methods and experiments are analogous to the work presented in Chapter 6 for the Odette Cancer Centre. The goal is to show that the addition of shifting start times to the dynamic template scheduling method can result in improvements to the makespan when applied to another instance of the problem.

As at the Odette Cancer Centre, the Juravinski Cancer Centre does not shift appointment times once they have been given to the patient. If they need to book a last minute appointment, they generally schedule the appointment following the last appointment scheduled for a given nurse.

7.4.1 Experimental Results

Table 7.9 provides the average difference in makespan when shifting is permitted compared to adding the appointments when shifting is not permitted. We followed the same experimental setup and details as in Section 6.2.1. By shifting appointments, the makespan of the day can be reduced by a maximum of the permitted time gap, *k*. The average makespan is calculated for each starting schedule and allowed shifting time, over all cases of added appointments. The last column of the table shows the case when the allowed shifting time is different for shifting appointments earlier versus later. In this case, shifting is permitted by two timeslots if the appointment is being moved earlier and only one timeslot if the appointment is being moved later. The entries in the table are the differences in makespan in 15-minute timeslots.

By permitting shifting of appointment start times, the centre could accommodate additional appointments more efficiently. Independent of the starting schedule, shifting results in improvements close to the maximum possible, *k*, even when appointments are added to the schedule.

Figure 7.5 shows the differences in makespan in minutes when varying amounts of shifting are permitted. The values shown are improvements in makespan beginning from the dynamic template schedule. Since cancellation data was not available for the Juravinski Cancer Centre problem sets, we can only compare the effects of shifting when appointments are added at both centres.

Table 7.9. Differences in makespan when adding appointments – the Juravinski Cancer Centre

Starting Model	**Time Gap k (timeslots)**			
	1	**2**	**4**	**2/1**
Dynamic	0.93	1.65	3.25	1.60
Real	0.85	1.70	3.90	1.50

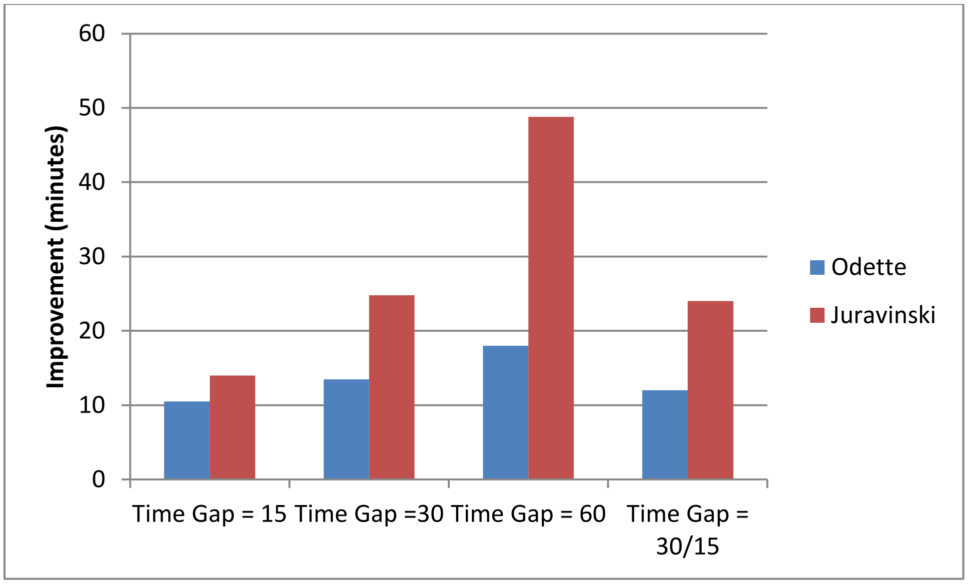


Figure 7.5. Differences in makespan when shifting is permitted at the Odette and Juravinski Cancer Centres

There is a larger difference in makespan when shifting is permitted on the problem sets at the Juravinski Cancer Centre. These results indicate that shifting has more of an effect on the makespan on the Juravinski problem sets than they do on the sets from the Odette Cancer Centre. This result may be due to the scheduling system in use at the Juravinski Cancer Centre, which is considerably less sophisticated than CHARM, used at the Odette Cancer Centre. In addition, as mentioned earlier the Juravinski problems are harder to solve than the problems at Odette, likely due to more constrained pharmacy resources and a higher number of appointments with longer durations. Because the problems are harder to solve, the resulting solutions are further from optimal indicating that there is more room for improvement, some of which can be accomplished through shifting.

7.4.2 Conclusions

In both problem applications, there is an improvement in makespan shown when appointments are added and the start times of appointments are allowed to shift.

7.5 Conclusion

Through the experiments at the Juravinski Cancer Centre, we validated the results of the testing done using the Odette Cancer Centre data in previous chapters. For both real-world problems, the method defining how to evaluate problem definitions is useful in analyzing the trade-off between solvability and quality. The *CRS* model was chosen for both the Odette Cancer Centre and the Juravinski Cancer Centre. This result indicates that in both centres the chart review stage of the treatment process is not important from a scheduling perspective. However, our experimental results show that considering all other resource intensive aspects of the treatment process (e.g., pharmacy, setting-up, and drug delivery) is important to achieve efficient use of clinic resources.

Dynamic template scheduling results in solutions with considerably better makespans than current practice at both centres. Additionally, allowing already scheduled appointment start times to shift by a pre-determined amount can help to accommodate additions to and, in the case of the Odette Cancer Centre data, cancellations from, the schedule.

In this chapter, we showed that the distribution of appointments at the two cancer centres is statistically different. By evaluating the proposed techniques to a second application with different characteristics, we demonstrated that the same process can be applied to both centres. Therefore, the methods are likely to be generalizable to other chemotherapy centres.

Chapter 8

[redacted] Conclusion

The central thesis of this dissertation is that a dynamically updated mathematical optimization model can be used to solve the chemotherapy outpatient scheduling problem. This dissertation has presented a technique for scheduling complex, real-world problems: dynamic template scheduling. The technique has been tested on two cases of chemotherapy outpatient scheduling problems and has demonstrated the potential for using optimization on this problem. A dry run of the system including cancellations and an algorithm which shifts appointment start times further showed the potential of the proposed technique. In the following sections we present the challenges that remain in order to solve the complete chemotherapy outpatient scheduling problem and future directions and extensions of this work.

8.1 Challenges

This section describes challenges that remain in order to solve the complete chemotherapy outpatient scheduling problem and that may make the proposed solution in its current form difficult to adopt for future researchers and practitioners. The following paragraphs outline weaknesses of the solution presented in this dissertation due to data and information availability, model structure and design, and methodology.

Due to information that was unavailable to us, either due to patient confidentiality limitations or because it is unknown to the Odette Cancer Centre and chemotherapy units in general, the following features were omitted from the problem application defined in this dissertation: (1) Chairs and beds should be modelled separately because some patients require a bed, while others can be treated in a chair or bed. The cancer centre generally does not know which patients require the use of a bed before they arrive, so we have modelled chairs and beds as a single resource. (2) Patients with multiple treatments in the same week should be scheduled at the start of the day because their drugs can be pre-mixed. We did not know which patients had multiple appointments, so we could not include this feature. (3) Patients with earlier appointments on the same day, prior to their chemotherapy appointment, should be scheduled after the time that their previous appointments are scheduled to end. (4) Patients who are receiving multiple drugs in a single treatment are listed in our data as two or more different patients.

All of this information is important to consider when scheduling. However, the schedules we produced have been validated by the users at the cancer centre. Their opinion is that even without taking into account this information, the results show that dynamic template scheduling will be useful to include in their scheduling system.

There are several weaknesses in the model structure and design. The first is the representation of the pharmacy stage. In many of our experiments, the pharmacy stage is modelled as having a duration of 30 minutes for all patients while, in reality, the stage can be from 10 to 60 minutes in duration. The second weakness is that treatment length is assumed to be deterministic. Although the estimates of treatment length are considered to be very accurate, there are patients that take longer due to complications during set up and to allergic reactions. We also assume that all patients will arrive on time for their appointments. In reality, patients may be late due to delays in previous appointments, traffic, and many other unknown reasons.

A weakness in the methodology is that a single day of chemotherapy is modeled in isolation. Although the single-day problem is, in itself, a complex problem, it sits within a broader scheduling problem that involves deciding which appointments to service on which days. Each chemotherapy regimen contains more than one treatment and the individual treatments are often connected by constraints that specify minimum and maximum time gaps between appointments. A better overall schedule may be achievable by considering the chemotherapy scheduling problem in its entirety.

8.2 Future Directions and Extensions

Future work can be categorized into three broad categories: (1) extensions, to make the system more robust and to address the weaknesses listed in the previous section; (2) adaptations that would make the system more generalizable and easier to adapt to different problems; and (3) further experimentation and testing of the system.

Many of the weaknesses listed in the previous section could be addressed as extensions of this research. As mentioned, the pharmacy stage has been greatly simplified. The pharmacy problem is complex because it involves scheduling of resources as well as constraints for specific drugs (e.g., for some drugs, only a limited number of batches can be mixed each day). At the Odette Cancer Centre, some of these constraints are built into the booking system; however, the information is not used to optimize the appointment schedule. One direction for future work is to incorporate the unique pharmacy constraints. Other possible extensions include modeling the uncertainty in treatment lengths and removing the assumption that patients will arrive on time for their appointments. A large extension, as noted above, would be to include the dynamic template scheduling system within a broader model that addresses the question of which appointments to schedule on which days.

The dynamic template scheduling algorithm can be made more adaptable to different organizations and easier to interface with existing systems. The current optimization model is simple but still requires additional knowledge in order to create and adapt it. Traditional software developers, who create and maintain the booking and administration software in a chemotherapy centre, do not know how to write mathematical programs or create constraint models using specialized software. Additionally, the software that was used in this dissertation, Comet, is no longer available from a commercial or academic source. Although the cancer centre could implement a constraint programming model using another software platform such as IBM ILOG CP Optimizer, which is not their preference. An alternative would be to use a heuristic in place of the optimization model. A heuristic can be easily programmed using any programming language. However, we are left with the question of what heuristic to use and if it will provide schedules of comparable quality to the optimization model. Recall that the underlying deterministic problem is challenging even for advanced techniques such as mixed-integer programming (Chapter 4). As a very interesting direction for future work, a researcher could develop several heuristics and compare the performance to the created constraint model.

Another direction for future work is to do further testing of dynamic template scheduling. The experiments presented in this dissertation use data from two chemotherapy centres, but there were no tests done within the existing scheduling systems. Future work could be to test the technique within the current system in use at the Odette Cancer Centre. The dry run was a step in this direction, but building dynamic template scheduling into the scheduling system would show what adaptations need to be made in order for the system to work well at the centre. Dynamic template scheduling could also be tested at more chemotherapy centres with different sizes and mixes of regimens. As well, the system is meant to be a general technique that could be applied to applications with similar features. It would be a very interesting direction for future work to test the technique on a different application that contains uncertainty due to real time arrival of requests.

8.3 Summary of Contributions

This dissertation studied the chemotherapy outpatient scheduling problem and proposed a technique to solve the problem and other problems like it. Dynamic template scheduling was demonstrated on two cases of chemotherapy outpatient scheduling and a dry run of the system was conducted. The contributions of this research are as follows:

- The chemotherapy outpatient scheduling problem was defined within the broader domain of scheduling research, a contribution to health care scheduling applications.
- A method for creating and evaluating problem definitions was developed and used to create a model of the chemotherapy problem, a contribution to the operations research literature and to health care scheduling research.
- Constraint programming was applied to a real-world health care scheduling problem, a contribution to operations research literature and constraint programming.
- Trends in appointment request data were statistically analyzed and used within the scheduling framework, a contribution to health care scheduling research.
- Dynamic template scheduling was introduced to handle real-time uncertainty arising from appointment requests that must be scheduled as they arrive, a contribution to scheduling and health care research.
- A shifting algorithm was created to address last minute uncertainty, a contribution to health care scheduling research.
- The experiments were conducted at two chemotherapy centres to show generalizability of the scheduling techniques, a contribution to health care scheduling research.

References

Adams J, Balas E, and Zawack D. (1988). The Shifting Bottleneck Procedure for Job Shop Scheduling. *Management Science,* Volume 34, pp. 391-401.

Adan I and Vissers J. (2002), Patient Mix Optimization in Hospital Admission Planning: A Case Study. *International Journal of Operations and Production Management*, Volume 22, Number 4, pp. 445-461.

Andersson K. (1998). Components for Integrating Heuristic and Model-Based Diagnosis. in the *Proceedings of the Knowledge Acquisition Workshop '98*.

Aytung H, Lawley M, McKay K, Mohan S, and Uzsoy R. (2005). Executing Production Schedules in the Face of Uncertainties: A Review and Some Future Directions. *European Journal of Operational Research*, Volume 161, Number 1, pp. 86-110.

Baptiste P and Le Pape C. (1997). Constraint Propagation and Decomposition Techniques for Highly Disjunctive and Highly Cumulative project Scheduling Problems. In the *Proceedings of the Third International Conference on Principles and Practice of Constraint Programming (CP'97)*.

Baptiste P, LePape C, and Nuijten W. (2001). Constraint-Based Scheduling. Kluwers editions.

Bartak R. (1999). Constraint Programming: In Pursuit of the Holy Grail, in the *Proceedings of the Week of Doctoral Students '99.* Prague.

Bartak R, Salido M, and Rossi F. (2010). Constraint Satisfaction Techniques in Planning and Scheduling. *Journal of Intelligent Manufacturing.*Volume 21. pp. 5-15.

Basnet C, Tang G, and Yamaguchi T. (2001). A Beam Search Heuristic for Multi-Mode Single Resource Constrained Project Scheduling. In the *Proceedings of the 36th Annual Conference of the Operational Society of New Zealand.* 8 pgs.

Beck J and Fox M. (1998). A Generic Framework for Constraint-Directed Search and Scheduling. *AI Magazine.* Volume 19, Number 4. pp. 103-132.

Beldiceanu N and Demassey S. (2010) Global Constraint Catalogue. http://www.emn.fr/z-info/sdemasse/gccat/. Accessed 31 October 2011.

Ben-Daya M and Al-Fawzan M. (1998), A Tabu Search Approach for the Flow Shop Scheduling Problem. *European Journal of Operational Research,* Volume 109, pp. 88 -95.

Berthold T, Heinz S, Lubbecke M, Mohring R, and Schulz J. (2010). A Constraint Integer Programming Approach for Resource Constrained Project Scheduling. In the *Proceedings of the International Conference on Integration of AI and OR Techniques in Constraint Programming*, pp. 313-317.

Bidot J, Vidal T, Laborie P, Beck J. (2009). A Theoretic and Practical Framework for Scheduling in a Stochastic Environment. *Journal of Scheduling,* Volume S, Number 3, pp. 315-344.

Billaut J and Roubellat F. (1996), A New Method for Workshop Real Time Scheduling. *International Journal of Production Research,* Volume 34, Number 6, pp. 1555–1579.

Blazewicz J, Lenstra J, and Kan J. (1983). Scheduling Subject to Resource Constraints. *Discrete Applications of Mathematics,* Volume 5, pp. 11-24.

Borodin A and El-Yaniv R. (1998), *Online Computation and Competitive Analysis.* Cambridge University Press, New York, NY.

Brah S and Wheeler G. (1998). Comparison of Scheduling Rules in a Flow Shop with Multiple Processors: A Simulation. *Simulation,* Volume 71, Number 5, pp. 302-311.

Brucker P, Drexl A, Mohring R, Neumann K, and Pesch E. (1999). Resource-Constrained Project Scheduling: Notation, Classification, Models, and Methods. *European Journal of Operational Research*. Volume 112, pp. 3-41.

Buscher U and Shen L. (2009). MIP-Based Approaches for Solving Scheduling Problems with Batch Processing Machines. In the *Proceedings of the Eighth International Symposium on Operations Research and Its Applications.* pp. 132-139.

Cardoen B, Demeulemeester E, and Belien J. (2009), Operating Room Planning and Scheduling: A Literature Review. *European Journal of Operational Research*. Volume 201, Number 3, pp. 921-932.

Cayirli T and Veral E. (2003), Outpatient Scheduling in Health Care: A Review of Literature, *Production and Operations Management*, Volume 12, Number 4, pp. 519.

Cayirli T, Veral E, and Rosen H. (2006), Designing Appointment Scheduling Systems for Ambulatory Care Services. *Health Care Management Science*, Volume 9, pp. 47–58.

Conforti D, Guerrierco F, and Guido R. (2008), Optimization Models for Radiotherapy Patient Scheduling. *4OR*, Volume 6, pp. 263-278.

Daniels R and Carrillo J. (1997), b-Robust Scheduling for Single-Machine Systems With Uncertain Processing Times. *IIE Transactions*. Volume 29, pp. 977–985.

Davenport A, Gefflot C, and Beck J. (2001). Slack-Based Techniques for Robust Schedules. In the *Proceedings of the Sixth European Conf. Planning*, pp. 7-18.

Davenport A and Beck J. (1999). A Survey of Techniques for Scheduling with Uncertainty, *Unpublished manuscript.*

De Reyck B and Herroelen W. (1998), A Branch-and-Bound Algorithm for the Resource-Constrained Project Scheduling Problem With Generalized Precedence Relations, *European Journal of Operational Research*, Volume 111, Number 1, pp. 114-128.

Dechter B and Dechter A. (1988). Belief Maintenance in Dynamic Constraint Networks. In the *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI'88)*. pp. 37-42.

Demir Y and Kursat Isleyen S. (2012). Evaluation of Mathematical Models for Flexible Job-Shop Scheduling Problems. *Applied Mathematical Modelling*, 2012.

Denton B, Viapiano J, and Vogl A. (2007), Optimization of Surgery Sequencing and Scheduling Decisions Under Uncertainty. *Health Care Management Science*, Volume 10, Number 1, pp. 13 - 24.

Dessouky M, Dessouky M, and Verma S. (1998). Flowshop Scheduling with Identical Jobs and Uniform Parallel Machines. *European Journal of Operational Research*, Volume 109, Number 3, pp. 620-631.

Erdogen S and Denton B. (2011). Dynamic Appointment Scheduling of a Stochastic Server with Uncertain Demand. *Informs Journal on Computing, Articles in Advance*. pp. 1-17.

Fattahi P, Mehrabad M, and Jolai F. (2007). Mathematical Modeling and Heuristic Approaches to Flexible Job Shop Scheduling Problems. *Journal of Intelligent Manufacturing.* Volume 18, Number 3, pp. 331-342.

Fleszar K and Hindi K. (2004). Solving the Resource-Constrained Project Scheduling Problem by a Variable Neighbourhood Search. *European Journal of Operational Research,* Volume 155, pp. 402-413.

Framanin J, Gupta J, and Leisten R. (2004). A Review and Classification of Heuristics for Permutation Flow-Shop Scheduling with Makespan Objective. *Journal of the Operational Research Society.* Volume 55, pp. 1243-1255.

Frisch A, Jefferson C, Hernandez B, and Miguel I. (2005) The Rules of Constraint Programming. In the *Proceedings of the 19th International Joint Conference on Artificial Intelligence.* pp. 109-116.

Garey M and Johnson D (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman.

Godard D, Laberthe P, and Nuijten W. (2005). Randomized Large Neighborhood Search for Cumulative Scheduling. *ICAPS05.*

Gonclaves J, Mendes J, and Resende M. (2008). A Genetic Algorithm for the Resource Constrained Multi-Project Scheduling Problem. *European Journal of Operational Research,* Volume 189, Number 3, pp. 1171-1190.

Grabowski J and Wodecki M. (2004), A Very Fast Tabu Search Algorithm For the Permutation Flowshop Problem With Makespan Criterion. *Computers & Operations Research*, Volume 31, pp. 1891–1909.

Graves S. (1981). A Review of Production Scheduling. *Operations Research*. Volume 29, Number 4, pp. 646-675.

Gueret C, Jussien N, and Prins C. (2000). Using Intelligent Backtracking to Improve Branch and Bound Methods: An Application to Open-Shop Problems. *European Journal of Operational Research*. Volume 127, Number 2, pp. 344-354.

Gupta J and Tunc E. (1994), Scheduling A Two-Stage Hybrid Flowshop with Separable Set-Up and Removal Times. *European Journal of Operational Research*, Volume 77, pp. 415–428.

Haouari M, Hidri L, and Gharbi A. (2006). Optimal Scheduling of a Two-Stage Hybrid Flow Shop. *Mathematical Methods of Operations Research,* Volume 64, pp. 107-124.

Hejazi R and Saghafian S. (2005). Flowshop-Scheduling Problems with Makespan Criterion: A Review. *International Journal of Production Research,* Volume 43, Number 14, pp. 2895–2929.

Herrman J. (2006) *Handbook of Production Scheduling*. Springer New York.

Herroelen W and Leus R. (2001). On the Merits and Pitfalls of Critical Chain Scheduling. *Journal of Operations Management*, Volume 19, pp. 559–577.

Herroelen W and Leus R. (2004). Robust and Reactive Project Scheduling: A Review and Classification of Procedures. *International Journal of Production Research*, Volume 42, Number 8, pp. 1599–1620.

Hong T, Huang P, Horng G, and Wang C. (2007), Three Algorithms For Flexible Flow-Shop Scheduling. *American Journal of Applied Sciences*, Volume 4. Number 11, pp. 889 – 897.

Hoogeveen J, Censtra J, and Veltman B. (1996) Preemptive Scheduling in a 2-Stage Multi-Processor Flowshop is NP-hard. *European Journal of Operational Research*. Volume 89. pp. 172-175.

Hsu V, de Matta R, and Lee C. (2003), Scheduling Patients in an Ambulatory Surgical Center. *Naval Research Logistics*, Volume 50, pp. 218-238.

ILOG. ILOG CPLEX User Manual. Available at: http://yalma.fime.uanl.mx/cplex11-manual/Content/Optimization/Documentation/CPLEX/_pubskel/XPlatform/User_man250.html. Accessed on April 15, 2013.

Jeffries R. (2001). What is Extreme Programming. Web document in *XProgramming.com an Agile Software Developement Resource*. available at: http://xprogramming.com/xpmag/whatisxp.htm. Accessed on April 22, 2007.

Kolisch R, Sprecher A, and Drexl A. (1995). Characterization and Generation of a General Class of Resource-Constrained Project Scheduling Problems. *Management Science,* Volume 41, Number 10, pp. 1693-1703.

Kone O, Artigues C, Lopez P, and Mongeau M. (2011). MILP Models for Resource-Constrained Project Scheduling Problems. *Computers and Operations Research,* Volume 38, pp. 3-13.

Kouvelis P, Daniels R, and Vairaktarakis G. (2000), Robust Scheduling of a Two-Machine Flow Shop With Uncertain Processing times. *IIE Transactions,* Volume 32, pp. 421–432.

Ku W. (2013). MIP and CP Models for Flexible Job Shop Scheduling Problems. *Working Paper.*

Lambrechts O, Demeulemeester E, and Herroelen W. (2008), Proactive and Reactive Strategies for Resource-Constrained Project Scheduling With Uncertain Resource Availabilities. *Journal of Scheduling*, Volume 11, Number 2, pp. 121-136.

Li T, Li Y, Zhang Q, Gao X, and Dai S. (2005). Constraint Programming Approach to Steelmaking-Making Process Scheduling. *Communications of the IIMA,* Volume 5, Number 3, pp. 17-24.

Li Z and Ierapetritou M. (2008). Reactive Scheduling Using Parametric Programming. *American Institute of Chemical Engineers Journal*, Volume 54, Number 10, pp. 2610-2623.

Lu M and Li H. (2003). Resource-Activity Critical-Path Method for Construction Planning. *Journal for Constructing Engineering and Management,* Volume 129, pp. 412-420.

Macal C. (2005). Model Verification and Validation. Workshop presented by *the Argone national Laboratory*, University of Chicago.

Marcon E and Dexter F. (2006). Impact of Surgical Sequencing on Post Anesthesia Care Unit. *Health Care Management Science*, Volume 9, pp. 87 - 98.

Mehta S and Uzsoy R. (1998). Predictable Scheduling of a Job Shop Subject to Breakdowns. *IEEE Transactions on Robotics and Automation.* Volume 14, Number 3, pp. 365-378.

Mohring R, and Stork F. (2000). Linear Preselective Policies for Stochastic Project Scheduling. *Mathematical Methods of Operations Research*, Volume 52, pp. 501–515.

Museros L and Escrig M. (2003). A Qualitative Theory for Shape Representation and Matching. in the *Proceedings of The Qualitative Reasoning Workshop*.

Muthuraman K and Lawley M. (2008), A Stochastic Overbooking Model for Outpatient Clinical Scheduling With No-Shows. *IIE Transactions*.

Naderi B, Ruiz R, and Zandieh M. (2009). Algorithms For A Realistic Variant of Flowshop Scheduling. *Computers and Operations Research*, Volume 37, Number 2, pp. 236-246.

Nawaz M, Enscore E, and Ham I. (1983). A Heuristic Algorithm for the m-Machine, n-Job Flow-Shop Sequencing Problem. *Omega*, Volume 11, pp. 91-95.

Nowicki E and Smutnicki C. (1998). The Flow Shop With Parallel Machines: A Tabu Search Approach. *European Journal of Operational Research*. Volume 106, pp. 226 – 253.

Nunes de Barros L, Lemos M, Volnys B, and Wainer J. (1999). Model Based Diagnosis for Network Communication Faults. In *the Proceedings of the Third International Workshop on Artificial Intelligence in Distributed Information Networking.* pp. 57-62.

O’Donovan R, McKay K, and Uzsoy R. (1999), Predictable Scheduling on a Single Machine With Machine Breakdowns and Sensitive Jobs. *International Journal of Production Research,* Volume 37, pp. 4217–4233.

Ouelhadj D and Petrovic S. (2009). A survey of Dynamic Scheduling in Manufacturing Systems. *Journal of Scheduling*. Volume 12, Number 4, pp. 417-431.

Patterson J. (1984). A Comparison of Exact Approaches for Solving the Multiple Constrained Resource Project Scheduling Problem. *Management Science*, Volume 30, Number 7, pp. 854-867.

Petrovic S, Leung W, Song X, and Sundar S. (2006). Algorithms for Radiotherapy Treatment Booking, in *the Proceedings of the 25th Workshop of the UK Planning and Scheduling Special Interest Group*, pp. 105-112.

Pinedo M. (2005). *Planning and Scheduling in Manufacturing and Services*. Springer Science and Business Media Inc.

Pinedo M. (2012). *Scheduling: Theory, Algorithms, and Systems,* 4th Edition. Springer.

Powell S and Baker K. (2013). *Management Science: The Art of Modeling With Sreadsheets 4th Edition.* Wiley.

Pressman R. (2000) *Software Engineering: A Practitioner's Approach.5th edition*, Ch.10-11. McGraw-Hill Higher Education.

Prosser P. (1993). Hybrid algorithms for the constraint satisfaction problem, *Computational Intelligence*. Volume 9, Number 3, pp. 268-299.

Rad S, Ruiz R, and Boroojerdian N. (2006). New High Performing Heuristics for Minimizing Makespan in Permutation Flowshops.

Rossi F, van Beek P, and Walsh T. (2006). *Constraint Programming*. Chapter 1. Elsevier.

Ruiz R and Vazquez-Rodriguez J. (2009). The Hybrid Flow Shop Scheduling Problem. *European Journal of Operational Research*, Volume 205, Number 1, pp. 1-18.

Sacchi W. (2001). Process Models in Software Engineering. in Marciniak, J.J. (ed.), *Encyclopedia of Software Engineering*, 2nd Edition. John Wiley and Sons, Inc.

Santibanez P, Aristizabal R, Puterman M, Chow V, Huang W, Kollmannsberger C, Nordin T, Runzer N, and Tyldesley S. (2012) Operations Research Methods Improve Chemotherapy

Patient Appointment Scheduling. *The Joint Commission Journal on Quality and Patient Safety*. Volume 38, Number 12, pp. 541-553.

Santibanez P, Chow V, French J, Puterman M, and Tyldesley S. (2009) Reducing Patient Wait Times and Improving Resource Utilization at British Columbia Cancer Agency's Ambulatory Care Unit Through Simulation. *Heal Care Management Science.* Volume 12, Number 4. pp 392 – 407.

Santibanez P. (2009), Improving the Chemotherapy Scheduling Process at BCCA Vancouver Centre. *Working Paper*.

Sherali H, Sarin S, and Kodialam M. (1990). Models and Algorithms for a Two-Stage Production Process. *Production Planning and Control: The Management of Operations,* Volume 1, Number 1, pp. 27-39.

Siegel. (1956). *Nonparametric Statistics For the Behavioural Sciences*. New York. McGraw-Hill.

Smith B and Grant S. (1998). Trying Harder to Fail First. in the *Proceedings of the* Thirteenth European Conference on Artificial Intelligence (ECAI 98). pp. 249-253.

Taha A. (2007) *Operations Research: An Introduction*, 8th Edition, Chapter 2. Prentice Hall.

Taillard E. (1993). Benchmarks for Basic Scheduling Problems. *European Journal of Operational Research,* Volume 64, pp. 278-285.

Thomas P and Salhi S. (1998). A Tabu Search Approach for the Resource Constrained Project Scheduling Problem. *Journal of Heuristics,* Volume 4, pp. 123-139.

Turkcan A, Zeng B, and Lawley M. (2012). Chemotherapy Operations Planning and Scheduling. *IIE Transactions on Health care Systems Engineering*. Volume 2, Number 1, pp. 31-49.

Van Hentenryck P. (2008). Comet CP Tutorial. *UCL*.

Van Hentenryck P and Bent R. (2006), *Online Stochastic Combinatorial Optimization.* MIT Press, Cambridge, Massachusetts.

Van Hentenryck P, Bent R, Mercier L, and Vergados Y. (2009) Online Stochastic Reservation Systems. *Annals of Operations Research*. Volume 171, Number 1, pp.101-126,

Van Houdenhoven M, Hans E, Klein J, Wullink G, and Kazemier G. (2007), A Norm Utilization for Scarce hospital Resources: Evidence From Operating Rooms in a Dutch University Hospital. *Journal of Medical Systems*, Volume 31, pp. 231 - 236.

Vissers J and Wijngaard J. (1979), The Outpatient Appointment System: Design of a Simulation Study. *European Journal of Operational Research*, Volume 3, pp. 459 – 463.

Wu, C, Brown K, and Beck J. (2009), Scheduling with uncertain durations: modelling beta-robust scheduling with constraints**.** *Computers and Operations Research*, Volume 36, Number 8, pp 2348—2356.

Yang B, Geunes J, and O'Brien W. (2001). Resource-Constrained Project Scheduling: Past Work and New Directions. *Department of Industrial and Systems Engineering, University of Florida.*

Appendix A: Constraint Programming Models – Explicit Definitions

CRS

min S_{End}

Subject to:

cumulative $(S_{2g}, D_{2g}, 1, \|DR\|)$,	$\forall g \in G$	(1)
cumulative $(S_{4g}, D_{4g}, 1, \|C\|)$,	$\forall g \in G$	(2)
cumulative $(S_{3g}, D_{3g}, 1, \|N3\|)$,	$\forall g \in G$	(3)
cumulative $(S_{4g}, D_{4g}, 1, \|N4\|)$,	$\forall g \in G$	(4)
$S_{1r} \geq B_r$,	$\forall r \in R$	(5)
$S_{3r} \geq S_{2r} + D_{2r}$,	$\forall r \in R$	(6)
$S_{4r} = S_{3r}$,	$\forall r \in R$	(7)
$S_{End} \leq \|T\|$,		(8)
$S_{4g} + D_{4g} \leq S_{End}$,	$\forall g \in G$	(9)
$S_{3r} \leq S_{2r} + D_{2r} + l$,	$\forall r \in R$	(10)
$S_{if} = S_{if}^{*}$,	$\forall f \in F, \forall i, i = 2,3,4$	(11)

CRS-PS

min S_{End}

Subject to:

cumulative $(S_{4g}, D_{4g}, 1, \|C\|)$,	$\forall g \in G$	(1)
cumulative $(S_{3g}, D_{3g}, 1, \|N3\|)$,	$\forall g \in G$	(2)
cumulative $(S_{3g}, D_{3g}, 1, \|N4\|)$,	$\forall g \in G$	(3)
$S_{3r} \geq B_r$,	$\forall r \in R$	(4)
$S_{4r} = S_{3r}$,	$\forall r \in R$	(5)
$S_{End} \leq \|T\|$,		(6)
$S_{4g} + D_{4g} \leq S_{End}$,	$\forall g \in G$	(7)
$S_{if} = S_{if}^{*}$,	$\forall f \in F, \forall i, i = 3,4$	(8)

CRS-RUS

The model definition is identical to *CRS-PS*. However, the available chairs, |C| changes over the course of the day. For the first 90 minutes there are 12 chairs, for the next 45 minutes there are 21 chairs, and for the remainder of the day there are 29 chairs available.

Appendix B: Choosing an Optimization Model – Complete Experimental Results for the Odette Cancer Centre

The following tables provide the solvability results for each model:

Base:

Test Set	**Time to Solve (s)**	**Solved to Optimality**
Day 1	6.52	Y
Day 2	3600.38	N
Day 3	3.59	Y
Day 4	3600.48	N
Day 5	3600.45	N
Average	2162.28	40 %

CRS:

Test Set	**Time to Solve (s)**	**Solved to Optimality**
Day 1	0.09	Y
Day 2	3600.03	N
Day 3	0.09	Y
Day 4	0.28	Y
Day 5	0.17	Y
Average	720.13	80 %

CRS-PS:

Test Set	**Time to Solve (s)**	**Solved to Optimality**
Day 1	0.02	Y
Day 2	0.02	Y
Day 3	0.03	Y
Day 4	0.06	Y
Day 5	0.05	Y
Average	0.03	100 %

CRS-RUS:

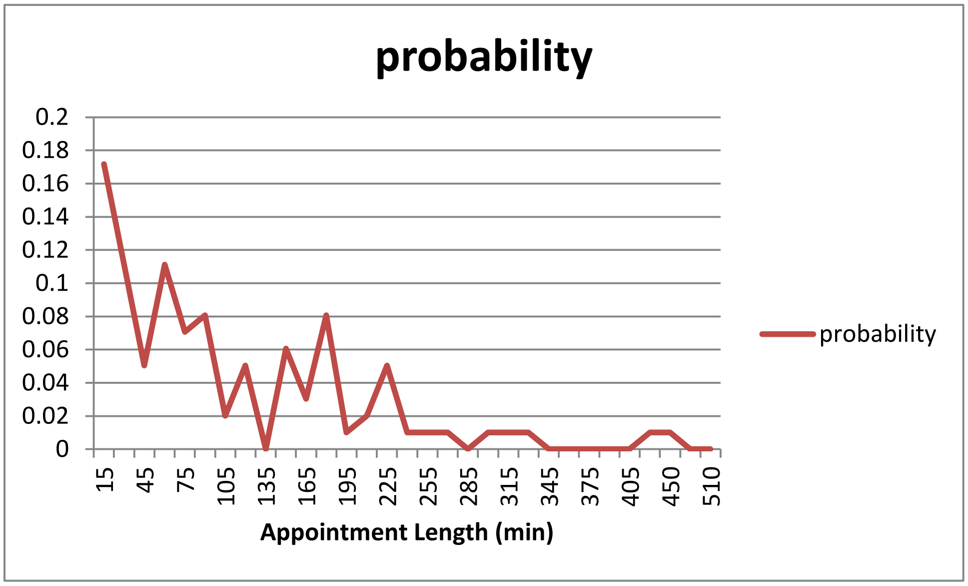
Test Set	Time to Solve (s)	Solved to Optimality
Day 1	0.64	Y
Day 2	446.41	Y
Day 3	0.03	Y
Day 4	0.06	Y
Day 5	0.06	Y
Average	89.44	100%

Appendix C: The Empirical Distribution

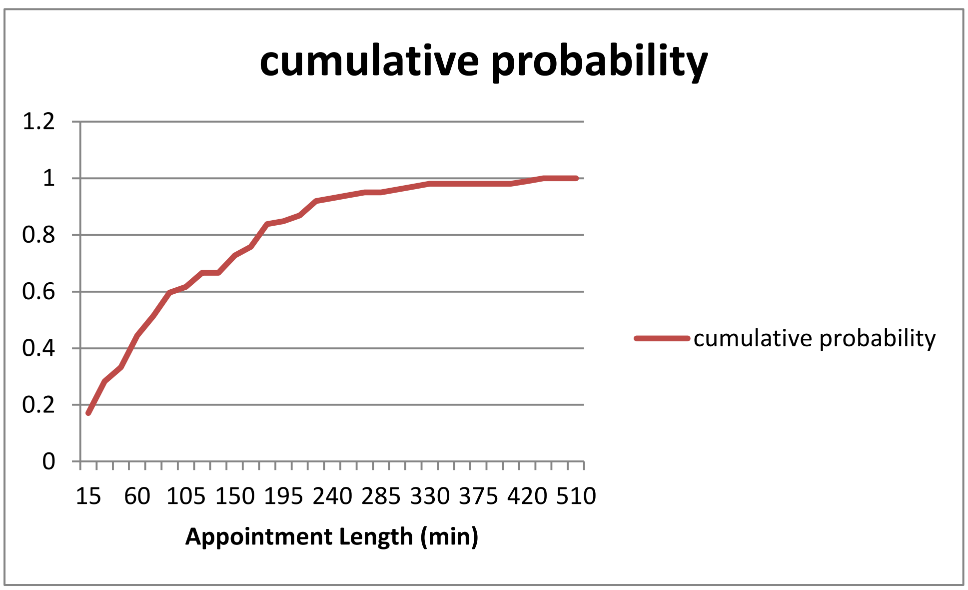
The average number of appointments of each length for the three month period of January through March 2010 was used to create the empirical distribution. The table below provides the average number of appointments of each length, the associated probability and cumulative probability. When sampling the distribution, random numbers are drawn between one and 100. If the number falls within the range provided in the table, the corresponding appointment length is used in the sample.

Appointment Length	Average Number	Probability	Cumulative Probability	Sampling Range
15	17	0.171717	0.171717	0 – 17
30	11	0.111111	0.282828	17 – 28
45	5	0.050505	0.333333	29 – 33
60	11	0.111111	0.444444	34 – 44
75	7	0.070707	0.515152	44 – 51
90	8	0.080808	0.59596	52 – 59
105	2	0.020202	0.616162	60 -61
120	5	0.050505	0.666667	62-66
150	6	0.060606	0.727273	67 – 72
165	3	0.030303	0.757576	73 – 75
180	8	0.080808	0.838384	76 – 83
195	1	0.010101	0.848485	84
210	2	0.020202	0.868687	85 – 86
225	5	0.050505	0.919192	87 -91
240	1	0.010101	0.929293	92
255	1	0.010101	0.939394	93
270	1	0.010101	0.949495	94
300	1	0.010101	0.959596	95
315	1	0.010101	0.969697	96
330	1	0.010101	0.979798	97
420	1	0.010101	0.989899	98
450	1	0.010101	1	99-100

The following graphs show the probability density function and the cumulative density function of the empirical distribution.



Probability Density Function



Cumulative Density Function

Appendix D: Shifting – Complete Experimental Results for the Odette Cancer Centre

Adding in Appointments:

Dynamic template starting schedule with 15 minute shift time:

Test Set	**Added Appointments**	**Difference in Makespan (min)**
Day 1	One 120 minute	15
Day 2	One 120 minute	0
Day 3	One 120 minute	15
Day 4	One 120 minute	15
Day 5	One 120 minute	15
Day 1	Four 120 minutes	15
Day 2	Four 120 minutes	0
Day 3	Four 120 minutes	15
Day 4	Four 120 minutes	15
Day 5	Four 120 minutes	15
Day 1	One four hour	15
Day 2	One four hour	0
Day 3	One four hour	0
Day 4	One four hour	15
Day 5	One four hour	15
Day 1	Two four hours	15
Day 2	Two four hours	0
Day 3	Two four hours	0
Day 4	Two four hours	15
Day 5	Two four hours	15

Dynamic template starting schedule with 30 minute shift time:

Test Set	**Added Appointments**	**Difference in Makespan (min)**
Day 1	One 120 minute	30
Day 2	One 120 minute	0
Day 3	One 120 minute	15
Day 4	One 120 minute	15
Day 5	One 120 minute	15
Day 1	Four 120 minutes	30
Day 2	Four 120 minutes	0
Day 3	Four 120 minutes	15
Day 4	Four 120 minutes	15
Day 5	Four 120 minutes	15
Day 1	One four hour	30
Day 2	One four hour	0
Day 3	One four hour	0
Day 4	One four hour	15
Day 5	One four hour	15
Day 1	Two four hours	30
Day 2	Two four hours	0
Day 3	Two four hours	0
Day 4	Two four hours	15
Day 5	Two four hours	15

Dynamic Template Schedule with 60 minute shift time:

Test Set	**Added Appointments**	**Difference in Makespan (min)**
Day 1	One 120 minute	60
Day 2	One 120 minute	0
Day 3	One 120 minute	15
Day 4	One 120 minute	15
Day 5	One 120 minute	15
Day 1	Four 120 minutes	60
Day 2	Four 120 minutes	0
Day 3	Four 120 minutes	15
Day 4	Four 120 minutes	15
Day 5	Four 120 minutes	15
Day 1	One four hour	60
Day 2	One four hour	0
Day 3	One four hour	15
Day 4	One four hour	15
Day 5	One four hour	15
Day 1	Two four hours	60
Day 2	Two four hours	0
Day 3	Two four hours	15
Day 4	Two four hours	15
Day 5	Two four hours	15

Dynamic template starting schedule with 30/15 minute shift time:

Test Set	**Added Appointments**	**Difference in Makespan (min)**
Day 1	One 120 minute	30
Day 2	One 120 minute	0
Day 3	One 120 minute	15
Day 4	One 120 minute	15
Day 5	One 120 minute	15
Day 1	Four 120 minutes	30
Day 2	Four 120 minutes	0
Day 3	Four 120 minutes	15
Day 4	Four 120 minutes	15
Day 5	Four 120 minutes	15
Day 1	One four hour	15
Day 2	One four hour	0
Day 3	One four hour	0
Day 4	One four hour	15
Day 5	One four hour	15
Day 1	Two four hours	15
Day 2	Two four hours	0
Day 3	Two four hours	0
Day 4	Two four hours	15
Day 5	Two four hours	15

Real starting schedule with 15 minute shift time:

Test Set	**Added Appointments**	**Difference in Makespan (min)**
Day 1	One 120 minute	15
Day 2	One 120 minute	15
Day 3	One 120 minute	15
Day 4	One 120 minute	15
Day 5	One 120 minute	15
Day 1	Four 120 minutes	15
Day 2	Four 120 minutes	15
Day 3	Four 120 minutes	15
Day 4	Four 120 minutes	15
Day 5	Four 120 minutes	15
Day 1	One four hour	15
Day 2	One four hour	15
Day 3	One four hour	15
Day 4	One four hour	15
Day 5	One four hour	15
Day 1	Two four hours	15
Day 2	Two four hours	15
Day 3	Two four hours	15
Day 4	Two four hours	15
Day 5	Two four hours	15

Real starting schedule with 30 minute shift time:

Test Set	**Added Appointments**	**Difference in Makespan (min)**
Day 1	One 120 minute	30
Day 2	One 120 minute	30
Day 3	One 120 minute	30
Day 4	One 120 minute	30
Day 5	One 120 minute	30
Day 1	Four 120 minutes	30
Day 2	Four 120 minutes	30
Day 3	Four 120 minutes	30
Day 4	Four 120 minutes	30
Day 5	Four 120 minutes	30
Day 1	One four hour	30
Day 2	One four hour	30
Day 3	One four hour	30
Day 4	One four hour	30
Day 5	One four hour	30
Day 1	Two four hours	30
Day 2	Two four hours	30
Day 3	Two four hours	30
Day 4	Two four hours	30
Day 5	Two four hours	30

Real starting schedule with 60 minute shift time:

Test Set	**Added Appointments**	**Difference in Makespan (min)**
Day 1	One 120 minute	60
Day 2	One 120 minute	60
Day 3	One 120 minute	60
Day 4	One 120 minute	60
Day 5	One 120 minute	60
Day 1	Four 120 minutes	60
Day 2	Four 120 minutes	60
Day 3	Four 120 minutes	60
Day 4	Four 120 minutes	60
Day 5	Four 120 minutes	60
Day 1	One four hour	60
Day 2	One four hour	60
Day 3	One four hour	60
Day 4	One four hour	60
Day 5	One four hour	60
Day 1	Two four hours	60
Day 2	Two four hours	60
Day 3	Two four hours	60
Day 4	Two four hours	60
Day 5	Two four hours	60

Real starting schedule with 30/15 minute shift time:

Test Set	**Added Appointments**	**Difference in Makespan (min)**
Day 1	One 120 minute	30
Day 2	One 120 minute	30
Day 3	One 120 minute	30
Day 4	One 120 minute	30
Day 5	One 120 minute	30
Day 1	Four 120 minutes	30
Day 2	Four 120 minutes	30
Day 3	Four 120 minutes	15
Day 4	Four 120 minutes	30
Day 5	Four 120 minutes	30
Day 1	One four hour	30
Day 2	One four hour	30
Day 3	One four hour	30
Day 4	One four hour	30
Day 5	One four hour	30
Day 1	Two four hours	30
Day 2	Two four hours	30
Day 3	Two four hours	15
Day 4	Two four hours	30
Day 5	Two four hours	30

Run-Through Appointment List:

Item Number	Appointment Length (timeslots)	Action
1	6	Book
2	5	Book
3	1	Book
4	4	Book
5	1	Book
6	4	Book
7	4	Book
8	1	Book
9	1	Book
10	1	Book
11	4	Book
12	4	Book
13	4	Book
14	13	Book
15	13	Book
16	3	Book
17	4	Book
18	4	Book
19	8	Book
20	10	Book
21	23	Book
22	4	Book
23	5	Book
24	15	Book
25	6	Cancel
26	3	Book
27	5	Cancel
28	5	Book
29	1	Cancel
30	3	Book
31	5	Book
32	4	Book
33	15	Book
34	15	Book
35	3	Cancel
36	5	Book
37	8	Book
38	15	Book
39	20	Book
40	21	Book
41	5	Book

42	5	Cancel
43	3	Book
44	2	Book
45	5	Book
46	4	Book
47	11	Book
48	4	Cancel
49	15	Cancel
50	12	Book
51	5	Book
52	2	Book
53	5	Book
54	12	Book
55	5	Book
56	15	Cancel
57	2	Book
58	1	Book
59	4	Book
60	15	Cancel
61	4	Book
62	1	Book
63	5	Book
64	5	Book
65	11	Cancel
66	14	Book
67	12	Book
68	11	Book
69	6	Book
70	4	Book
71	14	Cancel
72	8	Book
73	1	Book
74	6	Book
75	5	Book
76	6	Book
77	6	Book
78	12	Cancel
79	6	Cancel
80	15	Book
81	5	Book
82	6	Book
83	10	Book
84	4	Cancel
85	12	Book
86	5	Book
87	7	Book

88	2	Book
89	8	Book
90	1	Book
91	5	Book
92	3	Book
93	8	Cancel
94	1	Cancel

Appendix E: Full Experimental Results for the Juravinski Cancer Centre

Choosing an Optimization Model:

The following tables provide the solvability results for each model:

Base:

Test Set	**Time to Solve (s)**	**Solved to Optimality**
Day 1	3600.61	N
Day 2	3600.88	N
Day 3	3600.81	N
Day 4	3600.84	N
Day 5	3600.70	N
Average	3600.77	0

CRS:

Test Set	**Time to Solve (s)**	**Solved to Optimality**
Day 1	3600.03	N
Day 2	0.14	Y
Day 3	0.15	Y
Day 4	3600.03	N
Day 5	3600.03	N
Average	2160.08	40 %

PO:

Test Set	**Time to Solve (s)**	**Solved to Optimality**
Day 1	3600.02	N
Day 2	0.03	Y
Day 3	0.03	Y
Day 4	3600.02	N
Day 5	3600.02	N
Average	2160.02	40 %

CRS-PS:

Test Set	Time to Solve (s)	Solved to Optimality
Day 1	0.05	Y
Day 2	0.09	Y
Day 3	0.06	Y
Day 4	0.08	Y
Day 5	0.06	Y
Average	0.07	100 %

CRS-RUS:

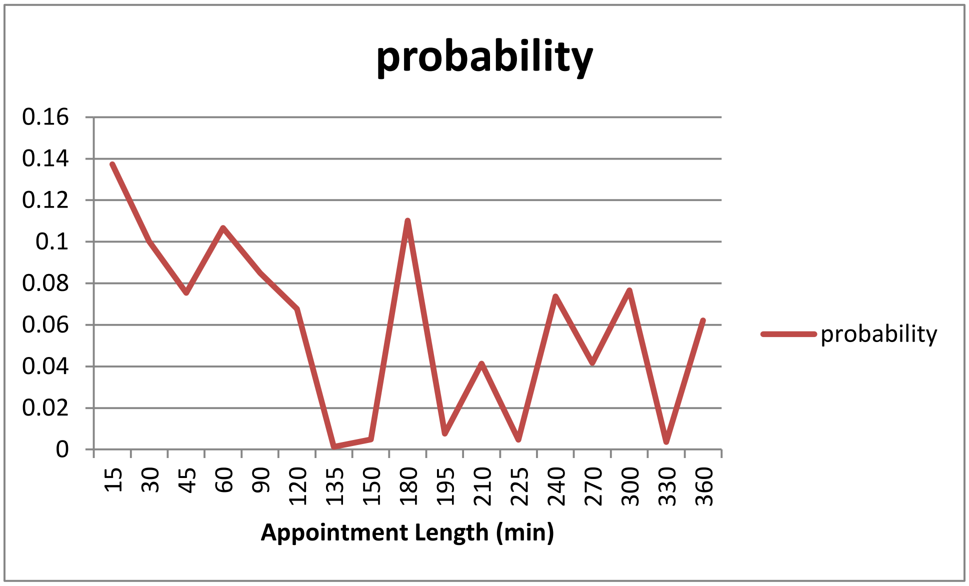
Test Set	Time to Solve (s)	Solved to Optimality
Day 1	116.41	Y
Day 2	3600.02	N
Day 3	3600.02	N
Day 4	3600.02	N
Day 5	3600.02	N
Average	2882.34	20%

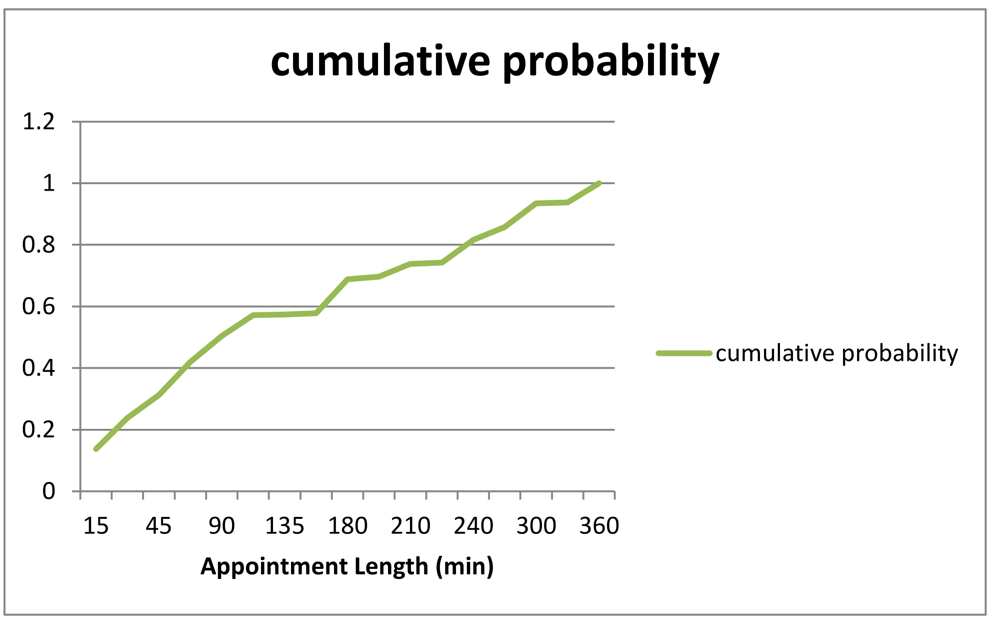
Empirical Distribution at the Juravinski Cancer Centre:

The average number of appointments of each length for the three month period of April through June 2011 was used to create the empirical distribution. The table below provides the average number of appointments of each length, the associated probability and cumulative probability. When sampling the distribution, random numbers are drawn between one and 100. If the number falls within the range provided in the table, the corresponding appointment length is used in the sample.

Appointment Length	Average Number	Probability	Cumulative Probability	Sampling Range
15	14	0.13729	0.13729	0 – 14
30	10	0.100257893	0.237547893	14 – 24
45	8	0.075351213	0.312899106	24 – 31
60	11	0.106641124	0.41954023	31 – 42
90	8	0.084929757	0.504469987	42 – 50
120	7	0.067688378	0.572158365	50 – 57
135	1	0.001277139	0.573435504	57
150	1	0.0048957	0.578331205	57 – 58
180	11	0.110259685	0.68859089	58 – 69
195	1	0.007662835	0.696253725	69 – 70
210	4	0.041294168	0.737547893	70 – 74
225	1	0.004682844	0.742230736	74
240	7	0.073648361	0.815879097	74 – 82
270	4	0.041719881	0.857598978	82 – 86
300	7	0.076628352	0.934227331	86 – 93
330	1	0.003618561	0.937845892	93 – 94
360	6	0.062154108	1	94-100

The following graphs show the probability density function and the cumulative density function of the empirical distribution.





Dynamic Template Scheduling Results:

Test Set	Day 1	Day 2	Day 3	Day 4	Day 5
Makespan 1	34	36	39	37	38
Time 1	26	333	36	327	331
Makespan 2	32	37	39	39	39
Time 2	19	332	335	324	25
Makespan 3	31	35	41	35	37
Time 3	16	641	22	946	27
Makespan 4	37	35	39	39	40
Time 4	19	336	337	327	26
Makespan 5	36	34	38	39	39
Time 5	22	327	24	635	24
Makespan 6	37	36	38	37	37
Time 6	18	639	329	334	26
Makespan 7	30	35	39	38	38
Time 7	19	326	657	326	332
Makespan 8	26	37	41	40	38
Time 8	26	634	649	329	29
Makespan 9	34	34	37	39	38
Time 9	329	328	330	327	25
Makespan 10	35	38	39	40	40
Time 10	18	330	51	333	21

Shifting Experiments:

Adding in Appointments:

Dynamic template starting schedule with 15 minute shift time:

Test Set	Added Appointments	Difference in Makespan (min)
Day 1	One 120 minute	15
Day 2	One 120 minute	15
Day 3	One 120 minute	15
Day 4	One 120 minute	15
Day 5	One 120 minute	15
Day 1	Four 120 minutes	15
Day 2	Four 120 minutes	15
Day 3	Four 120 minutes	15
Day 4	Four 120 minutes	15
Day 5	Four 120 minutes	15
Day 1	One four hour	15
Day 2	One four hour	15
Day 3	One four hour	0
Day 4	One four hour	15
Day 5	One four hour	15
Day 1	Two four hours	15
Day 2	Two four hours	15
Day 3	Two four hours	0
Day 4	Two four hours	15
Day 5	Two four hours	15

Dynamic template starting schedule with 30 minute shift time:

Test Set	**Added Appointments**	**Difference in Makespan (min)**
Day 1	One 120 minute	30
Day 2	One 120 minute	30
Day 3	One 120 minute	15
Day 4	One 120 minute	30
Day 5	One 120 minute	30
Day 1	Four 120 minutes	30
Day 2	Four 120 minutes	30
Day 3	Four 120 minutes	15
Day 4	Four 120 minutes	30
Day 5	Four 120 minutes	30
Day 1	One four hour	30
Day 2	One four hour	30
Day 3	One four hour	0
Day 4	One four hour	30
Day 5	One four hour	30
Day 1	Two four hours	30
Day 2	Two four hours	30
Day 3	Two four hours	0
Day 4	Two four hours	30
Day 5	Two four hours	30

Dynamic Template Schedule with 60 minute shift time:

Test Set	**Added Appointments**	**Difference in Makespan (min)**
Day 1	One 120 minute	60
Day 2	One 120 minute	60
Day 3	One 120 minute	15
Day 4	One 120 minute	60
Day 5	One 120 minute	60
Day 1	Four 120 minutes	60
Day 2	Four 120 minutes	60
Day 3	Four 120 minutes	15
Day 4	Four 120 minutes	60
Day 5	Four 120 minutes	60
Day 1	One four hour	60
Day 2	One four hour	60
Day 3	One four hour	0
Day 4	One four hour	60
Day 5	One four hour	60
Day 1	Two four hours	60
Day 2	Two four hours	60
Day 3	Two four hours	0
Day 4	Two four hours	60
Day 5	Two four hours	60

Dynamic template starting schedule with 30/15 minute shift time:

Test Set	**Added Appointments**	**Difference in Makespan (min)**
Day 1	One 120 minute	30
Day 2	One 120 minute	30
Day 3	One 120 minute	15
Day 4	One 120 minute	30
Day 5	One 120 minute	30
Day 1	Four 120 minutes	30
Day 2	Four 120 minutes	30
Day 3	Four 120 minutes	15
Day 4	Four 120 minutes	30
Day 5	Four 120 minutes	30
Day 1	One four hour	15
Day 2	One four hour	30
Day 3	One four hour	0
Day 4	One four hour	30
Day 5	One four hour	15
Day 1	Two four hours	30
Day 2	Two four hours	30
Day 3	Two four hours	0
Day 4	Two four hours	30
Day 5	Two four hours	30

Real starting schedule with 15 minute shift time:

Test Set	**Added Appointments**	**Difference in Makespan (min)**
Day 1	One 120 minute	15
Day 2	One 120 minute	15
Day 3	One 120 minute	15
Day 4	One 120 minute	15
Day 5	One 120 minute	0
Day 1	Four 120 minutes	15
Day 2	Four 120 minutes	15
Day 3	Four 120 minutes	15
Day 4	Four 120 minutes	15
Day 5	Four 120 minutes	0
Day 1	One four hour	15
Day 2	One four hour	15
Day 3	One four hour	15
Day 4	One four hour	15
Day 5	One four hour	0
Day 1	Two four hours	15
Day 2	Two four hours	15
Day 3	Two four hours	15
Day 4	Two four hours	15
Day 5	Two four hours	0

Real starting schedule with 30 minute shift time:

Test Set	**Added Appointments**	**Difference in Makespan (min)**
Day 1	One 120 minute	30
Day 2	One 120 minute	30
Day 3	One 120 minute	30
Day 4	One 120 minute	30
Day 5	One 120 minute	0
Day 1	Four 120 minutes	30
Day 2	Four 120 minutes	30
Day 3	Four 120 minutes	30
Day 4	Four 120 minutes	30
Day 5	Four 120 minutes	0
Day 1	One four hour	30
Day 2	One four hour	30
Day 3	One four hour	30
Day 4	One four hour	30
Day 5	One four hour	0
Day 1	Two four hours	30
Day 2	Two four hours	30
Day 3	Two four hours	30
Day 4	Two four hours	30
Day 5	Two four hours	0

Real starting schedule with 60 minute shift time:

Test Set	**Added Appointments**	**Difference in Makespan (min)**
Day 1	One 120 minute	60
Day 2	One 120 minute	60
Day 3	One 120 minute	60
Day 4	One 120 minute	60
Day 5	One 120 minute	30
Day 1	Four 120 minutes	60
Day 2	Four 120 minutes	60
Day 3	Four 120 minutes	60
Day 4	Four 120 minutes	60
Day 5	Four 120 minutes	30
Day 1	One four hour	60
Day 2	One four hour	60
Day 3	One four hour	60
Day 4	One four hour	60
Day 5	One four hour	30
Day 1	Two four hours	60
Day 2	Two four hours	60
Day 3	Two four hours	60
Day 4	Two four hours	60
Day 5	Two four hours	30

Real starting schedule with 30/15 minute shift time:

Test Set	**Added Appointments**	**Difference in Makespan (min)**
Day 1	One 120 minute	30
Day 2	One 120 minute	30
Day 3	One 120 minute	15
Day 4	One 120 minute	30
Day 5	One 120 minute	0
Day 1	Four 120 minutes	30
Day 2	Four 120 minutes	30
Day 3	Four 120 minutes	15
Day 4	Four 120 minutes	30
Day 5	Four 120 minutes	0
Day 1	One four hour	30
Day 2	One four hour	30
Day 3	One four hour	15
Day 4	One four hour	30
Day 5	One four hour	0
Day 1	Two four hours	30
Day 2	Two four hours	30
Day 3	Two four hours	15
Day 4	Two four hours	30
Day 5	Two four hours	0