MATHEMATICAL PROGRAMMING ENHANCED METAHEURISTIC APPROACH FOR SIMULATION-BASED OPTIMIZATION IN OUTPATIENT APPOINTMENT SCHEDULING

By

ALIREZA SAREMI

A Thesis submitted to the Faculty of Graduate Studies of

The University of Manitoba

in partial fulfilment of the requirements of the degree of

DOCTOR OF PHILOSOPHY

Department of Mechanical and Manufacturing

University of Manitoba

Winnipeg

Abstract

In the last two decades, the western world witnessed a continuous rise in the health expenditure. Meanwhile, complaints from patients on excessive waiting times are also increasing. In the past, many researchers have tried to devise appointment scheduling rules to provide a trade-off between maximizing patients' satisfaction and minimizing the costs of the health providers. For instance, this challenge appears appointment scheduling problems (ASP).

Commonly used methods in ASP include analytical methods, simulation studies, and combination of simulation with heuristic approaches. Analytical methods (e.g., queuing theory and mathematical programming) face challenges of fully capturing the complexities of systems and usually make strong assumptions for tractability of problems. These methods simplify the whole system to a single-stage unit and ignore the actual system factors such as the presence of multiple stages and/or resource constraints. Simulation studies, conversely, are able to model most complexities of the actual system, but they typically lack an optimization strategy to deliver optimal appointment schedules. Also, heuristic approaches normally are based on intuitive rules and do not perform well as standalone methods.

In order to reach an optimal schedule while considering complexities in actual health care systems, this thesis proposes efficient and effective methods that yield (near) optimal appointment schedules by integrating mathematical programming, a tabu search optimization algorithm and discrete event simulation. The proposed methodologies address the challenges and complexities of scheduling in real world multistage healthcare units in the presence of stochastic

service durations, a mix of patient types, patients with heterogeneous service sequence, and resource constraints.

Moreover, the proposed methodology is capable of finding the optimum considering simultaneously multiple performance criteria. A Pareto front (a set of optimal solutions) for the performance criteria can be obtained using the proposed methods. Healthcare management can use the Pareto front to choose the appropriate policy based on different conditions and priorities.

In addition, the proposed method has been applied to two case studies of Operating Rooms departments in two major Canadian hospitals. The comparison of actual schedules and the ones yielded by the proposed method indicates that proposed method can improve the appointment scheduling in realistic clinical settings.

Acronyms

AS	Appointment systems
ASP	Appointment scheduling problem
ASR	Appointment scheduling rules
BBD	Box-Behnken design
BPETS	Binary linear programming enhanced tabu search
CRC32	32-bit cyclic redundancy check
CV	Coefficient of variance
DSM	Deterministic scheduling module
DSR	Dome-shape rule
ETS	Enhanced tabu search
FCFS	First-come first-served
GDP	Gross domestic product
HV	Hypervolume indicator
IAC	Integer appointment constructor
IP	Integer programming

IPETS	Integer programming enhanced tabu search
LP	Linear programming
LPT	Longest processing time
MATS	Multi-agent tabu search
MIP	Mixed integer programming
MOTS	Multiobjective tabu search
MP	Mathematical programming
MPETS	Mathematical programming based enhanced tabu search
MSS	Master surgery schedule
NSGA-II	Non-dominated sorting genetic algorithm
OASP	Outpatient appointment scheduling problem
OR	Operating room
PACU	Post anaesthesia unit
R&S	Ranking and selection
SCV	Increasing coefficient of variance of service time
SDSR	Stochastic DSR
SE	Square root of error

SICU	Surgical intensive care unit
SPT	Shortest processing time
SSPT	Stochastic SPT
STS	Simulation-based tabu search
SVR	Increasing variance of service time

Acknowledgement

I want to extend my gratitude to my advisers Dr. T. ElMekkawy and Dr. G. Wang. This thesis would not have been achieved without their precious support and guidance. I wish to offer my sincerest thanks to Dr. P. Jula for his endless support. I also want to thank Dr. Q. Peng and Dr. U. Annakkage, my committee members, for their constructive and valuable suggestions throughout the work. I wish to thank Mr. E. Nobakht and Mr. J. Gurling from Fraser Health Authority for sharing their priceless insight in healthcare with me. Finally, I would like to thank the University of Manitoba for the Graduate Fellowship.

Dedication

*To my beloved **parents** whose sacrifices granted me this life changing experience.*

Table of contents

List of figures

List of tables

CHAPTER ONE

1. Introduction

Health expenditures account for a remarkable part of Gross Domestic Product (GDP) of many countries in the last decade. Centers for Medicare and Medicaid Services reported that more than 16% of US GDP has been assigned for healthcare and made it the largest industry in the United Sates [1]. On the other side of the border, the Canadian Institute for Health Information reported that health expenditure accounted for 10.5% of Canadian GDP in 2007 and expected its continuous increasing trend [2]. The continuous increase of healthcare expenditure in recent years raised concerns for government and healthcare management. Consequently, the health industry is under pressure to reduce costs while thriving to improve the quality of healthcare.

Operations Research has been used as a tool to improve efficiency of processes and to reduce the cost in the healthcare. An important instance of Operations Research application in healthcare is scheduling outpatient appointments. In last 50 years, many studies focused on looking for better appointment systems (AS). Researchers and practitioners sought to find appointment scheduling rules (ASR) which reduces provider idle time, overtime, and patient waiting time. Along with these research studies, others targeted different performance measures such as the number of patients in the clinic, or the variance of patient waiting time over a clinic session.

In the last decade, there has been an increase in the number of outpatient cases and a corresponding decline in inpatient hospitalization. Availability of effective drug treatments and faster diagnosis (e.g., MRI and CT scans) has attributed to this increase by reducing the patients'

length of stay in healthcare services. Accordingly, many studies have been carried out to improve the efficiency of procedures in order to address the increase in demand for outpatient services. Outpatient services are performed in outpatient clinics, which refer to facilities in which patients leave the facility on the same day, after receiving the service.

Appointment scheduling aims to seek a trade-off between the needs of both patients and healthcare providers. In particular, healthcare facilities desire to maximize the utilization of their resources while minimizing the overtime incurred to serve all scheduled patients. On the other hand, patients expect to receive the service with minimum waiting time.

A literature review shows a room for advancement in the literature on methods that address the challenges of appointment scheduling in multistage clinics where multiple types of patients are served with stochastic service times. Operations at such a clinic may be impacted by additional environmental factors and could involve multiple constraints such as multiple servers at each stage, resource availability, human and equipment resource compatibility, and heterogeneous service sequence for patients (i.e., for each patient the stages and the order of visiting them may vary based on the patient type).

This thesis proposes a simulation-based scheduling approach that tackles such challenges by incorporating three methods of (a) mathematical programming, (b) discrete event simulation, and (c) metaheuristics. The advantage of this proposed approach is in its flexibility at incorporating complexity of the system to the required level of detail, by application of simulation as well as employing metaheuristic methods (e.g., tabu search) to seek optimum schedules. In addition, the mathematical programming reinforces the algorithm by delivering promising initial solutions obtained from solving the deterministic version of the problem.

1.1. Research objectives

In the literature of appointment scheduling, several performance measures have been considered individually to evaluate appointment schedules. However, for the most part, the emphasis has been given to maximizing the utilization of the clinic's resources. The opportunities for advancement of literature which recognized by reviewing previous studies can be attributed to the challenges of addressing multiple criteria optimization on one hand and the existing gap between the real clinics' environment and the corresponding simplified models used in the literature.

Capturing the complexities of appointment scheduling using analytical and optimization methods is challenging. These methods usually make strong assumptions or simply focus on a few elements of the system in order to tackle it.

A number of influencing factors make the appointment scheduling challenging. The first factor is the presence of service time uncertainty at each stage of care delivery. Several elements such as requirements of the patient and the qualifications of the providers often contribute to the uncertainty of the service duration. For instance, in the surgery appointment scheduling that will be addressed in Chapter four, several factors including experience level of the surgical team and patient characteristics such as age, weight, and other unpredicted conditions affect the surgery durations.

In addition to the uncertainty of the service duration, considering the clinic as a multistage system increases the complexity of the problem. When the scheduling of multistage clinics is considered, the service duration of each patient type at each stage can be different.

Handling the resource constraints in appointment scheduling of multistage clinics is the next factor which influences the complexity of the problem. Two types of resource constraints often exist in appointment scheduling. Constraints on the availability of resources are the most commonly addressed issue concerning resources in the appointment scheduling. A scheduler needs to make sure that the required amount/number of medicine, equipment, and personnel are available to serve the scheduled patient while considering the uncertainty of the service durations at each stage. The constraints of this type may simply be addressed by the amount/number of required resources (e.g., the number of available beds), or further expanded to handle time-window constrains on the availability of resources (e.g., time window of surgeon's availability).

The second challenge regarding the resource management in this context deals with the compatibility of resources with the patient type. In the OR department, for instance, patients with a trauma condition are required to be operated on by a surgeon with orthopedic specialty. These requirements further complicate the resource management in appointment scheduling.

Most studies in the literature have focused on the clinics in which patients follow the same route and have the same sequence of services. However, in reality there are clinics that offer a variety of services ranging from doctor consultation to diagnostic services or minor surgeries. In this light, appointment scheduling is more complicated due to the fact that patients do not often follow the same path in these clinics, revisit the same stage through their journey, or simply go through the stages in different orders.

This thesis proposes several models and methods to better deliver appointment schedules in clinics and OR departments with the same amount/number of resources while addressing the above challenges. Overall, the objective of this work can be summarized as follows:

- To develop an efficient and effective appointment scheduling approach in multistage, multi-server outpatient and surgery facilities.
- To develop a method for solving more realistic problems, which is capable of addressing the following challenges:
    - Serving multiple patient types with stochastic service duration at each stage
    - Existence of non-identical and independent service time distributions for each type of patient at each stage; i.e., the scheduling method should not be limited in terms of the type of distribution that it supports.
    - Addressing patients with heterogeneous service sequences reflecting each patient type
    - Optimizing multiple performance criteria and delivering a Pareto optimal front
- To develop methods to provide a promising initial solution for appointment scheduling optimization

Analyzing the performance of the proposed methods as compared with the simple scheduling rules provides insights for practitioners in appointment scheduling. The scope of this work includes the appointment scheduling of a deterministic number of patients based on the time block appointment system. The uncertainty in arrivals of patients including patients with no-shows or those who have tardiness in arrivals has not been included in this work.

1.2. Overview of the proposed methodology

The proposed approach includes simulation and tabu search algorithm that represents the optimization component. In addition, the proposed approach includes a mathematical model which addresses the deterministic version of the patient appointment scheduling. Mathematical

programming considers the same assumption used by the stochastic version of the problem including: a specified number of patients, multiple stages, block time, and so on. However, the mathematical model does not account for stochastic duration of the service time. That is, contrary to the simulation model, the service time for each patient is deterministic and is predetermined. In simulation model, each stage follows a service time distribution that determines the service time when a patient enters the stage for service. The mean value of the service time distribution is used in the mathematical model.

Considering the relationship among the components of the proposed method, mathematical programming is the first step of the algorithm. A deterministic version of the problem is solved by mathematical programming. The obtained optimum solution is then sent to the tabu search as its starting point. Thus, the (near) optimal solution of the original stochastic problem is expected to be close to the solution obtained from the mathematical model.

In the next step, Tabu search uses the solution from the mathematical model to start the search process. Tabu search generates a neighbourhood of solutions using the provided result by mathematical programming. The neighbourhood is constructed based on the definition of local searches that are provided in the Tabu search algorithm. Tabu search selects a candidate solution from the generated neighbourhood with a number of techniques. This solution is sent to the simulation component to be evaluated. The simulation model applies the candidate solution (an appointment schedule) and determines its performance measure (e.g., average patient waiting time) by running several replications.

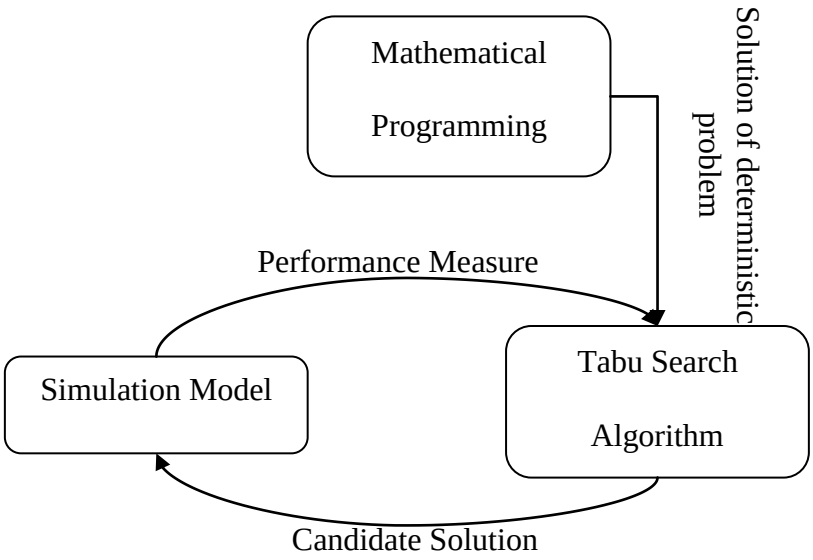

Figure 1.1 Relationship among different components of the proposed approach.

After evaluating the solution provided by the simulation component, the result is reported to the optimization component. Tabu search continues improving the solution through information that is obtained from the simulation. A different solution, which is expected to result from the search, is presented in the next iteration. These steps are repeated iteratively until the termination condition is met. Figure 1.1 represents the relationship of the three components of the proposed method (mathematical programming, simulation model, and Tabu search).

1.3. Thesis organization

The organization of this thesis is as follows. Chapter two presents a literature review for the appointment scheduling problem in outpatient clinics and operating room departments. It also gives an introduction to simulation-based optimization methodology.

Chapter three addresses the challenges of the appointment scheduling in a multistage clinic serving multiple types of patients with stochastic service time. Chapter four is dedicated to the appointment scheduling of patients in the OR departments. It includes our proposed method to schedule multiple types of patients with stochastic service time under resource availability and compatibility constraints. In addition, it investigates the application of the proposed algorithm in a case study of an OR department of a major Canadian hospital. Furthermore, performance of the simple scheduling rules is studied in comparison with the proposed methods. The comparison yields insights for practitioners to schedule patients under different sets of constraints.

Chapter five focuses on the appointment scheduling of patient appointments in clinics that serve multiple patient types with different service sequences. In this chapter, the proposed method presents a multiobjective approach which offers a Pareto (near) optimal front of schedules for the patients considering patient waiting time and overtime. The Pareto front would assist the clinic management to adopt the best scheduling scheme.

Chapter six applies the multiobjective scheduling method which is presented in Chapter five to a case study OR department of a major Canadian hospital. The performance of applying the proposed method is then evaluated using actual schedules that retrieved from actual data.

Finally, Chapter seven summarizes the conclusions of previous chapters and discusses the future research opportunities.

CHAPTER TWO

2. Literature survey

In the first section of this chapter the extensive literature of appointment scheduling in healthcare is reviewed. This chapter specifically focuses on the recent articles in appointment scheduling of outpatient and surgical departments that consider new and important challenges. The chapter begins by discussing the traditional appointment scheduling rule and performance measures used in the literature. Later, classification of the literature based on the employed methods in these articles is presented. Finally, the outstanding challenges and opportunities for further research directions are discussed.

The second section includes a review of simulation-based optimization approach. This is the approach adopted mainly to solve the appointment scheduling problems in this thesis. The basic concepts of the simulation-based optimization are described, and the existing methods within the context of simulation-based optimization are discussed. Section three discusses the contribution of this thesis to the existing literature.

2.1. Outpatient and surgical patient appointment scheduling

Outpatient clinics refer to facilities in which patients do not stay overnight in the system after receiving the service. They are often discharged on the same day of admission. Outpatient medical units can be described with characteristics such as random consultation times, uncertain patient arrivals, and uncertain provider arrival times. The performance of the appointment

scheduling for outpatient clinics is mainly evaluated through patients waiting time, providers' idle time, and clinic's overtime.

The appointment scheduling problem (ASP) has attracted many researchers since the 1950's. Different methods have been developed to provide an optimal appointment system by considering the distribution of service times, patients' punctuality, and provider arrival lateness. Additionally, the effect of environmental factors such as patients' arrival pattern, providers' arrival pattern, and missed appointments has been covered in the literature.

Two definitions are available for the patient waiting time in the literature. Patients' waiting time refers to the time between the latest of either arrival time or the appointment time of patient , and the actual start time of service [3], [4]. Others simply define it as the time between patient arrival and start of the service [5]. In this work, waiting time refers to the total time that a patient spends in queues within the clinic; i.e., if a patient arrives earlier than their appointment time, the tardiness will not be included in waiting time calculations.

2.1.1. Traditional appointment scheduling rules

Among the available measures in the literature to evaluate the performance of outpatient clinic, patient waiting time and provider idle time are the most commonly used criteria. Many works attempted to find appointment scheduling rules (ASR) which provide a trade-off between the waiting time of patients, idle time of health providers, and the overtime of the clinics. The reason for such studies lies in the fact that increasing one of these factors typically leads to a reduction in other criteria [5].

The available ASRs in the literature can be categorized into three main classes: single-block ASR, individual ASR, and multiple-block ASRs. Single-block ASR refers to a rule which

allocates all patients to the initial block of the session. This rule was used in clinics in the 1950s. However, due to the excessive amount of patient waiting time caused by this rule, studies recommended a shift from single block to individual ASRs. Individual ASRs assign each patient a specific appointment time [6–8]. Later, multiple-block ASRs have been recommended for outpatient clinics with short consultation time and unpunctual patient arrivals. The reason for this development is that the individual ASRs assign a single patient to each time block. In case the patient does not show up, the provider will be idle. Whereas multiple-block ASRs allocate multiple patients to a single time block of the session which leads to less provider's idleness.

Three different types of individual ASRs are available: fixed-interval individual ASR, fixed-interval ASR with an initial block, and variable-interval individual ASR [5]. Fixed-interval individual ASR sets the appointments with equal intervals [9–11]. In the fixed-interval ASR with an initial block, two or three patients are assigned to the beginning of the session in order to minimize the risk that the provider stays idle in case of no shows or tardy patients. This ASR schedules two or three patients at the first block while the rest are assigned at equal intervals [8], [10], [12–16]. On the other hand, variable interval individual ASR assigns each patient one of the varying appointment slots [16], [17].

The multiple-block ASRs can be divided into two classes depending on the number of patients scheduled for each time block n_i. The ASRs with fixed n_i are referred to as fixed-interval multiple-intervals [13], [18]. In contrast, the ASRs with varying intervals are called variable-interval multiple-block [3], [19–23].

Qu [5] stated that several analytical studies suggest a dome shape pattern for appointment intervals when patient arrival is punctual and there is a linear relationship between the patient

waiting time and provider idle time. That is, the length of intervals increases as it approaches the middle of the session and then decreases toward the end of session [24–28].

2.1.2. Performance measures

As mentioned earlier, the most common measures of solving the outpatient appointment scheduling in the literature are patient waiting time, the provider's idle time, and working overtime. One of the reasons that many studies considered these measures is their measurability and tangibility. There are, however, variations in the way that the measures are employed. Some articles considered the waiting time as a time measure while others derived a cost function using the time measures including waiting time, idle time, and overtime [29].

The literature also includes works that consider resource levelling objectives which target reducing the peak number of patients in a given time interval to level the system load. Levelling of the resources leads to less operational costs and eases the resource planning exercises. Marcon and Dexter [30] proposed a sequencing rule which satisfies the levelling objective. They [31] used discrete event simulation to study how classic sequencing rules can help levelling different stages of a surgery department.

In the patient scheduling context, cost-based measures usually address patient waiting time as well as doctors' idle time. The common cost function assumes a linear relationship between waiting cost and waiting time of patients. Some surveys suggest 30 minutes as the maximum acceptable waiting time for patients [32], [33].

In the context of operating room and surgery scheduling, the major performance measures include waiting time of patients and surgeons throughput, utilization of operating rooms, ward, and intensive care units, and overtime/overutilization of operating room, recovery and intensive

care units. In addition, workload levelling of operating room, recovery unit and intensive care units along with refusal of patient or cancellations are among the widely used performance measures studied in the literature. Cardoen et al. [34] considered the waiting time as the most widely used performance measure in literature due to the large number of complaints about the long waiting lists for surgeries. As well, the waiting time in the surgery steps is of significant importance as it leads to blocking of bottleneck resources such as ORs. Denton [35] developed a stochastic programming model to study how patient sequencing affects the waiting time of patients and idle time of ORs.

Cardeon et al. [34] reported the throughput as the second performance measure used in the literature of OR scheduling. They indicated that the close relationship between throughput and waiting time can be simply described by Little's law. That is, the average number of patients in the system equals the average length of stay of patients in the system (including waiting time and processing time) multiplied by average rate of arrivals to the system [36]. For instance, VanBerkel and Blake [37] studied the effect of varying throughput on the waiting time of patients using a simulation model. More specifically, they studied the throughput by running different scenarios on the number of available beds and ORs that are located in multiple facilities.

The third widely used criterion is overutilization (overtime) and underutilization. For instance, Lamiri et al. [38] proposed a stochastic model to schedule elective surgeries and minimize the total overtime cost in addition to the waiting time and penalties for violations of patients preferences for the time of the surgery. They solved the model using the branch and bound method and then compared their method with a simulation-based optimization approach. They

concluded that simulation-based optimization approach may offer superior results even in small-size problems.

Workload levelling of resources is the fourth measure commonly used in the literature of surgery scheduling. This measure targets developing operating room schedules which result in a smooth resource utilizations without peaks. In addition, levelling of bed occupancies and holding area, post anaesthesia care unit (PACU) utilization have been considered in the literature. For instance, Macron and Dexter [31] used a simulation model to examine the application of sequencing rules such as shortest processing time (SPT) and longest processing time (LPT) in order to minimize the maximum number of patients in the PACU and the holding area.

Makespan (in this thesis is called completion time) is the fifth measure that is widely used in the literature. Generally, it can be defined as the time between first patient's arrival time to the last patient's discharge time. Marcon and Dexter [30] studied the completion time of a OR department including the PACU. Cardoen et al. [34] mentioned that most studies within this group only considered the completion time of OR and they further suggested that completion time should be studied in combination with other measures such as waiting time since reducing the makespan often leads to a dense schedule.

The final measure which is considered in the literature of surgery scheduling is the number of patient deferrals or cancellations. For instance, in an attempt to reduce waiting time of surgery patients, Kim and Horowitz [39] reduced the number of cancelled elective cases resulting from lack of intensive care beds without deteriorating admission of other patient streams into intensive care units (ICU).

2.1.3. Methods used for optimization of appointment scheduling

Different approaches have been employed to evaluate and optimize appointment scheduling in outpatient facilities. Literature of outpatient scheduling can be divided into four main classes: queuing theory, mathematical programming, simulation, and combination of simulation with optimization techniques.

2.1.3.1. Queuing theory

Many works considered appointment scheduling problem as a queuing problem. They considered different queuing models to represent different appointment systems and environmental factors. Qu [5] reported that the models range from the simple D/M/1 queuing model Jansson [14] to the M(t)/G/m/K queuing model Brahimi and Worthington [40]. Readers can refer to queuing theory textbooks (e.g., [41]) for a detailed description of queuing models. Readers can refer to work by Qu [5] and Cayirli [29] for comprehensive reviews of literature relevant to application of queuing theory in outpatient appointment scheduling.

Lindley [7] considered servers with Erlang service times and made a comparison between two appointment systems (AS). The first AS schedules the patients at regular intervals while the other permits random patient arrivals. The author used the special characteristics of Erlang distribution and proposed an AS that could significantly decrease patient waiting time.

Jansson [14] studied a queuing system with service time following exponential distribution and constant inter-arrival times. The author derived transient and steady-state distribution of doctor idle time and patient waiting time while representing the clinic as a queuing system with one server.

Mercer's work [42], [43] can be distinguished from the others. The author assumes unpunctual patient arrivals. Mercer [42] developed a distribution for queue length in presence of patients with late arrivals or no-shows. In this model they considered processing patients in single or multiple stages. They broadened their model in [43], by including distribution of queue length and addressing more general conditions such as batch-arrivals.

Vanden Bosch and Dietz [4] proposed an easily applicable scheduling method for outpatient clinics in order to minimize a weighted sum of patients' waiting time and clinics' overtime. They developed a queuing model in which arrivals are deterministic while no-shows are possible. They assumed different types of patients based on their service time and represented the service time using phase-type durations, which stem from a combination of one or more inter-related Poisson processes. They provided an optimal method to sequence the patients based on the specific structure of their model.

Pegden and Rosenshine [17] addressed a S(n)/M/1 model with a single server in order to minimize the cost of customers waiting and server's availability. The model considered exponential service time and a specified number of scheduled customers in outpatient clinics. They derived the optimal appointment interval for maximum of three consecutive patients that minimizes the expected total cost of system, including costs of patient waiting time and provider's idle time.

Brahimi and Worthington [40] considered the M(t)/G/m/K model with multiple servers with general service time. Their model schedules patients with different arrival rate in a clinic with limited waiting space using a Markovian chain with heterogeneous interarrival times and discrete service time. The authors developed an algorithm to approximate continuous service time

distribution with a discrete service time. They compared current and proposed appointment scheduling systems for a M(t)/G/m/25 model. In addition, in Brahimi and Worthington [15], they demonstrated that for a given probability of servers being busy, their AS decreases the number of patients in the clinic.

Zeng et al. [44] studied the application of overbooking in the patient scheduling in order to remedy the effects of no-shows. They presented a model in which patients were grouped with different no-shows probabilities. They reported that grouping patients using their no-show probabilities lead to better schedules.

While queuing models deliver important insights with regard to the impacts of uncertainty in the appointment scheduling systems, they suffer from shortcomings in the outpatient and surgical scheduling. Many papers in this category assumed the system in steady state and subsequently used the properties of such systems in modeling and scheduling optimization. However, considering the limited number of served patients in specific clinic operating hours, the steady state cannot be reached. In addition, Erdogan [45] mentioned that the queuing systems tend to make restricting assumptions concerning the distribution of service time. For instance, several models assumed exponential or discrete distributions for service time where practical studies suggest distributions such as lognormal distribution [46], [47].

2.1.3.2. Mathematical programming (MP)

The second analytical method that has been used in outpatient scheduling is mathematical programming. Qu [5] suggested that most research in this category apply generic scheduling methods that can be employed for outpatient scheduling problem. There are, however, some works that emphasize the outpatient clinics and medical services scheduling.

Fries and Marathe [48] addressed both the static and dynamic scheduling problems when optimizing variable-block multiple-block ASR. They used dynamic programming to solve the scheduling problem. The size of the time blocks is determined by solving a dynamic scheduling problem at the end of previous time block. The static scheduling problem, however, calculates the size of time block prior to the start of clinic session.

Liao et al. [20] employed dynamic programming to schedule clinics with Erlang service time distribution. They used the solution of the dynamic scheduling problem as a lower bound for proposed branch and bound algorithm, which then solves a static scheduling problem.

Wang [25] optimized scheduling on the variable interval individual ASR using static and dynamic scheduling problems. They considered the exponential distribution and showed that for this problem the distribution is phase-type. Using the advantageous properties of phase-type distributions, a method was proposed for the static and dynamic problems which deliver the optimal appointment time for a finite number of patients. Wang considered the minimization of the expected total cost, which is calculated from patient flow time and providers' completion time. The author indicated for a finite number of identical patients, the optimal schedule presents a "dome" shape pattern for appointment intervals.

The appointment scheduling for other facilities like medical screening and surgical units are also included in the literature. For instance, Baker and Atherill [49] used stochastic programming to determine the block size of a variable-block multiple-block ASR for an ambulatory day-care unit. The authors considered 31 groups of patients and classified them considering their history of no-shows and their last five visits.

Denton and Gupta [24] presented a stochastic programming model in which appointment times were determined optimally for a fixed appointment sequence. They proposed upper bounds that were independent from distribution of service time for their algorithm and solved the model using an L-shaped algorithm with sequential bounding. Their results indicated that the expected total cost had a direct relationship with the standard deviation of service time and the number of patients scheduled for the session. Robinson and Chen [28] addressed the same problem by Monte Carlo sampling and proposed a heuristic method which exploited dome shape patterns of optimal schedules.

Hsu et al. [50] developed a deterministic two-stage no-wait flow shop model for an ambulatory surgery clinic. They proposed a heuristic to minimize the number of PACU nurses and the makespan of the schedule. Guinet and Chaabane [51] developed a no-wait flow shop method for inpatient surgery. Ozkarahan [52] proposed a deterministic mixed integer programming (MIP) model for assignment of the surgery cases to operating rooms (ORs) in order to minimize the under-time and overtime. In addition, the author studied rules for patients sequencing.

Sier et al. [53] suggested a mixed integer non-liner programming model to assign surgery time blocks to patients. They considered a penalty function including patient's age and resources such as scarce equipment and ORs. They developed a simulated annealing approach to solve their model. Pham and Klinkert [54] proposed a deterministic MIP model based on multi-blocking job shop scheduling problem with the goal of minimizing makespan in surgery-case scheduling. They also suggested methods to extend their model to address minimization of overtime. In the model, they allowed emergency cases by imposing deadline constraints and using job insertion methods.

Testi and Tanfani [55] developed a binary linear programming model to solve the master surgical schedule problem, together with the surgical case assignment problem. Their model has the goal of minimizing the overall patients' welfare loss calculated based on the waiting time of patients on the waiting list. They combined urgency levels for patients and studied the impact of "what-if" scenarios such as assigning additional ORs.

Min and Yih [56] proposed a stochastic programming model for case scheduling problem. They considered OR and surgical intensive care units that included several specialties. However, the model did not consider the intake procedure and other resources such as nurses, surgeons and equipment. They solved the model using the sample average approximation.

Lamiri et al. [57] developed a stochastic programming model for surgery planning to minimize elective patients' assignment costs and expected overtime costs. They considered elective and emergency cases and presented an "almost-exact" Monte Carlo simulation method. They studied the performance of their method compared with several heuristic and metaheuristic approaches (such as simulated annealing and tabu search). The authors reported better performance, compared to heuristic and metaheuristic methods, for small to medium sized problems. However, the resulting computation times were found to be significantly higher. For large problems, however, tabu search provided better solutions than those provided by the almost-exact method in a reasonable amount of time.

Batun et al. [58] proposed a two-stage stochastic mixed integer programming model in order to minimize the expected operating cost of OR department. The expected operating cost includes fixed operating cost of the OR department and the costs incurred from overtime and surgeons waiting time. They studied operating room pooling and parallel surgery processing and reported

that it led to significant cost reductions. Their model included decisions such as the number of ORs in work each day, allocation of surgeries to each OR, and the sequence of surgeries within each OR, and start time of each surgeon. Since their model is computationally intractable to be solved in real-sized problems, they employ some structural properties of their model to introduce a set of "applicable valid inequalities" which makes those problems solvable.

Overall, MP models have been used in several studies and delivered promising results and important insights to the appointment scheduling. However, most MP methods (except for stochastic programming) do not address stochastic nature of service times in outpatient clinic scheduling. Despite the fact that stochastic programming can accommodate stochastic service time, they often simplify the clinics and OR departments as a single stage facility. Moreover, stochastic programming models are usually analytically intractable, and suffer from long computational time for realistic scheduling problems.

2.1.3.3. Simulation

There have been many studies that benefited from flexibility of simulation models to address the challenges of patient scheduling in outpatient and surgical facilities. Many articles investigated the relationship between various environmental factors (such as no-shows, case cancelations, and patient priorities) and different performance measures. Moreover, simulation enables researchers to perform experiments to evaluate the performance of alternative scheduling rules.

Bailey [12], [59] studied initial block, appropriate block size, and clinic session size for the individual interval ASR by means of Monte-Carlo simulation. Assuming gamma service time distributions, Bailey indicated that the best rule for appointment scheduling for the defined

context is to assign two or three patients at the beginning of each day and then set the block size equal to the average consultation time.

Fetter and Thompson [9] studied several performance measures and examined impact of the multiple environmental factors on them. They expressed the significance of clinic load and providers' punctuality for optimal scheduling.

Ho and Lau [16], [60] , also, investigated the impact of various factors on the performance criteria of several ASRs using a simulation study. They considered multiple factors such as type distribution of service time, no-show rate, and the number of patients in the clinic. They reported that the performance of scheduling rules are sensitive to, in a decreasing order of importance, no-show rates, the coefficient of variations of consultation time, and the number of patients in one clinic session.

Vissers [61] addressed the impact of the punctuality of patients' arrivals on a cost function of patient waiting time and doctor idle time by means of a simulation model. The authors also developed a heuristic to determine optimal appointment schedule minimizing the expected patient waiting time and provider idle time.

Klassen and Rohleder [10] used simulation model to compare multiple scheduling rules in order to minimize patient waiting time and provider idle time. They especially focused on two important factors of the scheduling rule and position of appointment slots for potential urgent patients. Their experimental results suggested that the best scheduling rule and the position of emergency slots is dependent on the mean and variance of service time.

White and Pike [13] studied the performance of multiple ASRs. They used a range of different means and variance of service times while they studied punctual patient arrivals, no-show rates,

and provider lateness. They reported the importance of punctuality of doctors and the number of patients in a session.

Considering surgery scheduling, Schmitz and Kwak [62] studied the determination of required number of ORs in a multi-OR surgical unit with recovery. Dexter et al. [63] used simulation to address general surgery scheduling. They proposed a method to assign the time block to the surgeons and schedule patients to improve utilization of ORs. Marcon and Dexter [30] used simulation to analyze the impact of different sequencing rules on OR utilization and workload of post anesthesia care unit (PACU). Lowery [64] determined the bed requirement of a hospital's critical care units such as OR, ICU and recovery using a simulation tool. Lowery and Davis [65] used a simulation model to examine effects of surgery schedule and variability in surgery durations on the number of required beds. Tyler et al. [66] developed a simulation model for an OR to improve the OR utilization. They also examined the impact of other factors such as average patient waiting time and variability of surgery duration on OR utilization.

An advantage of simulation over other methods, particularly analytical approaches, lies in the ability of modeling complex system such as patient priority, multistage facilities, and servers with different service time distribution. In addition, simulation does not require strong assumptions such as steady-state behaviour that is not reached in most of cases in healthcare. Despite the advantages that simulation modeling offers, there are a few drawbacks in application of simulation models in appointment scheduling. Simulation models often require intensive computation and developing results may be time consuming compared to the analytical methods. Furthermore, simulation models lack an optimization strategy and do not deliver optimal results.

2.1.3.4. Simulation-based optimization

In light of the pros and cons of solution methods discussed in previous sections, some researchers tried to develop heuristics or metaheuristics that may lead to the optimum in simulation-based studies. For instance, Liu and Liu [21] employed simulation modeling to analyze performance of the fixed-interval multiple-block ASR. They considered that the clinic included several providers and assumed provider lateness. Based on the result of experiments and the properties of the best obtained appointment system, they developed a heuristic that recommended the best ASR according to the factors that were involved in the clinic.

Some recent works considered the application of simulation-based optimization techniques. Huschka et al. [67] considered a discrete event simulation model for the analysis and design of an endoscopy suite to investigate different surgeon-to-OR allocation scenarios. Various performance measures such as suite overtime and patient waiting time were analyzed in the model. A simulated annealing heuristic was used to improve the scheduled start time of cases with respect to expected overtime and patient waiting time.

Klassen and Yoogalingam [68] considered a single stage outpatient clinic and used OptQuest to decide on the arrival time of patients. OptQuest is a simulation-based optimization package that accompanies many discrete event simulation tools. They studied dome patterns in appointment scheduling and suggested that practitioners can employ the "plateau-dome" type rule in many different environments.

Gul et al. [69] considered an outpatient surgical suite, and investigated the impact of several sequencing and scheduling heuristics on competing performance criteria. They developed a simulation model which incorporated a bi-criteria genetic algorithm. The authors demonstrated

the impact of different surgery schedules on the competing objectives of the mean patient waiting time and amount of overtime of the outpatient surgical suite. They indicated that the arrival time schedules substantially influenced the expected overtime and patient waiting time, while surgery allocation and sequencing heuristics had a smaller effect.

Jerbi and Kamoun [70] studied several ASRs in a nephrology clinic in Tunisia using a simulation model and goal programming. They evaluated the performance of each ASR using simulation model in terms of waiting time of patients and utilization of doctors. In the next step, they developed a goal programming model, based on the performance criteria obtained from simulation studies, to choose the most appropriate ASR for the clinic.

Overall, although analytical methods may obtain optimal solution of the problem, they suffer from limitations which encourage application of simulation. Most of the studies that have used queuing theory assume that clinics show steady-state behavior. However, this assumption cannot be achieved in most of clinics with a finite and small number of patients [7], [29].

On the other hand, the studies that used mathematical programming approaches are always vulnerable to the curse of dimensionality. That is, if they aim at modeling a realistic problem, the algorithm would demand for substantial computational efforts which make the algorithm ineffective for large problems. In addition, analytical approaches are not able to consider several environmental factors such as patient priority, multistage facilities, and servers with different service time distribution in the same time.

Simulation is a very powerful tool for evaluating performance of different appointment systems. It also can accommodate various conditions and constraints that other methods cannot provide. Thus, it can be considered an effective evaluation tool when the system is too complex to be

presented by mathematical terms. In spite of all flexibilities and capabilities which simulation modeling offers, this technique possesses its own drawbacks when is used individually to deliver the optimal schedule. In comparison with analytical methods, simulation does not include any optimization strategy. This is because simulation is developed to model mechanics of a system in different situations through a course of time. Simulation can deliver the outcome of practicing different appointment rules while it cannot choose which modifications can alter the course of changes that result in reaching an optimal policy. Hence, researchers have to design a number of experiments and manually lead the optimization progress by comparing results obtained from multiple simulation studies.

In order to remedy the shortcoming of solution methods discussed above, researchers tried to combine the simulation and optimization methods. Combination of simulation models with optimization procedures allows for more flexibility in modeling of complex systems while also enabling the search for optimal (or near optimal) solutions.

From the literature review, it is apparent that there is a need that has not been satisfied in the literature of appointment scheduling in surgery and outpatient facilities for efficient and effective methods that address multistage and multi-server facilities in presence of multiple patient types with stochastic service time. Furthermore, in such facilities, further challenging factors need to be addressed in scheduling, such as heterogeneous service sequence, availability of resources (physicians, surgeons, and PACU beds), compatibility of resources (i.e., which type of surgeon can serve a particular type of patients), and time window of availability for resources.

2.2. A brief description of simulation-based optimization methods

As the proposed approach in this thesis is based on simulation-based optimization modes, in this section the general techniques are described followed by an overview of the proposed methodology.

Historically, simulation and optimization were separate methods due to limited computational capabilities. As the computational power drastically improved, it became popular that major simulation software tools include an optimization module (e.g., OptQuest, which is included in Rockwell Arena™ and several other packages) [71].

The most common goal of discrete–event simulation studies is the determination of the best system setting and design among a set of proposed alternative settings. The simulation presents flexibility to model systems in detail such that various constraints and environmental factors are analyzed through simulation. On the other hand, optimization techniques are able to efficiently and effectively identify optimal settings for simulation to achieve defined objectives.

However, both these methods own shortcomings when applied to real world problems. Optimization techniques usually require explicit mathematical representation of system, which in many cases cannot be obtained due to the system complexity or lack of analytical methods. Simulation on the other hand, suffers from lack of a search strategy to find best settings for the modeled system. Considering the pros and cons of both approaches, many researchers made an effort to integrate these two, in hope that this combination can result in a synergy to reach better results. This combination resulted in the development of simulation-based optimization methods.

More specifically, methods that are available in the context of appointment scheduling have been discussed in the previous section and the advantages and shortcomings of each method have been

described. Based on those discussions, simulation, bearing many advantages over queuing methods, is encouraged to be combined with an optimization algorithm to solve real world appointment scheduling problems. Fu [71] describes the problem of simulation-based optimization as the determination of a configuration which minimizes the objective function:

$$\min_{\phi \in \Phi} F(\phi) = E[S(\phi, \omega)] \tag{2-1}$$

Where, ϕ accounts for (a vector of) input variables, $F(\phi)$ is the objective function and ω represents simulation replications, and S is the performance measure. Usually an estimator for $F(\phi)$ is used. For example, $E[S(\phi, \omega)]$ provides an unbiased estimator. The constraint set Φ can be explicitly defined or implied by the simulation model. Figure 2.1 shows the relationships between the simulation and the optimization procedure in a simulation-based optimization system. In the literature, inputs and outputs of simulation are called different names. Inputs are referred to as controllable parameters settings, values, configuration, variables, solutions, designs, or factors [71]. Outputs are called performance measures, criteria, or responses. Some of the outputs are used in the objective function while some of inputs are used to constrain the problem.

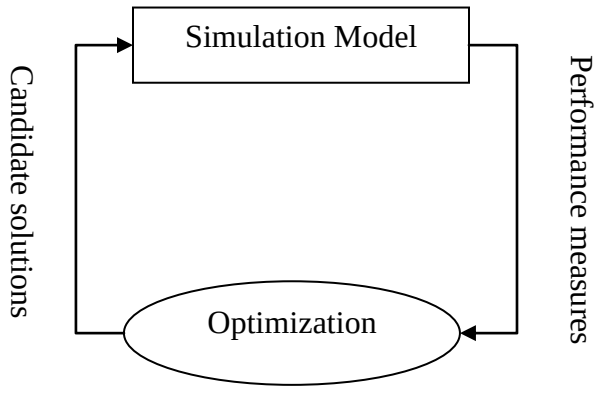

Figure 2.1 Relationship between simulation model and optimization.

Input variables can be divided into qualitative and quantitative categories. The former concerns the inputs that do not have natural ordering. Where, the latter can be further divided to discrete or continuous variables. The discrete input variables demand different approaches, similar to discrete optimization problems.

The actual process of simulation-based optimization can be described in two steps: (1) generating candidate solutions; and (2) evaluating solutions. The first step is done by the optimization algorithm in the same way as with deterministic approaches. The second step concerns the simulation model as an evaluating component. Contrary to conventional discrete optimization, in simulation-based optimization, function evaluation accounts for most of the computational efforts. Estimating $F(\phi)$ for different values of ϕ in search expends a huge amount of computation. A major factor for determination of computational cost of a simulation-based optimization algorithm is the number of replications to estimate $F(\phi)$.

The research literature of simulation-based optimization includes the following categories as solution approaches [71]:

- ranking and selection
- stochastic approximation
- response surface methodology
- random search methods

Among the available methods of simulation-based optimization, ranking and selection and random search can be applied to the current problem. The reason lies in the discrete nature of the

problem and the lack of derivatives for the objective functions and constraints. Next, a brief introduction to the aforementioned methods has been provided. In addition, some applications of each method will be reviewed in brevity.

2.2.1. Ranking and selection (R&S)

This method basically focuses on choosing the best configuration (set of parameters) over a set of finite choices. Swisher et al. [72] stated that this method is applicable to problems with a finite set of parameters which has a small number of members (i.e., two to 200). The goal of this method is to determine which of k input parameters can minimize $\mathrm{F}(\phi)$ using the two-step procedure previously described. In this method, the output of different configurations is obtained and can be used to minimize $\mathrm{F}(\phi)$. The difference between solutions is determined in order to recognize pairs whose difference is less than a user-specified value. By choosing the length of the simulation runs carefully, one can make sure that P^*, the probability of making the optimal selection, is close to one [72]. Gray and Goldsman [73] presented an application of indifference-zone R&S to select the best airspace configuration in European airports. Hsiao-Chang et al. [74], [75] provided a discussion about optimal computing budget allocation in discrete-event simulation by applying indifference-zone R&S. Morrice et al. [76] addressed an indifference-zone R&S for optimizing multiple performance measures.

2.2.2. Random search method

The random search methods have benefited from their generality and existence of theoretical convergence proof [71]. Practically, they have been applied to discrete optimization problems while there is sufficient theoretical ground for this to be applied to continuous optimization. In essence, the algorithms establish a neighbourhood structure and random search algorithms

pursue iteratively the neighbour points to find the best configuration. Algorithms of random search methods vary depending on two characteristics: first, according to their method to select the next point and second, through the way the optimal solution estimation is made (i.e., definitions of the estimate) [71].

To describe the random search algorithms, let $n(\phi)$ be the neighbourhood of $\phi \in \Phi$. The general scheme of algorithm would be as follows:

- Step 1: initializing the algorithm by determining initial values for variables randomly or using any constructive procedure.
- Step 2: selecting the next values for the variables (i.e., selecting the next candidate solution) from the neighbourhood.
- Step 3: evaluating the candidate solution using the simulation model, updating counters and variables including the best known solution.
- Step 4: performing step two and three until the stopping condition is met.

Andradóttir [77] provided a simple version of random search which requires the feasible set of configurations to be finite. Banks [78] mentioned that although this algorithm seemed simple, difficulties could occur when sampling randomly from the neighbourhood.

General search strategies like simulated annealing [79], [80], tabu search [81], and genetic algorithms [82] had been applied to stochastic problems relevant to discrete event simulation. Several studies used simulation-based optimization methods in different fields [83–85]. Brady and McGarvey [86] made a comparison among simulated annealing, tabu search, genetic algorithms, and a frequency-based heuristic to improve staffing levels in a simulation model of a pharmaceutical manufacturing laboratory. Glover et al. [87] used tabu search for several models.

As shown in Figure 2.1, simulation-based optimization method includes two components, i.e., simulation and optimization. The optimization module considers the simulation as a black box which presents the performance measures. On the other hand, the simulation looks at optimization as a component which only generates various settings and designs. In other words, the function of these two components is independent from each other and insight or knowledge of the simulation model is not shared by optimization. Moreover, optimization procedures usually start from a random solution while almost no knowledge about the problem is utilized to guide the search. Although some articles report merit of starting from a random point in the search space, it should be advantageous to use the available knowledge of the problem in the optimization procedure. This fundamentally changes the simulation-based optimization from a black-box optimization to a "gray-box" one.

More precisely, if any of the analytical methods that are applicable to the problem are included in the simulation-based optimization, it can incorporate knowledge of the problem into the optimization process. For instance, integrating a mathematical programming model can guide the optimization procedure to start from a more promising region of the search space. The proposed methodology has roots in such an observation. Through developing a mathematical programming enhanced tabu search method, this methodology aims at achieving better solutions compared to conventional simulation-based optimization in outpatient and surgical appointment scheduling.

2.3. Contributions to the literature

Despite the large number of articles in the literature of appointment scheduling, there is still abundant room for advancing appointment scheduling of multistage and multi-server facilities in the presence of uncertainty. Further opportunity for enhancement exists to address other challenging factors such as heterogeneous service sequence of patients, availability of resources,

patient-type compatibility of resources (i.e., which type of surgeon can serve a particular type of patients), and resource availability time windows. Furthermore, most work in the optimization of appointment scheduling in the presence of uncertainty tend to simplify the model to single stage facilities or to make strong assumptions due to challenges that arise in solving the model for realistic problems.

To address these challenges, multiple mathematical programming enhanced simulation-based optimization methods are developed. Chapter three proposes a novel approach for appointment scheduling of a multistage outpatient clinic with multiple resources and patient types (including patients returning from lab/X-ray), while considering general stochastic service time at each stage. This method integrates simulation, mathematical programming, and metaheuristics to generate results effectively and efficiently.

Chapter four addresses the challenges of appointment scheduling in a multi stage and multiserver operating room department serving multiple patient types with stochastic service times. The proposed model in Chapter four includes the resource compatibility and availability constraints where surgeons' availability is restricted by time window constraints. Three simulation-based optimization methods are proposed, which incorporate simulation and tabu search method along with binary and integer programming models to reach (near) optimal appointment schedules. Three objectives of completion time, average waiting time of patient, and the number of cancelations are considered to evaluate the quality of schedules. Furthermore, one of the proposed methods with superior performance has been applied to a case study OR department in a major Canadian hospital. Finally, a study of the performance of several simple scheduling rules in the appointment scheduling of OR department has been conducted, which provides several insights for practitioners to improve appointment scheduling in this context.

Chapter five proposes a multiobjective simulation-based optimization method to address the appointment scheduling of clinics that serve patients with different service sequences in order to minimize the waiting time and overtime. That is, each type of patient in this clinic has a particular pathway and undergoes a specific set of stages. Although several works in the literature considered multi criteria appointment scheduling, most studies used a weighted sum of objectives to generate a single function optimization problem which resulted in single appointment schedules rather than a Pareto front for appointment schedules. Our proposed method offers the Pareto (near) optimal set of schedules, which considers the trade-offs between the factors that influence patients and providers. The multiobjective scheduling method proposed in Chapter five is applied to a case study OR department and the performance of the method is compared with the performance of actual schedules for multiple days.

CHAPTER THREE

3. Appointment scheduling of an outpatient clinic

3.1. Introduction

Outpatient clinics refer to facilities in which patients leave the system on the same day after receiving the service. This chapter focuses on appointment scheduling of an outpatient clinic in which different types of patients are served using several resources such as receptionists, nurses, and doctors with stochastic service times. Considering the availability of the resources and the required service times, the goal of this chapter is to propose methods to determine the arrival time of the patients at the clinic in order to minimize the average waiting time of the patients within the clinic.

Researchers have suggested different methods to provide optimal schedules for the patients and the providers in outpatient appointment scheduling problem (OASP). These methods can be summarized in three main classes: analytical approaches (e.g. mathematical programming and queuing theory), simulation, and a combination of the simulation with heuristic methods. Analytical methods may offer optimal solutions while they mostly lack the flexibility required to accommodate several system parameters such as multistage units and constraints on resources required in the system. Simulation approaches, on the other hand, provide this flexibility and

may accommodate many conditions/restrictions, and can evaluate different scenarios. However, simulation approaches are very time-consuming and often lack the optimization capabilities.

In light of shortcoming and strength points of solution methods that were discussed in Chapter two, some studies tried to develop heuristics or metaheuristics that may assist the optimization process in simulation-based studies. For instance, Liu and Liu [21] employed simulation modeling to analyze performance of the fixed-interval multiple-block ASR. In addition, some articles considered the application of simulation-based optimization techniques. Klassen and Yoogalingam [68] considered a single stage outpatient clinic and used OptQuest to decide on the arrival time of patients. OptQuest is a simulation-based optimization package that accompanies many discrete event simulation tools. They studied dome patterns in appointment scheduling and suggested that practitioners can employ "plateau-dome" type rules in many different environments. Plateau-dome appointment rule indicates that the appointment allowance increases in the beginning of the session and decreases in the end of the session while it maintains equal appointment allowances in the remaining of the session.

However, there is still room for enhancement in existing literature for efficient and effective methods to address the challenges in the outpatient appointment scheduling problem. In this thesis, *efficiency* of a method refers to the amount of computation time required by the method to obtain solutions, while *effectiveness* addresses the quality of solutions generated by the method.

To address this need, two simulation-based optimization methods for appointment schedules are presented by integrating simulation and tabu search algorithm. The first method, termed enhanced tabu search (ETS), is composed of a tabu search algorithm and a simulation model of the clinic. In addition, it uses an auxiliary objective function to reduce the number of simulation

runs. The second method, mathematical programming based enhanced tabu search (MPETS), uses a mathematical programming model to improve the performance of the enhanced tabu search.

The proposed approaches, in this chapter, can be distinguished from the existing literature based on the following aspects. First, it addresses challenges in a multistage clinic with different resources and patient types, while considering general stochastic service time at each stage. Most existing articles focus on one aspect of this problem. Mathematical programming methods mainly do not consider the stochastic nature of the problem and when they consider the uncertainty in the model they are usually bound to address the clinic as a single stage due to computational challenges to solve the model.

Second, MPETS integrates simulation, mathematical programming, and metaheuristics to generate results effectively and efficiently. To the best of our knowledge, no previous work has integrated all these three methods to address OASP challenges. The relevant literature of OASP indicates that analytical methods may offer optimal solutions while they mostly lack the flexibility required to accommodate several system parameters such as priorities in the queues, dispatching rules, and so on. Simulation approaches, on the other hand, provide the flexibility and may accommodate many conditions and restrictions, and they can evaluate different scenarios. However, simulation approaches are very time-consuming and often lack the optimization capabilities. By integrating these techniques the method benefits from the power of optimization of analytical approaches, while having the flexibility of simulation.

Third, this chapter simultaneously addresses the challenges of considering both the clinic and diagnosis facility. Considering the patients returning from the lab/X-ray (diagnosis facility)

requires introduction of a loop in the patient flow, which may not be easily addressed by queuing theory models.

Finally, several techniques are presented to enhance the performance of the algorithms. For instance, an auxiliary objective function is introduced to tabu search to reduce the number of simulation runs, therefore reducing the computation time. Also, Box-Behnken approach is used to analyze the performance of proposed methods and to identify the factors that may affect the algorithms. The performance of the proposed method is then compared with the widely used method, OptQuest, and the factors impacting performance of each method have been studied.

This chapter has been organized as follows: Section two discusses the problem description. Section three describes the architecture of the proposed approaches and elaborates on different components of the algorithm. Section four reports out the design of experiments results and analyses. Finally, section five provides the conclusions.

3.2. Problem description

This chapter considers scheduling of a number of patients in a schedule horizon of a single day of an outpatient clinic. The problem concerns determining the arrival time of a pre-specified number of patients in order to minimize the average waiting time of served patients in a clinic. The problem considers different patient types and their corresponding stochastic service times. The availability of multiple resources such as receptionist, nurses, lab/X-ray technologist, and doctors are considered in the problem.

In this chapter, a hypothetical clinic is modeled to represent an outpatient clinic (see Figure 3.1). The clinic includes several stages including, reception, nurse visit, doctor consultation, and lab/X-ray. The last stage represents the X-ray department or lab unit for diagnosis and only a

portion of patients visit that stage. Each patient is received by reception after arrival. The patient is registered and his/her medical record will be retrieved. Then, the patient is seated in the waiting area to be called by a nurse. The nurse performs the first examining step and obtains any needed information related to the patient's condition. Next, each patient is directed to the assigned doctor. Using the information gathered by the nurse in the previous step and all historical record of the patient, the doctor provides recommendations to the patient. Based on the patient condition, the doctor determines if any test or X-ray would be necessary. Otherwise, the patient is discharged. The patients, who are instructed for an X-ray or a test, are received by the reception of lab or X-ray. Patients have to wait until being served by a lab or X-ray technician. After lab test or X-ray, patients return to the doctor for result review. Then, the patients are prescribed with necessary medications and treatments and discharged from the clinic. Figure 3.1 demonstrates different stages of the clinic.

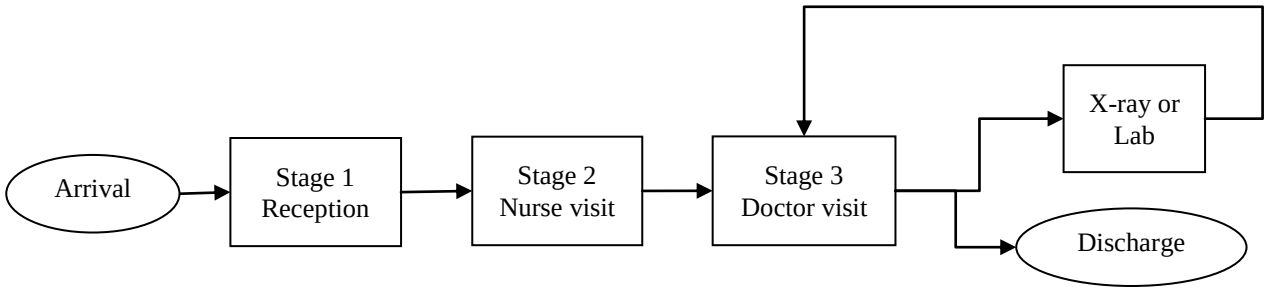

Figure 3.1 Different stages of a typical outpatient clinic.

3.3. Proposed methodology

Incorporation of simulation and optimization techniques provides the opportunity to apply simulation-based optimization approaches to the outpatient appointment scheduling. The proposed approaches, ETS and MPETS, include simulation and tabu search algorithm. In addition, MPETS includes a mathematical programming model which solves the deterministic

version of the outpatient appointment scheduling problem. Mathematical programming model has been included in the algorithm to propose a promising initial point to tabu search which is expected to improve the performance of the algorithm. It is assumed that the optimal solution for the deterministic version of the problem is sub-optimal or at least a good solution for the stochastic version of the problem.

Most studies from the available literature considered OASP as two separate problems, i.e., sequencing patients and then determining patient appointment times based on the proposed sequence. The proposed algorithms in this chapter consider both problems simultaneously and deliver the arrival schedule of patients that embodies both patient sequence and their appointment time.

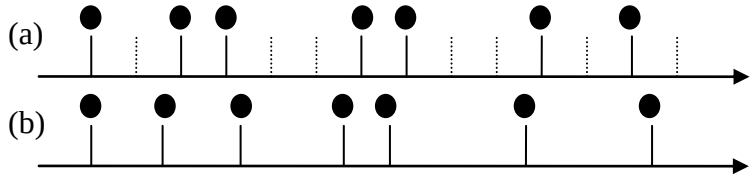

Figure 3.2 Comparison of AS; (a) current work and (b) studies that include different block sizes.

It is assumed that each session of the clinic includes a determined number of equal-length time blocks. Patient arrival would be allowed at the beginning of each time block and multiple patients' show-ups are permitted. Nonetheless, it is noteworthy that some researchers consider a variable block size and try to find an optimal block size for a determined sequence of patients which can result in variable block sizes. In this chapter, the variable block size can be achieved through combining several time block units. In other words, discrete timing of appointments is compared to the continuous one. Figure 3.2 illustrates these two different timing methods. The solid circles represent patient appointments and the lines define the time blocks. In Figure 3.2a,

the appointment schedule (AS) which is employed in this chapter is shown. It contains equal-length time blocks and the time blocks which include patient appointments are presented by solid lines. Figure 3.2b presents an AS with variable block size while showing the application of these two timing schemes on an identical sequence of patients. As the time block length shrinks, the discrete method becomes more similar to the continuous one. However, the algorithm may require more computational efforts.

MPETS is composed of simulation model, tabu search, mathematical programming model, and flowshop model. Considering the relationship among the four components, the mathematical programming model is the first step of the algorithm. In the next step, tabu search generates a neighbourhood of solutions using the result which is provided by mathematical programming. The neighbourhood is constructed based on the definition of local searches. The solutions in the neighbourhood are evaluated using an auxiliary objective function termed flowshop model. The flowshop model has been developed to enhance the performance of tabu search. Although the flowshop model neither accurately simulates the processes of the clinic nor can it evaluate the average waiting time of patients, it gives us a rough estimate of how good the solutions are. A number of solutions are sent to the simulation component to be evaluated. The simulation model uses the candidate solution (the appointment schedule) and evaluates its performance (i.e., average patient waiting time in this chapter) by running several replications. This process iterates until termination conditions (which are discussed in 3.4) are satisfied. Figure 3.3 shows the relationship between the components of the proposed method. Since MPETS is an improved version of ETS using mathematical programming and flowshop model, the description below will be based on MPETS.

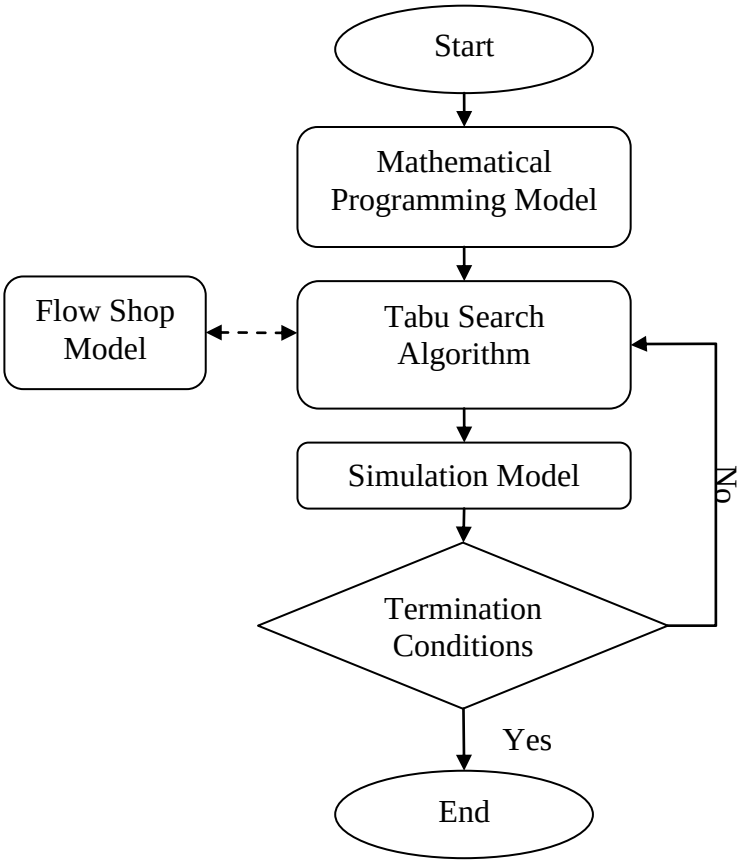

Figure 3.3 Relationship among different components of the proposed approach.

3.3.1. Simulation model

A simulation model of the outpatient clinic is developed which represents a multistage clinic with multi-server queuing system. Patient arrivals are assumed punctual and service providers do not show any lateness. These are considered standard assumptions and hold in many outpatient facilities such as surgical daycare clinics in the real world. In addition, in the context of outpatient clinics, it is believed that patient lateness is largely because of past experiences of poor scheduling and long waiting times. Robinson and Chen [28] argued whether establishing a better scheduling system would in turn, decrease patients' tardiness overtime.

The simulation model considers a day session of nine working hours and each time block is considered to be ten minutes. A specified numbers of patients are released at the beginning of each block. The patients will follow the steps mentioned in the problem description section. Different types of patients are considered in this study, i.e., patients are of different service times. For instance, patients might just visit for a cold while others may have complex cardiac problems which will impact their corresponding service times. The arrival pattern is determined according to an array variable which contains patient arrival schedule. This schedule can be set internally or can be changed externally to the simulation. At the first stage, patients are served in the order of their appointment, and then based on the first-come first-served (FCFS) rule in next stages.

The developed simulation model in this chapter is considered as a terminating simulation [88] where, a predetermined number of patients will be served in the specified time during which the clinic is open.

Since procedures within the simulation model include stochastic distributions, every run of the simulation model gives a sample from the distribution of the average waiting time of patients. Thus, in order to construct an estimate for the system performance (average patient waiting time) several replications of the simulation are required to be run for a single schedule. The average waiting time in the system is the focus of this study which is calculated from the data gathered through replications.

3.3.2. Arena and OptQuest

The model is developed in Rockwell Arena™ software tool. Arena™ has been selected since it is a comprehensive and widely-used simulation tool in healthcare and manufacturing. It also enables us to control simulation model from an external programming language and facilitates

passing data in real-time. The variables of simulation which account for resource control, distribution parameters, and patient arrival schedules can be transferred and changed by an external optimization component. In addition, Rockwell Arena™ includes an optimization package called OptQuest. OptQuest treats the simulation model as a *black box*; i.e., it considers only the input and output of the simulation model and assumes no knowledge of the model itself. In the clinic model, the input consists of the patients' arrival schedule as well as their type, which specifies the parameters of the service time distribution. The output in the current simulation model is the average waiting time of the patients. OptQuest incorporates the metaheuristics of tabu search, neural networks, and scatter search into a single search heuristic and is found reliable and efficient. For more information readers are referred to following articles[89–92]. Unfortunately, the exact algorithm of OptQuest is proprietary and unknown to us. As mentioned in previous section, OptQuest has been used to solve the outpatient appointment scheduling problem [68]. Here, OptQuest has been used as a benchmark to evaluate the performance of proposed methods. The quality of solutions and computation time were compared for the algorithms.

3.3.3. Mathematical programming (MP)

Here, a MP for patient appointment scheduling is presented. The model applies the same concept that is used in the integer programming model proposed by Jula and Leachman [93] for scheduling in semiconductor manufacturing. Each stage is a pair of a queue and a process. In our approach, the system is composed of a number of stages. Different stages are linked to each other in a sense that each stage receives the patients from the previous stage and delivers the patients to the next stage. When a patient enters a stage, it first stays in the queue. When the process has the capacity to serve, the patient is released and enters the process to be served. The patient will

spend the processing time and is then released to join the queue of the next process. Figure 3.4 illustrates the concept of a stage.

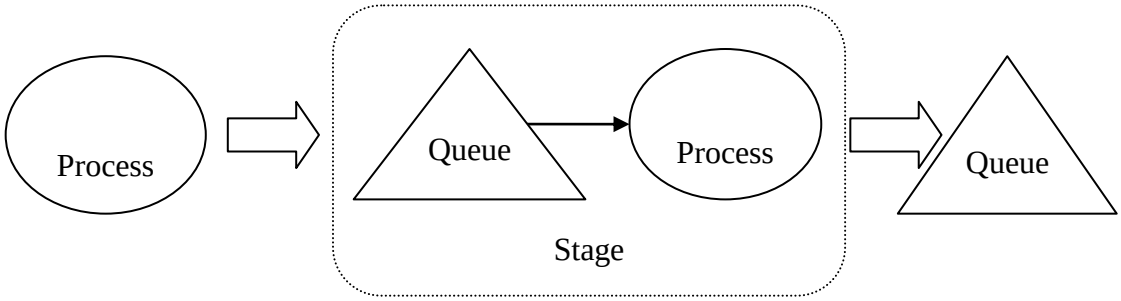

Figure 3.4 Components of a stage in mathematical programming model.

However, the mathematical programming model does not account for the stochastic duration of the service time. Contrary to the simulation model, the service time for each patient is deterministic and predetermined. In simulation model, each stage follows a service time distribution while in mathematical programming model, the mean of the service time distribution is used as the service time.

Another simplification in the mathematical programming model concerns the service time. In the mathematical programming model, it is assumed that the service time is an integer multiple of a time grid. This assumption simply demands the process component to seize and release the patients at the beginning/end of a time grid. It stipulates a discrete time-line to the system and is along the same line with the concept of time blocks in appointment scheduling. In addition, time blocks used in the simulation are assumed to be integer multiplier of time grid.

In general, MP model holds the following assumptions:

1. The processing time of each patient type at each stage is deterministic.

2. The processing time of each patient type at each stage is an integer multiple of the size of time grid.
3. The block size of simulation component is an integer multiple of time grid.
4. Each patient should go through all stages and cannot skip any stage.

Mathematical model (1):

The notation of the model is presented as follows:

Notation:

t discrete time index, t=1,...,T, where T is the time horizon and the total number of time grids in each day;

j stage index, j=1,...,M, where M is the number of stages in the clinic;

p patient type index, p=1,...,P, where P is the number of patient types which are served in the clinic.

Parameters:

α percentage of patients who need to do a test or an X-ray;

$S_{j,p}$ service time of patient type p in stage j;

v_p service time of patient type p at lab/X-ray stage;

R_j number of available servers or operators in stage j at the beginning of the scheduling horizon;

γ_p the penalty coefficient of waiting a single time grid for a patient.

Variables:

$I_{j,t,p}$ initial number of patients of type *p* in the line, waiting to be served at stage *j* (including the patients who have appointment in the first stage)

$x_{j,t,p}$ the number of patients of type *p* at stage *j* and time *t*; to start being processed at time *t*;

$Q_{j,t,p}$ the number of waiting patients to be served at stage *j* and time *t*;

$X_{j,t,p}$ the cumulative number of patients of type *p* at stage *j* and time *t*; to start being processed at time *t*;

$r_{j,t}$ the number of available idle resources at stage *j* and time *t;* each stage has its dedicated resources;

$d_{t,p}$ total number of patients type *p* discharged from the clinic by the end of time *t*;

$C_{j,t}$ waiting time of all patients at stage *j* and time *t*;

$b_{t,p}$ the number of patients type *p* who entered the doctor visit stage at time *t* after performing the lab test or X-ray;

$l_{t,p}$ the number of patient type *p* who entered the lab or X-ray stage at time *t*;

$B_{t,p}$ the cumulative number of patients of type *p* to start being processed at doctor visit stage by the end of time *t* after performing the lab test or X-ray;

$L_{t,p}$ the cumulative number of patients of type *p* to start being processed at lab or X-ray stage by the end of time *t*;

$O_{t,p}$ the number of patients type p who are in the queue at time t waiting to be served by the lab or X-ray section;

$P_{t,p}$ the number of patients type p at time t who are waiting in the queue to be served by doctors after performing the lab or X-ray;

Model can be expressed as follows:

- Objective function:

$$\text{Minimizing} \quad \sum_{j}\sum_{t} C_{j,t} \tag{3-1}$$

- Queue balance constraints:

$$Q_{j,t,p} = I_{j,t,p} - X_{j,t,p} + X_{j-1,t-S_{j-1,p},p} \qquad \forall j,t,p, \quad (3\text{-}2)$$

$$O_{t,p} = \alpha \,.\, X_{M,t-s_{M,p},p} - L_{t,p} \qquad \forall t,p, \quad (3\text{-}3)$$

$$P_{t,p} = L_{t-v_p,p} - B_{t,p} \qquad \forall t,p, \quad (3\text{-}4)$$

$$d_{t,p} = (1-\alpha)X_{M,t-s_{M,p},p} + B_{t-S_{M,p},p} \qquad \forall\, t,p. \quad (3\text{-}5)$$

- Cumulative variables:

$$X_{j,t,p} = \sum_{\tau=1}^{t} x_{j,\tau,p} \qquad \forall\, j,t,p, \quad (3\text{-}6)$$

$$L_{t,p} = \sum_{\tau=1}^{t} l_{\tau,p} \qquad \forall\, t,p, \quad (3\text{-}7)$$

$$B_{t,p} = \sum_{\tau=1}^{t} b_{\tau,p} \qquad \forall\, t,p. \quad (3\text{-}8)$$

- Capacity constraints:

$$r_{j,t} = R_j - \sum_{p} X_{j,t,p} + \sum_{p} X_{j,t-S_{j,p},p} \qquad \forall\, t, j \neq M, \quad (3\text{-}9)$$

$$r_{M,t} = R_m - \sum_p X_{M,t,p} + \sum_p X_{M,t-S_{j,p},p} + \sum_p B_{t-S_{M,p},p} - \sum_p B_{t,p} \qquad \forall\, t, p. \qquad (3\text{-}10)$$

- Waiting time of patients:

$$C_{j,t} = \sum_p \gamma_p * Q_{j,t,p} \qquad \forall j, \forall t. \qquad (3\text{-}11)$$

- Number of patients who have to be served:

$$\sum_p d_{T,p} = \sum_j \sum_p I_{j,1,p}. \qquad (3\text{-}12)$$

$x_{j,t,p}$, $I_{j,t,p}$, $X_{j,t,p}$, $r_{j,t}$, and $d_{t,p}$ are non-negative integers .

- Initial conditions:

  The values for $x_{j,t,p}$, $I_{j,t,p}$, $X_{j,t,p}$, $r_{j,t}$, $d_{t,p}$ should be pre-specified from previous scheduling horizon if applicable.

The objective of the model, Equation (3-1) is to minimize the waiting of patients in the queues of all stages. This objective is designed to be as close as possible to the simulation model average waiting time measure and considers the same performance criteria which are used by the tabu search. In order to develop the objective function, the famous *Little's law* [41], which addresses the relationship between the number of patients in the queue and the queue waiting time has been used:

$$L = \lambda W \qquad (3\text{-}13)$$

Where L is the expected number of patients in the system, W is the expected time spent by each patient in the system, and λ is the rate of patient arrival to the system. It is assumed that the rate of arrival of patient to the system, λ, is pre-specified. Since the determination of waiting time for each patient is hard to track, the number of patients in the queue is considered. Minimizing the

patient waiting time in the queues will lead to the minimization of the total patient waiting time in the system.

Queue balance constraints define the relationship among the stages and maintain the links between stages. Equations (3-2 to 3-5) state the patient flow among different stages in the model. For instance, Equation (3-2) ensures that the number of patients of different type in the queue of j^{th} stage is equal to the initial number of patients who are available in the j^{th} stage at beginning of session plus the number of patients that arrive at the queue from the previous stage at time *t* subtracted by the number of patients leaving the queue. Equation (3-5) defines that the total number of patients of type *p* discharged from the clinic at time *t* is equal to the (1-α) times the total number of patients left the doctor stage for the first time by time *t* plus the patients who have been visited by the doctor again after getting their lab or X-ray done.

Equations (3-6) determine the cumulative variables that are associated with the number of patients in different stages. Equation (3-7) specifies that the cumulative number of patients at lab or X-ray stage by time *t*. Equation (3-8) states that cumulative number of patients who have been served by the lab or X-ray and visited the doctor for the second time. Equations (3-9 and 3-10) specify the current idle resource in different stages. Equation (3-10) determines the number of doctors who are idle at time *t*. Equation (3-11) defines that the waiting time of patients in stage *j* at time *t* is equal to the total number of patients who are staying in the queue of stage *j* at time block *t* multiplies by the waiting cost of patient type *p* per time block. Equation (3-12) stipulates that the pre-specified number of patients should be served while the model seeks to minimize the total waiting time.

Overall, the MP model employs the idea of time block scheduling from the literature in order to present optimal schedules with minimum waiting time for patients. To develop a time block appointment schedule for patients, model uses a discrete time grid.

The patient flow is of major attention in the proposed MP model in order to manage a patient's journey in the clinic. Additionally, it creates a correspondence with the simulation model which presents the stochastic version of the problem. This feature further facilitates the use of the result of MP model as an initial solution to the proposed method. For instance, MP model is capable of modeling the journey of patients who need to return to doctor consultation stage after having an imaging or lab procedure using Equations (3-3) and (3-4) and assuming that a proportion of patients require such service. MP model also, considers the resource availability at each stage and its impact on the patients waiting time through Equations (3-9) and (3-10).

3.3.4. Tabu search

Over the last 20 years many researchers applied tabu search to several combinatorial optimization problems. Many of these optimization methods iteratively perform a number of steps till a termination condition is satisfied. In such algorithms, each step consists of generating a next solution *j* from the current solution *i* through some specified steps. A neighbourhood is then defined for each current solution, *N(i)*. The next solution is obtained by searching around *N(i)* using neighbourhood search methods. The move refers to change of the seed of the neighbourhood in consecutive iterations. In order to improve the performance of search, tabu search keeps record of local information by means of different types of memory structures. While many iterative search methods only keep the best found solution, tabu search records the best known solution as well as information of the last predefined number of previous moves.

This additional information could include parts of the solutions that have been changed as well as the iteration at which the change happened.

Tabu search allows non-improving moves. That is, even if the best solution that tabu search finds in the current neighbourhood is a non-improving solution compared to the best known solution so far, the non-improving solution will be used in next iteration as the current solution, provided that the solution is obtained by a non-tabu move. As soon as non-improving moves are allowed, the chance of cycling and revisiting a solution exists. To prevent such moves, tabu moves are defined. The tabu list consists of attributes of recent moves that are prohibited in the next specified number of iterations. Then a specified number of latest points visited and moves are recorded in the tabu list to prevent occurrence of any revisiting and cycling in the algorithm. Thus, current move will not be visited for a number of iterations. However, a tabu move is permitted if its evaluation delivers better results than the best solution found so far. This exemption is called aspiration in the context of tabu search literature. The aspiration rule addresses the conditions that if a solution satisfies, the algorithm will treat the tabu solution as a regular solution; not a tabu solution. In this work, including next chapters, aspiration rule defines that if a tabu move results in a solution that is better than the best known solution so far it will be exempted and will be considered as the next seed for generating the neighbourhood.

3.3.4.1. Initialization

Typically, metaheuristic optimization methods are initialized using either randomly selected solutions or result of a constructive heuristic. The quality of initial solution plays a significant role in the performance of algorithms especially when their performance is compared based on a specified number of function evaluations or iterations. In this work, the optimal result of mathematical programming is translated to tabu search and used to generate the first

neighbourhood. Using the optimal results of deterministic version of the problem provides MATS with the promising initial solutions.

3.3.4.2. Solution presentation

Each solution consists of two parts. The first part is an array of size *n,* where *n* is the number of the patients. This array shows the time block to which each patient belongs. The second part is of size *T,* where *T* is the number of time blocks. The second part of a solution describes the number of patients who will be served in each time block. While the second part can be independently constructed from the first part, generating the first part from the second part requires additional information. This information includes the patients who are assigned to each time block and will be recorded for future conversions that are required along with each solution. For instance, if the clinic has 6 time blocks and serves 10 patients, first part of the solution may include {1, 2, 2, 4, 6, 6, 3, 3, 3, 1} and the second part presents {2, 2, 3, 1, 0, 2}. Any modification in each part is reflected in another part through the iteration of the algorithm.

3.3.4.3. Neighbourhood structure

Swapping: let *i* and *j* be two positions in the sequence *s* which are selected randomly. By performing swap-move iteratively, a neighbourhood of *s* is obtained by interchanging the patients in positions *i* and *j*. In the proposed method, swapping is applied to the second part of a solution structure. Basically, swapping neighbourhood structure accounts for interchanging of patients between two time blocks.

Insertion: let *i* and *j* be two positions in the sequence *s*, which are selected randomly. A neighbourhood of *s* is obtained by inserting the patients assigned to position *i* to position *j*, pushing the cells between these positions backward (forward), including the patients of position

j, if *j* is greater (less) than *i*. This change of positions is performed on the first part of a solution and the second part will be updated accordingly.

3.3.4.4. Tabu list

To construct the tabu lists, two different moves (swap and insertion moves) are considered. Based on each move definition, two separate tabu lists have been developed with different sizes, swap tabu list and insertion tabu list. These lists contain the history of recent moves. For example, when swapping scheme is applied, a list of moves is recorded. This list prevents any reverse moves (i.e. if positions 3 and 7 are swapped, the next moves cannot include the swap of 7 and 3). Each item on the lists remains effective for 30 iterations of the algorithm.

3.3.4.5. Stopping condition

In the tabu search algorithm different conditions can be considered as the stopping criteria such as a limit on the number of iterations, the number of function evaluations, or the computation time. In this chapter it is assumed that the algorithm will be terminated after a specified number of function evaluations in order to set a basis for comparison between different methods.

Figure 3.5 shows the flowchart of the MPETS algorithm. It starts by receiving the initial solution from the mathematical programming. Two neighbourhoods are built based on the initial solution using swap and insertion local searches. A random number of solutions in each neighbourhood are evaluated by using the flowshop model which is to be described in section 3.3.5. The solutions are sorted and ranked based on the value which has been assigned to them through the evaluation. A number of solutions from the current set of solutions are selected according to their rank and evaluated using simulation model. The algorithm determines whether the best solution is tabu or not. If the best solution is not tabu or is a tabu solution that satisfies the aspiration rule,

the best solution in the neighbourhoods is selected as the seed. Based on the new seed, the neighbourhood will be constructed and the tabu list is updated. On the other hand, if the best solution does not satisfy the aspiration conditions while it is tabu, the seed will remain the same. The algorithm iterates till the stopping condition is satisfied.

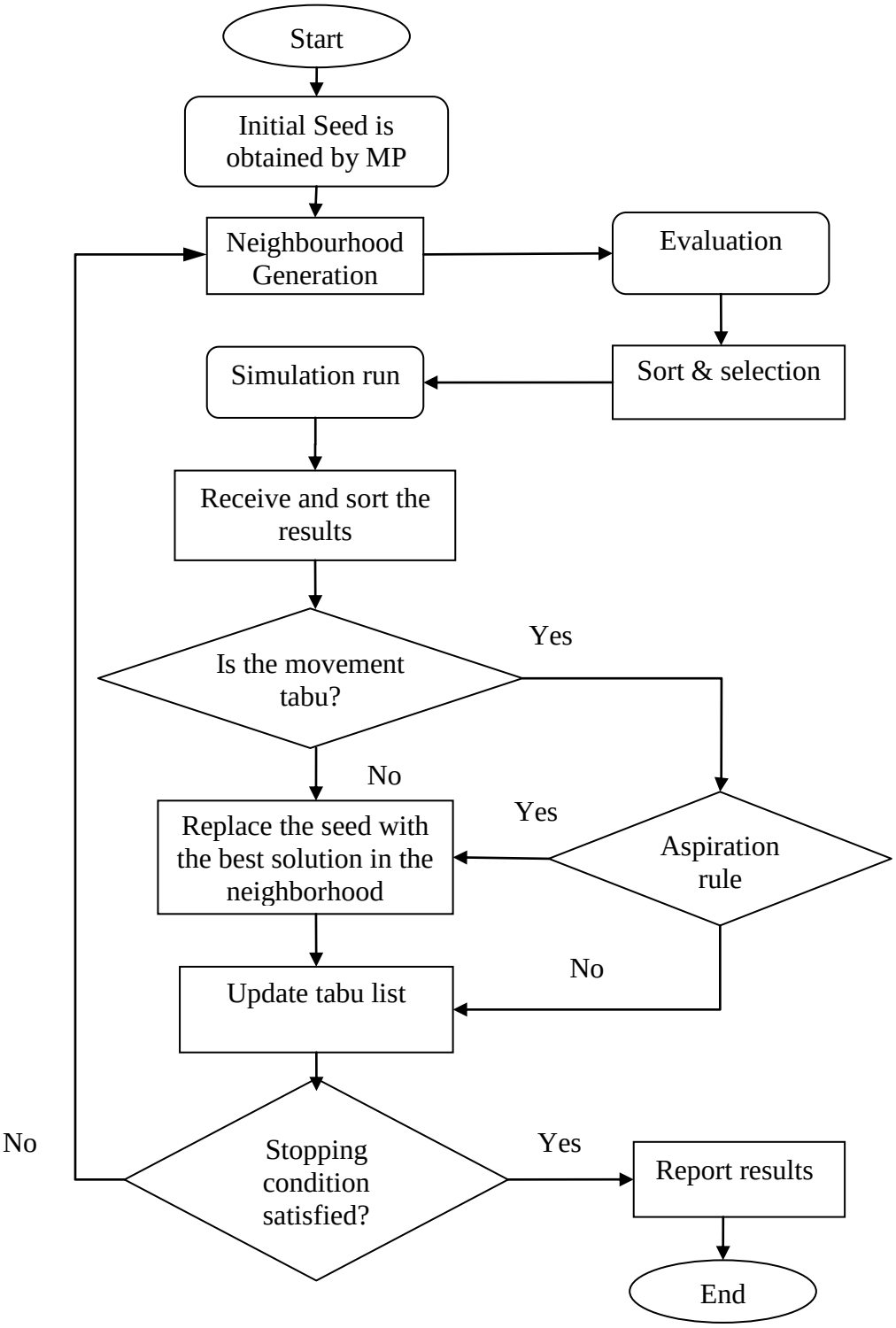

Figure 3.5 Flowchart of MPETS algorithm.

3.3.5. Enhancement of tabu search using a flowshop model

To enhance the performance of the tabu search, ETS determines an estimate of the average waiting time of patients to screen the solutions before submitting them to the simulation model. Average waiting time estimate is determined by calculating the average flow time of the schedules in a flowshop. The flowshop scheduling problem addresses scheduling problems that involve processing of each job on a series of ordered processers such as machines. The flowshop model is used by our algorithm to simply screen solutions in the tabu search and select the most promising solutions to be evaluated by the simulation model. This approach reduces the number of simulation runs and consequently the computational efforts.

The similarity between the OASP and flowshop scheduling problem suggests the idea that the optimal solution for one of these problems may be a sub-optimal or at least a promising solution for the other. This assumption is the ground for development of the flowshop model in the MPETS. However, as the flowshop problem by itself is an NP-hard problem, the flowshop model has to be kept as computationally inexpensive as possible.

A multiple machine flowshop problem with job families has been considered. The jobs in the flowshop context are equivalent to patients; while in each stage machines represent resources at stages of the clinic such as doctors. The processing time of each stage is assumed to be equal to the mean of service time probability distributions in the simulation. The same processing time has been applied in the mathematical programming model.

In order to estimate the average waiting time, the amount of average idle time of a job caused by the schedule in the flowshop model is calculated. This value will be considered as an estimate of the average waiting time of patients in the clinic. Equation (3-14) defines the average idle time in

the flowshop model where, f_i is the finishing time of processing job i, s_i is the starting time of processing job i, and $p_{i,j}$ is the processing time of job i on machine j. N is set of jobs (patients in the clinic) to be processed and M is the set of available machines (stages in the clinic).

$$\bar{I} = \frac{1}{|N|} \cdot \sum_{i \in N} \left(f_i - s_i - \sum_{j \in M} p_{i,j} \right) \quad (3\text{-}14)$$

3.4. Tests design and analysis

The proposed algorithms are coded using Microsoft Visual C#. The link between the tabu search and simulation package has been implemented using application of Arena™ object model. The simulation model has been developed in Rockwell Arena™ 12. In addition, the mathematical programming model has been developed as a part of MPETS. To solve the mathematical programming model, GAMS™ 225 software mathematical programming tool has been utilized.

In order to evaluate the performance of the ETS and MPETS, they have been compared with the OptQuest optimization package. The main reason of comparing the performance of proposed methods with OptQuest is that this tool has been applied in a recent work by Klassen and Yoogalingam [68]. This work is the first instance of using simulation-based optimization in OASP. In addition, OptQuest is a good choice for this comparison since it can employ the same simulation model which is used by the ETS and MPETS. Thus, this guarantees that all approaches have the same ground for comparison. Furthermore, OptQuest is a state-of-the-art simulation-based optimization tool, which has been extensively used in many industries.

Based on our preliminary experiments, three factors which may affect the performance of the proposed methods are identified. The three selected factors are the patient mix and the number of

patients, the type of probability distribution function used by the simulation, and the variance of the service time at each stage of the clinic.

To analyze the sensitivity of the results based on the selected probability distribution functions, three different probability distributions, at each stage of the clinic, have been considered in our experiments: a) the triangular distribution due to its simplicity which can be specified solely based on expert estimate of minimum, maximum, and most probable service time; b) the gamma distribution, and c) lognormal distribution, which is widely used in simulations models to represent length of processes and service times in the healthcare [94].

Table 3.1 Specifications of the test problems considering the patient mix and number of patients factors.

No	Number of patients	Number of Patient Type 1 & mean service time at 4 stages (min.)	Number of Patient Type 2 & mean service time at 4 stages (min.)	Number of Patient Type 3 & mean service time at 4 stages (min.)
1	15	6 (10,30,50,30)	7 (10,20,50,30)	2 (10,10,60,30)
2	24	6 (10 ,30,20,30)	10 (20,10,30,30)	8 (10,20,30,30)
3	38	14 (10,10,20,30)	13 (10,10,10,30)	11 (10,10,30,30)

In each of these distributions, the variance affects the range of expected variability in service time and the patient waiting time in the line. Three different values of 0, 4, and 8 minutes are considered for variance which represent low, medium, and high variances respectively. Table 3.1 includes the number of patients, mix of patients which indicates the number of patients of each type, and mean of service time of each stage for each type of patients in test problems. For instance, in row one of Table 3.1 under patient type one, 6 means the number of patients of type

one. Also, (10,30,50,30) addresses service times of four stages in the clinic for the patient type one in this test problem.

A test case with 38 patients is considered in order to gauge the complexity of the model at hand. MP model requires 3,828 integer variables and 2,785 constraints to present and solve the problem. In order to reduce the computation time, the integrality requirement is relaxed on all variables except for those involved in the arrival time of patients.

The parameters of all distributions (derived from specified mean and variance) can be set internally by the operator or externally by another program. The parameters of these distributions vary according to the type of patients. Given the historical data, the type and parameters of distributions can be set for modeling a specific clinic. The optimization strategies proposed in this chapter are independent of the choice of these parameters.

In order to evaluate the performance of the proposed methods, a design of experiment has been presented based on the Box-Behnken design (BBD). Considering previously selected factors, i.e., number of patients, service distribution, and service time variance, 27 different combinations are required if a full factorial DOE is employed. However, considering the large amount of computations required for this study, it is decided to use Box-Behnken design as the fractional (incomplete) experimental design to reduce the number of experiments from 27 test problems to 15 while preserving the quality of the experiments. BBD is considered as a rotatable design which is based on a three-level fractional factorial design. Table 3.2 presents the BBD design of experiments for our study. Myers and Montgomery [95] (page 322) is used to design the experiments.

For each of these 15 designs, the proposed methods are run 15 times (replications) which includes 5 runs for every three different seeds. In addition, for each design, OptQuest is run 15 times using three different seeds for simulation with five different initial solutions for each seed. The initial solutions are generated randomly. In total, 675 experiments are done in our study (225 experiments for each method).

Table 3.2 Design of experiments based on Box-Behnken design.

Number	Size (number of patients)	Distribution	Variance (min.)
1	15	Triangular	Medium (4)
2	15	Gamma	Medium (4)
3	38	Triangular	Medium (4)
4	38	Gamma	Medium (4)
5	15	Lognormal	Low (0)
6	15	Lognormal	High (8)
7	38	Lognormal	Low (0)
8	38	Lognormal	High (8)
9	24	Triangular	Low (0)
10	24	Triangular	High (8)
11	24	Gamma	Low (0)
12	24	Gamma	High (8)
13	24	Lognormal	Medium (4)
14	24	Lognormal	Medium (4)
15	24	Lognormal	Medium (4)

Table 3.3 shows the results of these experiments. This table includes the average waiting time of patients, as well as the computation time of all methods (ETS, MPETS, and OptQuest). The computation time for MPETS includes the mathematical programming model solution time. Our observation shows that although ETS is slightly faster than other methods, its quality of results is consistently under-performing both MPETS and OptQuest, and therefore might not be suitable for practical purposes. Therefore, our efforts are focused on analyzing MPETS and comparing its performance with the performance of OptQuest. The last two columns of Table 3.3 show the

results of statistical t-test with a null hypothesis of whether the means of the distributions of the results generated by MPETS and OptQuest, are equal. This work uses one-tail t-test with error type I equal to 5% for our analysis.

The results in Table 3.3 suggest that the mean of distribution of results presented by the MPETS is significantly different from the mean of distribution of results offered by OptQuest. Considering the value of wait times and the average computation time presented in the table, it is concluded that MPETS offers better results over OptQuest in terms of both the solution quality and computation time.

In order to investigate the factors' effect on the performance of the algorithms, a response surface method analysis has been carried out based on the differences between the values of solutions, and their computation times obtained by the MPETS and OptQuest. Three factors have been studied in order to learn which one has a significant impact on the quality or the computation time of the methods. Similar to previous analysis, the three selected factors are the patient mix and number of patients, type of probability distribution function, and variance of the service time at each stage of the clinic.

Table 3.4 demonstrates the results of our response surface analysis on the differences in the quality of solutions obtained from the MPETS and OptQuest. This table shows the coefficients of each factor in addition to the coefficient of square root of error (SE), the value of random variable in t-distribution (T), and the P-value of each term (P). The results show that the number and mix of patients are the most important factors influencing effectiveness of the algorithms, since this factor and its quadratic term present a very small P-value.

Table 3.3 Comparison results of MPETS, ETS and OptQuest.

No	Test problem	Effectiveness (Average Waiting time, minutes)			Efficiency (computation time, seconds)			MPETS & OptQuest comparison (T-test)	
		ETS	MPETS	OptQuest	ETS	MPETS	OptQuest	P-Value	Null Hypothesis
1	15_Tri_4	2.095	0.096	0.489	34	50	156	0.002	Rejected
2	15_Gam_4	2.268	0.081	0.533	36	51	151	0.006	Rejected
3	38_Tri_4	6.253	1.772	3.124	75	91	162	0.000	Rejected
4	38_Gam_4	5.676	1.489	2.592	68	97	185	0.000	Rejected
5	15_Log_0	2.805	0.006	0.172	36	70	169	0.014	Rejected
6	15_Log_8	2.783	0.116	0.695	37	67	178	0.000	Rejected
7	38_Log_0	3.763	0.076	0.975	64	81	184	0.000	Rejected
8	38_Log_8	6.651	2.262	3.740	71	83	165	0.000	Rejected
9	24_Tri_0	6.640	0.883	3.438	38	79	158	0.000	Rejected
10	24_Tri_8	8.240	3.600	4.952	41	78	176	0.002	Rejected
11	24_Gam_0	6.128	0.584	3.590	43	86	177	0.000	Rejected
12	24_Gam_8	7.839	3.004	4.412	37	98	180	0.000	Rejected
13	24_Log_4	7.814	2.436	4.088	45	97	177	0.000	Rejected
14	24_Log_4	6.031	2.692	4.613	45	88	183	0.000	Rejected
15	24_Log_4	7.062	2.341	4.462	46	91	175	0.000	Rejected

The variance of service time distribution is the second significant factor for the performance of algorithms. The P-value reported by the analysis is less than type I error=0.05, which means the factor is significant. In addition, the analysis proposes a negative value for the coefficient. Therefore, it indicates that although based on the t-test analysis MPETS always performed better in the test problems, the gap between the performances of MPETS and OptQuest algorithms decreases as the service time variances increases.

Table 3.4 ANOVA results on the difference of solutions obtained from OptQuest and MPETS.

Term	Coef.	T	P
Constant	1.89823	12.558	0
Patient number	0.40509	4.376	0
Distribution	0.03973	0.429	0.668
Variance	-0.22648	-2.447	0.015
Patient number*Patient number	-1.18621	-8.706	0
Distribution*Distribution	0.11281	0.828	0.409
Variance*Variance	0.069	0.506	0.613
Patient number*Distribution	-0.07707	-0.589	0.557
Patient number*Variance	0.04169	0.319	0.75
Distribution*Variance	-0.09874	-0.754	0.451

The same study has been performed on the gap between the computational time of OptQuest and MPETS. Table 3.5 presents the result of this analysis.

Table 3.5 ANOVA results of the difference of computation time of OptQuest and MPETS.

Term	Coef	T	P
Constant	86.3248	27.646	0
Patient number	-17.0309	-8.907	0
Distribution	0.5668	0.296	0.767
Variance	7.9011	4.132	0
Patient number*Patient number	0.3482	0.124	0.902
Distribution*Distribution	4.7277	1.68	0.094
Variance*Variance	-3.6504	-1.297	0.196
Patient number*Distribution	6.2466	2.31	0.022
Patient number*Variance	8.2796	3.062	0.002
Distribution*Variance	-6.9967	-2.587	0.01

The analysis reveals that the number and mix of patients have the most significant effects on the difference between the computation times of MPETS and OptQuest, since the P-value relevant to the patient number is zero. The coefficient for this factor is negative which suggests the larger the problem size, the less difference in computation time of two methods is observed. The

variance significantly affects the difference in the computation time of two algorithms as its P-value is zero. The positive coefficient regarding the variance factor suggests that as the service time variance increases, the difference between two algorithms increases. The average of computation times for the test problems indicates that MPETS needs less time than OptQuest. Hence, it is concluded that the OptQuest needs to do more computations as service time variance increases.

3.5. Conclusions

This chapter proposes two simulation-based optimization methods (ETS and MPETS) to address the lack of an effective and efficient method to solve the OASP. ETS is developed based on integration of a simulation model with a tabu search which uses an auxiliary objective to decrease simulation runs. MPETS enhances the ETS by using a mathematical programming model.

In order to examine the performance of proposed methods, several test problems have been developed based on the factors deemed to affect the performance of algorithms. These factors include the patient mix and number of patients, the type of probability distribution function used by the simulation, and the variance of the service time at each stage of the clinic. Box-Behnken design of experiment is used to conduct the experiments. The results are then compared with the results generated from OptQuest on effectiveness and efficiency in terms of producing quality results and the computing time to achieve the results. Our observations show that although ETS is slightly faster than other methods, its quality of results is consistently under-performing both MPETS and OptQuest, and therefore ETS may not be suitable for practical purposes. The comparison of ETS and MPETS reveals that the incorporation of the mathematical programming model can drastically improve the simulation-based optimization algorithms. Our experiments

further demonstrated that MPETS method outperforms OptQuest in terms of both the solution quality and computation time in all instances.

In addition to the comparison of three algorithms, an analysis has been carried out to examine the effect of the selected factors on the observed differences in the performance of MPETS and OptQuest. Our study suggests that the number and mix of patients are the most important factors influencing effectiveness and efficiency of the algorithms. The variance of service time distribution is the second significant factor affecting the performance of the algorithms. Our analysis shows that although MPETS performed better in all tested problems, the gap between the performances of MPETS and OptQuest algorithms decreases as the service time variance increases. On the other hand, compared to MPETS, OptQuest needs significantly more computation time as service time variance increases. However, the larger the problem size, the less difference in computation time is observed between MPETS and OptQuest. Among studied methods, MPETS appears to be a reliable method with superior effectiveness and acceptable level of efficiency.

Several directions for future research are apparent from this study. First, the proposed methods could be expanded to address scheduling challenges in surgical departments. For instance, challenges of assuming multiple ORs (multiple parallel servers in a single stage) in OR department is an avenue to extend the proposed method. Moreover, including constraints on the compatibility of resources with patient types in addition to introducing time-window constraints for the availability of resources in the model, would further reflect the complexities that exist in scheduling of OR departments. These problems will be dealt with in Chapter four.

CHAPTER FOUR

4. Appointment scheduling of operating room department

4.1. Introduction

Operating room (OR) departments are in a constant battle to use their limited resources in order to serve a maximum number of patients. Appointment scheduling plays an important role in this context, by providing a smooth flow of patients while minimizing the patient waiting time, completion time, and number of cancellations in OR departments. In this chapter, appointment scheduling refers to the determination of the time at which each patient should arrive at the OR department, and waiting time is the time that a patient spends in the facility waiting to be served. Completion time refers to the time that the last patient leaves the OR department. Case cancellations, simply called cancellations here, refer to scheduled surgery cases that are cancelled due to lack of time or resources. Scheduling in this environment is a challenging task due to the stochastic service times and constrained resources. Surgery scheduling is not only restricted by the availability of resources, but also constrained by the compatibility requirements (e.g., only a specific surgeon type can serve a patient type). In this study, we focus on outpatient surgeries, in which patients leave the system on the same day after receiving the service.

This chapter considers minimizing the waiting time of patients, and the completion time of OR department while monitoring the cancellation of scheduled cases. As reported by several studies

such as Gül et al. [69], and Klassen and Yoogalingam [68], in this environment, improving one measure often leads to the deterioration of other criteria. For example, minimizing waiting time may decrease the utilization, or increase the completion time and number of cancellations in the OR department.

Previous studies have applied optimization or simulation methods to schedule surgery cases. Typically, optimization methods use analytical approaches to achieve optimal (or near optimal) solutions. These approaches have difficulty addressing large-complex systems and, therefore, have often focused on elements of the system, or have overly simplified the system. For instance, many optimization methods consider only single stage systems with Exponential or Erlang distributions for service times. On the other hand, simulation methods are capable of addressing complexities of large systems. Hence, simulation literature has considered detailed multistage systems with constraints on resources, accounting for several environmental factors such as patient priorities, unpunctual patients, and different service time distributions. However, simulation approaches are time-consuming and often do not deliver a competitive optimization strategy. Therefore, ample opportunities exist in the literature questing for efficient and effective methods to address the challenges in outpatient surgery scheduling. The *efficiency* of a method refers to the amount of computation time required by the method to produce meaningful results, while *effectiveness* addresses the quality of solutions generated.

In this study, the *discrete-event* simulation model, hereafter called simulation model, is integrated with metaheuristics, to propose three simulation-based optimization methods, and further improve the performance of the proposed methods using mathematical programming (MP). The proposed methods address the problem of appointment scheduling of a predetermined number of patients of different types with stochastic durations in a multistage OR department.

The availability of several resources including ORs, recovery beds, and human resources such as surgeons are considered. Furthermore, other constraints such as the compatibility of resources and the number of available surgeons for each surgeon type are considered in our model. In addition, each surgeon is constrained by a time window, which indicates his/her availability in the scheduling horizon.

The first method, termed simulation-based tabu search (STS), integrates simulation with tabu search. The second method, integer programming enhanced tabu search (IPETS), improves the tabu search by incorporating an integer programming model. The third method, binary programming enhanced tabu search (BPETS) uses a binary programming model along with a heuristic and simulation-based tabu search to solve the problem.

In order to evaluate the performance of proposed methods, a number of test problems have been developed based on our findings in the understudy OR department over an extended range of three major factors, namely, the number of patients, number of ORs, and coefficient of variability of service time. The proposed methods then have been analyzed based on their performance in terms of solution quality and the computation time. Furthermore, several scheduling rules (such as shortest/longest processing time, etc.) are applied and compared with the proposed methods. This study provides insights on applications of the proposed scheduling approaches to assist practitioners. Additionally, BPETS is applied in a case study of an OR department in a major Canadian hospital and the results are compared with those of the actual schedules used in the OR department for several days.

In a very recent study, Gül et al. [69] presented a simulation-based multiobjective genetic algorithm for the appointment scheduling of an outpatient procedure center. They found that the

metaheuristics approaches did not offer schedules that are superior to the rules used for a scheduling period of one day. This interesting finding is studied and discussed later in this chapter.

In summary, existing analytical methods have difficulty in addressing large and complex systems, and therefore, mainly have focused on the elements of the system or simplified models of the system. Simulation methods can address many complexities in large systems, but are time-consuming and often do not deliver a competitive optimization strategy. Therefore, a lack of efficient and effective methods in existing literature is observed. These methods are expected to provide optimal or near-optimal solutions while encompassing details of real complex OR departments. In these OR departments, usually care providers deal with multistage systems that are constrained by limited resources while serving patients of different arrival and service time distributions.

This chapter tries to address the patient appointment scheduling problem under the following conditions, which eliminate many assumptions currently used in literature and should bring the problem closer to the reality. It addresses the challenges of a multistage OR department that serves different types of patients with possibly different stochastic service times at each stage. In contrast to most existing articles, the proposed approach addresses the stochastic service time of patients at each stage, regardless of the type of their probability distribution function. An OR department which includes multiple ORs and multiple recovery beds has been considered. In addition to resource availability constraints, the compatibility of resources is addressed (i.e., each patient type can be only served by a specific surgeon type).

Each surgeon follows a time window constraint, which determines the number of available surgeons of each type at each time block. The schedule is generated according to the Master Surgery Schedule (MSS) decided by management.

Based on simulation models, new optimization approaches are proposed to efficiently search for reliable solutions. Furthermore, the proposed method is applied to a case study OR department and its performance is compared with the actual schedules.

Finally, the application of several scheduling rules in appointment scheduling and their impacts on the waiting time and completion time are discussed. This study results in insight for practitioners who seek practical approaches for patient scheduling.

The remainder of the chapter has been organized as follows: Section 4.2 states the problem definition. Section 4.3 introduces the architecture of the proposed methods. Section 4 describes the design of experiments and presents analysis of the results. Section 4.5 presents the case study of an OR department. Finally, Section 4.6 discusses the conclusions.

4.2. Problem description

In a typical OR department, each patient goes through three stages: pre-operation (surgery preparation), surgery, and recovery. In the first stage, an OR nurse identifies the patient and extracts patient's charts and information such as lab results and consent forms. This stage includes different preparation procedures required for each type of surgery such as taking drugs and anesthetics, having blood tests, and waiting for the medicine to take effect. The surgery stage includes anesthesia and operation. The last stage accounts for the procedures and the time required for patient's recovery. Figure 4.1 represents the three stages of an OR department.

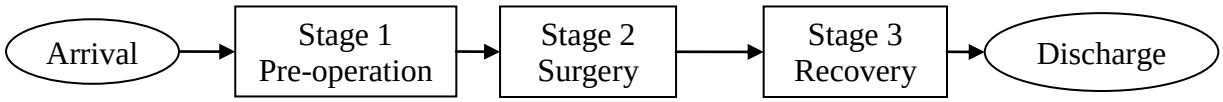

Figure 4.1 Stages of an OR department.

Appointment scheduling refers to the determination of the time at which each patient should arrive at the OR department. This chapter addresses the appointment scheduling of a specified number of patients in order to minimize the waiting time of patients, completion time, and number of cancellations. The completion time refers to the time that the last served patient leaves the post-anesthesia care unit (PACU). The cancellations account for the patients who could not be served due to the lack of time or resources. The number of patients and patient types are pre-determined in the higher-level planning according to the available resources. The problem considers several patient types with stochastic service time at each stage whose arrivals are punctual. The stages in the department (except for the first stage) work according to the first-come-first-served rule. The first stage admits the patients according to the schedule. Each type of patient is served by a specific specialty, i.e., the patient is served by a specific surgeon type, which intensifies the importance of resource compatibility in this problem.

Each type of surgeon encompasses a number of doctors of the same specialty. The number of available surgeons for each surgeon type per time block is provided by surgeon schedules, which is determined based on the surgeons' availability time window. The surgeon schedules are generated using master surgery scheduling (MSS), which is often developed by management, in the tactical level. In addition, the problem considers resources such as available ORs and available beds in PACU.

To provide a better understanding of the problem, consider scheduling of five patients of three different types. The specification of these patients is provided in Table 4.1.

Table 4.1 Specification of patients.

Patient	Patient type	Pre-operation time	Surgery time	Recovery time
1	Cardiac	LOGN(48.8, 12.4)	391 * BETA(2.94, 4.96)	240 + EXPO(71.5)
2	Neurology	TRIA(14.5, 50, 77.5)	490 * BETA(2.16, 6.3)	45 + 270 * BETA(1.76, 4.06)
3	Orthopedic	14.5 + GAMM(17.7, 1.39)	LOGN(132, 200)	40 + WEIB(93.2, 1.22)
4	Orthopedic	14.5 + GAMM(17.7, 1.39)	LOGN(132, 200)	40 + WEIB(93.2, 1.22)
5	Neurology	TRIA(14.5, 50, 77.5)	490 * BETA(2.16, 6.3)	45 + 270 * BETA(1.76, 4.06)

In the interest of brevity, we used LOGN, TRIA, GAMM, BETA, WEIB, and EXPO to represent lognormal, triangular, gamma, beta, weibull, and exponential distributions, respectively. A specific type of surgeon serves each patient type. Assume that the department in this example includes two ORs that are shared by four surgeons. There are two beds in the preparation holding area, and two PACU beds. Surgeons' availability and type are provided in Figure 4.2. Considering the patients' type, the availability of the surgeons, and the number of resources, a possible schedule is showed. The circles represent patients and indicate the time that the patient is expected to arrive.

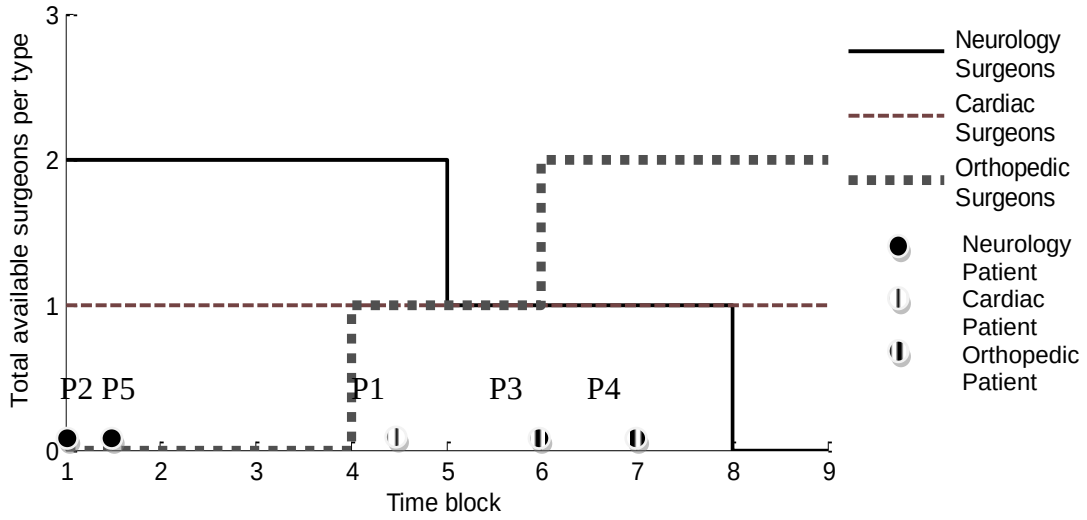

Figure 4.2 Number of available surgeons in a session and a possible schedule.

4.3. Methodology

This chapter proposes three simulation-based tabu search methods for outpatient scheduling in OR departments. The proposed STS method integrates simulation with tabu search. IPETS and BPETS methods improve on STS by incorporating an integer and binary programming models, respectively, with STS. The integer and binary programming in the last two methods solve the deterministic version of the problem. Tabu search utilizes the result of deterministic models as the initial solution. All proposed methods share the simulation model and tabu search components. In order to examine the applicability of proposed methods, the performance of BPETS method is studied on a simulation model built based on the actual data of an OR department in a Canadian hospital.

Furthermore, in this section, it is attempted to study the application of scheduling rules in appointment scheduling of an OR department, which functions as a reference and comparison to the proposed methods.

4.3.1. Tabu search

Tabu search (TS) [96] is a metaheuristic method that has been successfully used to solve many optimization problems. This method iteratively proposes solutions using heuristic procedures while employing a flexible memory to guide the exploration in the solution space. Employing the memory and examining properties of solutions, it determines the search direction. In each iteration, a neighborhood of solutions is built based on a seed solution. This neighborhood consists of all solutions generated from the seed solution modified by the heuristic procedure. This chapter only describe the details of tabu search that is specific to this research. Readers are recommended to refer to Glover and Laguna [81] for the detail of tabu search algorithms.

Tabu search in this work starts by determining an initial solution. In the STS method, an arbitrary schedule of surgeries has been used as the initial solution. IPETS and BPETS, however, improve the performance of tabu search by starting the search from a more promising initial solution (generated by IP). The neighborhood of solutions is generated based on the determined initial solution by means of swap and insertion heuristics. In other words, the neighborhood consists of different surgery schedules that are generated based on the initial schedule. In STS, a number of solutions in the neighborhood are evaluated by simulation. The best solution based on the result of evaluations is nominated as the seed solution for the next iteration.

In order to reduce the number of simulation runs, BPETS and IPETS do not evaluate all solutions. They utilize a heuristic called *deterministic scheduling module* (DSM) which

approximates the average waiting time and completion time to rank the solutions before evaluating them by simulation. Solutions in the neighborhood are ranked based on their objective value and a selected number of best solutions are put in the *candidate list.* The solutions in the candidate list are then evaluated using the simulation model. The best solution, based on the simulation model results, is then nominated as the seed solution for the next iteration.

In the next step, the nominated solution is checked against the tabu list and is used as the seed for the next iteration as long as it is not a tabu solution. However, a tabu solution may be used for the next iteration if it satisfies aspiration conditions. Tabu search iteratively continues the described procedure till the termination condition is satisfied, which is set as the maximum allowed number of iterations in the proposed methods.

Although tabu search is a powerful optimization tool, its application does not guarantee reaching the optimal solution. Furthermore, it hardly uses the knowledge of the problem. In order to develop an efficient and effective optimization method, employing other methods to get a good initial solution seems beneficial. This strategy has been empoyed in IPETS and BPETS.

In order to incorporate the average waiting time of patients, along with the completion time of the facility, and the number of cancellations in the objective function of tabu search, a linear weighted-sum of these terms is used. Expression (4-1) presents the objective function used by tabu search, where n is the number of patients, and w_i is the waiting time of patient i. The objective function includes the completion time, m, and the number of cancellations, v. Coefficients $\alpha, \eta, and\ \xi$ are the weights that are assigned based on the empirical data or managerial preferences to reflect the relative importance of each term.

$$\alpha \sum_{i=1}^{n} \frac{w_i}{n} + \eta . m + \xi . v \qquad (4\text{-}1)$$

4.3.2. Integer programming enhanced simulation-based tabu search (IPETS)

IPETS incorporates integer programming (IP) with simulation-based tabu search. The IP model represents the deterministic version of the problem, which uses the mean of service time distributions as the service time to construct the initial solution neighborhood of tabu search. IPETS uses the appointment schedule yielded from solving IP model. The integer programming model proposed in this work is an extension of the model provided in the previous chapter. The previous models are expanded to capture the complexities in appointment scheduling.

In the IP model, the scheduling horizon is divided into a number of equal-length time grids. Each time block consists of one or more time grids. The discrete timeline assumes that the length of service times is an integer multiple of the basic time grid length. In addition, it assumes that each process starts only at the beginning of a time grid. The output of the IP model provides a surgery schedule with the minimum wait time and completion time assuming deterministic service times. The notation of the model is presented as follows:

Notation:

t discrete time index, *t*=1, ..., T, where T is the time horizon and number of time grids in each day;

j stage index, *j*=1, 2, 3, where stage 1, 2, and 3 represent the pre-operation, operation (ORs), and post-operation (PACU) stages. Please note that the model is not restricted to the three stages and can be applied for cases with more than three stages;

p patient type index, p=1, ..., P, where P is the number of patient types;

s surgeon type index, s=1, …, S, where S is the number of surgeon types;

B_s set of patient types who can be served by surgeon type s. $p \in B_s$ means that the patient type p can only be served by the surgeon type s.

Parameters:

o_t total capacity of operating rooms at time t;

$l_{s,t}$ number of available surgeons of type s at time t;

$d_{j,p}$ service duration of patient type p in stage j;

$I_{j,p}$ initial number of patients of type p in the line, waiting to be served at stage j (including the patients who have appointments in the first stage);

R_j number of available resources in stage j at the beginning of the scheduling horizon;

M an arbitrarily large number;

γ_p penalty coefficient of waiting a single time grid for a patient of type p;

β penalty coefficient of operating the clinic per time grid;

n_p number of nurses required for a patient of type p in PACU.

Variables:

$x_{j,t,p}$ number of patients of type p at stage j to start being processed at time t;

$q_{j,t,p}$ number of waiting patients of type p to be served at stage j at time t;

$X_{j,t,p}$ cumulative number of patients of type p at stage j, whose treatment started by time t;

$r_{j,t}$ number of available idle resources at stage j and time t; each stage has its dedicated resource;

m last time block, in which all patients have been discharged (completion time);

y_t equals 1 if some patient is discharged at time t; 0 otherwise.

The optimization model is expressed as follows:

- Objective function:

$$\text{Minimizing } \sum_j \sum_t \sum_p \gamma_p q_{j,t,p} + \beta m \tag{4-2}$$

- Queue balance constraints:

$$q_{j,t,p} = I_{j,p} - X_{j,t,p} + X_{j-1,t-d_{j-1,p},p} \qquad \forall j,t,p. \tag{4-3}$$

- Cumulative variables definition:

$$X_{j,t,p} = \sum_{\tau=1}^{t} x_{j,\tau,p} \qquad \forall j,t,p. \tag{4-4}$$

- Capacity constraints:

$$r_{j,t} = R_j - \sum_p X_{j,t,p} + \sum_p X_{j,t-d_{j,p},p} \qquad \forall j = 1,2, \forall t, \tag{4-5}$$

$$r_{3,t} = R_3 - \sum_p n_p X_{3,t,p} + \sum_p n_p X_{3,t-d_{3,p},p} \qquad \forall t. \tag{4-6}$$

- Surgeon availability constraint:

$$\sum_{p \in B_s} (X_{2,t,p} - X_{2,t-d_{2,p},p}) \le l_{s,t} \qquad \forall t,s. \tag{4-7}$$

- OR availability constraint:

$$\sum_{p}(X_{2,t,p} - X_{2,t-d_{2,p},p}) \le o_t \qquad \forall t. \qquad (4\text{-}8)$$

- Number of patients who have to be served:

$$X_{3,T-d_{3,p},p} = \sum_{j} I_{j,p} \qquad \forall p. \qquad (4\text{-}9)$$

- The completion time indicator constraint:

$$My_t \ge \sum_{p} x_{3,t-d_{3,p},p} \qquad \forall t, \qquad (4\text{-}10)$$

$$m \ge t.y_t \qquad \forall t. \qquad (4\text{-}11)$$

- Initial conditions and integer constraints:

  $x_{j,t,p}$ *and* $r_{j,t}$ are non-negative integer variables. The values for $X_{j,t,p}$, $x_{j,t,p,}$ *and* $r_{j,t}$ should be pre-specified for $t<0$ if applicable.

Expression (4-2) minimizes the total penalized sum of patients' waiting time and completion time. Equation (4-3) defines the relationship among stages and maintains the patient flow within the system. It states that the number of patients in the queue of each stage is equal to the number of patients who were initially there plus the patients who entered the stage subtracted by patients who left so far. Equation (4-4) determines the cumulative variables that address the total number of patients processed in each stage.

Equation (4-5) controls the number of available general resources in all stages through time. Resources such as receptionists and surgery preparation nurses are handled by this constraint. Equation (4-6) addresses resource capacity of PACU. Expression (4-7) requires that there are enough number of surgeons of each type needed for surgery at each time block, considering the

surgeons' availability and schedule. Expression (4-8) ensures that the total number of ongoing surgeries at each time block does not exceed the available number of ORs. Equation (4-9) stipulates that all patients will be served by the end of the session. Expression (4-10) and (4-11) determine the last patients' discharge time block (completion time).

The Initial condition states that for equations such as Equations (4-3 to 4-5), the time block index is determined by terms in which *t* is subtracted by the service time of patients. Having such a term in the time block index results in negative values when *t* is smaller than service time. In this case, model need to assign zero values to the variables with negative time block index as an initial condition.

In contrast to the simulation model, IP model assumes that the service times must be multiples of a basic time grid. Note that the basic time grid of 15 minutes is used in the experiments. Using shorter time grids will result in an increase in the number of variables in the model, which increases the computation efforts required to solve the model.

Although the IP model presents optimal surgery schedules for small to medium size problems, it may not efficiently provide optimal solutions for large problems due to huge computational efforts needed to solve the problems. Therefore, the integrality constraints in the integer programming model are relaxed and a new binary programming based algorithm is proposed in the next section.

MP model (2) is differentiated from the MP model (1) that was presented in Chapter three in following aspects:

- MP model (2) addresses the completion time concept — including objective function and relevant constraints — in addition to the waiting time.

- Resource compatibility constraints by which the appropriate type of surgeon is assigned to the each patient type are included in MP model (2).
- MP model (2) introduces time-window constraints which manage the availability of surgeons in the model.

4.3.3. Binary programming simulation-based tabu search (BPETS)

BPETS incorporates binary programming with simulation-based tabu search. The BP model used in this method is similar to the IP model used in IPETS while relaxing the integer assumption on the variables. Consequently, its optimal results may contain non-integer values in schedules. For example, it may suggest that a non-integer number of patients appear at several time blocks. To remedy this problem, a heuristic, termed *integer appointment constructor* (IAC) is developed that takes the result of linear programming model as the input and converts it to a feasible integer schedule as the output. IAC considers the schedule generated by the BP model for each type of patients, and constructs appointments accordingly. The details of IAC are as follows:

p patient type index; $p=1, ..., P$, where P is the number of patient types;

n_p^0 number of patients of type p;

$\bar{S}_p = \{\bar{s}_t\}$ integer arrival schedule array for type p patients, in which the number of arriving patients at time t, $\bar{s}_t$, is integer;

$\underline{S}_p = \{\underline{s}_t\}$ non-integer arrival schedule array for type p patients, in which the number of arriving patients at time t, $\underline{s}_t$, is not integer;

n_p^n the sum of all items of $\underline{S}_p$;

n_c counter for non-integer arrivals;

Algorithm 1 (Integer Appointment Constructor)

Step 0: Increment p by one; if $p \leq P$, determine the number of patients in patient type p, n_p^0 from input string; otherwise stop.

Step 1: Determine $\bar{S}_p$ and $\underline{S}_p$.

Step 2: Determine n_p^n; set $n_c = 0$; set $t^* = 0$.

Step 3: If $n_p^n = 0$, go to Step 0 ; otherwise, for t=1,…,T, set $n_c = n_c + \underline{s}_t$ until n_c is an integer; set $t^* = t$;

Step 4: for t=1, …, t^*, select the n_c number of highest values of $\{\underline{s}_t\}$; increment the equivalent items in $\bar{S}_p$ by one, and for t=1,…, t^*, set $\underline{s}_t = 0$; go to Step 2.

Figure 4.3 shows an example of IAC algorithm assuming all patients are of the same type. Here, $\underline{S}_p = \{0.15, 0.55, 0, 0, 0.30,\ 0.33, 0, 0, 0.66, 0\}$, and in the first iteration $n_p^n = 2$ and $t^* = 5$ with $n_c = 1$. Then for t=1, …, 5, $\underline{s}_t = 0$ and $\bar{s}_2 = 1$. Then, in the second iteration,$n_p^n = 1$ and $t^* = 9$ with $n_c = 1$. For $t = 1, \ldots, 9$, $\underline{s}_t = 0$ and $\bar{s}_9 = 1$. The two largest $\underline{s}_t$ are located in the second and the ninth position of $\bar{S}_p$ in the first and second iteration. The algorithm is terminated in the third iteration as $n_p^n = 0$.

Time	Arrivals
1	1.15
2	0.55
3	0
4	0
5	0.30
6	0.33
7	0
8	0
9	0.66
10	1

Time	Arrival
1	1
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	1

Time	Arrival
1	0.15
2	0.55
3	0
4	0
5	0.30
6	0.33
7	0
8	0
9	0.66
10	0

Time	Arrival
1	1
2	1
3	0
4	0
5	0
6	0
7	0
8	0
9	1
10	1

Figure 4.3 An example of IAC heuristic.

4.3.4. Simulation model

The proposed methods use a *discrete-event simulation* model to evaluate the schedules generated by optimization components. The simulation model encompasses three stages of the OR department including holding area, ORs, and PACU, as depicted in Figure 4.1. At the beginning of each simulation run, the simulation model accepts, as inputs, the patient schedule and parameters of the problem. Problem parameters include the number of patient types, number of patients in each patient type, information on patient service times at each stage, number of surgeon types, schedule of each surgeon type, number of ORs, and number of PACU beds.

Using stochastic service time in the simulation model, 30 replications of each simulation run for each schedule are performed. Based on several experiments, we determined that 30 replications is an appropriate tradeoff between computation time and half width of the confidence intervals. Over the 30 replications, the simulation model tallies and reports the results as the final output. The simulation model has been validated using actual data of 24 days of the OR department described in Section 4.5, considering the service times derived from actual data.

4.3.5. Scheduling rules

In this section, it is attempted to study the application of several scheduling rules in appointment scheduling of patients to establish a reference for comparing the results. Many practitioners rely on a first-come-first-served rule to assign the patient appointments based on available spots in the schedule. In order to address the variability of service times in appointment scheduling rules, a job hedging approach, introduced by Yellig and Mackulak [97] and later implemented in outpatient scheduling by Gül et al. [69] is used. In this approach, the adjusted service time, $S' = \mu + \alpha\sigma$, is used for scheduling, where μ is the mean of service time, σ is the standard deviation of service time, and $\alpha \in [0,1]$ is a real number. Furthermore, in this section we propose a new sequencing rule inspired by the finding of Robinson and Chen [28]. Also, sequencing patients based on adjusted service times in addition to solely rely on the mean of service time for scheduling is examined.

Following scheduling rules are considered: (a) *increasing mean of service time (shortest processing time, SPT),* (b) *decreasing mean of service time (longest processing time, LPT),* (c) *increasing variance of service time (SVR),* (d) *increasing coefficient of variability of service time (SCV),* and (e) *dome-shape rule (DSR).* In addition to using mean of service time, we considered applying SPT and DSR rules on adjusted service times, termed stochastic SPT (SSPT) and stochastic DSR (SDSR), respectively. SVR sorts patients increasingly according to their variance of service time. SCV sorts the patients increasingly according to coefficient of variation of their service time.

The DSR first decreasingly sorts the patients based on their service time. Then, it starts generating the patients' sequence by placing the patient with the largest service time in the middle of sequence and putting patients with less service time before and after it alternatively.

Robinson and Chen [28] suggested in their article that optimal appointment intervals present a dome pattern for patients' service time in appointment scheduling of an outpatient clinic. This insight from outpatient clinic scheduling is adopted and extended to surgery scheduling. For instance, consider six patients and their associated service times of {1, 4, 2, 1, 5, 6}. The sequence resulted by DSR will be {P1, P3, P5, P6, P2, P4}. Whereas, applying SPT and LPT results in {P1, P4, P3, P2, P5, P6} and {P6, P5, P2, P3, P4, P1} respectively. Figure 4.4 compares outcomes of three scheduling rules of DSR, SPT, and LPT for these patients. In Figure 4.4, the brackets represent the order of patients' arrival. For example, "Patient[3]" represents the third patient arriving. The resulted sequence indicates that DSR places the patients with longer service times in the middle of sequence while sets the ones with shorter service time in the beginning or end of the sequence.

Considering a specific patient sequence, the next step is the appointment time determination for each patient. As a performed study indicates that the OR stage is the bottleneck of the system, the OR service time is used to determine the appointment times. In order to determine the appointment time, following items are defined:

i patient index, $i=1, \ldots, N$; where N is number of patients;

l OR index, $l= 1, \ldots, n;$ where n is the number of ORs;

c_l variable which represents the earliest time that l^{th} OR becomes idle;

s'_i adjusted service time of i^{th} patient which is defined earlier in this section.

The earliest available OR is noted by k, and the time this OR becomes available is $c_k = min(c_l), l = 1\ to\ n$. The appointment time for patient is $A_i = c_k + s'_i$. The value of c_k is updated to $c_k + s'_i$ after the patient is assigned to the k^{th} OR.

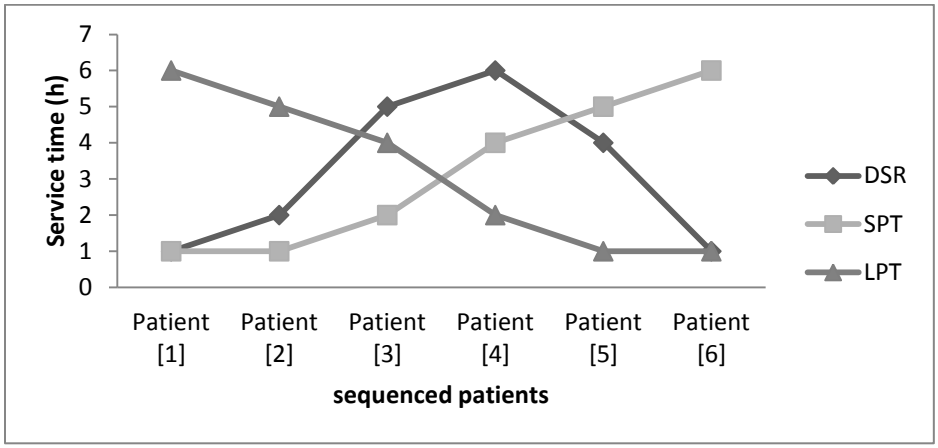

Figure 4.4 DSR, LPT and SPT scheduling rules

4.4. Experiments and results

In this study, the Rockwell Arena™ version 12 software was used for the simulation model; the proposed tabu method was implemented with Microsoft Visual C#, which was integrated with Arena™. The integer and binary programming models have been developed and solved by GAMS™ 225 software tool. A 2.53 GHz Intel® Core 2 Duo CPU with 3 GB RAM to perform experiments.

In order to develop a number of test problems, levels are considered of several factors that are expected to be the most effective on the performance of the algorithms. Based on our preliminary experiments, three factors have been identified – the number of patients, number of operating rooms and coefficient of variability of service.

To analyze the performance of the proposed methods with respect to the size of the problems, problems with 15, 33, and 50 patients per day are generated. This range not only covers but also exceeds the range of the number of patients served in the OR department that has been studied in the case study section. Depending on the type of surgeries, these problems may represent patient load of a medium to large OR department. Table 4.2 presents the configuration of test problems and includes the number of patients of each type and the number of resources. PT stands for patient type in the table.

Table 4.2 Specification of test problem based on patient types

Total # of patients	# of PACU Beds	# of patients per patient type									
		PT 1	PT 2	PT 3	PT 4	PT 5	PT 6	PT 7	PT 8	PT 9	PT 10
15	6	3	3	3	3	3					
33	10	4	2	4	5	5	4	4	5		
50	14	3	4	4	5	3	4	5	4	7	6

The second factor employed in the development of the test problems is the number of operating rooms. Two different values for each test problem are selected. Test problems with 15 patients include four and five ORs, while eight and nine ORs are used for the test problems with 33 patients. For the test problems with 50 patients, nine and ten operating rooms are used.

Since the variance of service times plays a significant role in the magnitude of actual service times at each stage, the coefficient of variability (CV) is used as the third factor involved in the design of test problems. CV is defined as the ratio of variance to the mean of service time. The same CV for the service time is used in the surgery stage. Three different levels of CV have been

used for the test problems, namely 0.5, 1, and 2. These values of CV are selected based on the extended range of the CV of service time distributions obtained in our case study for different patient types. In the design of test problems, the same type of patients who were available in our case study are used. For each patient type, the service times of Stages 1 and 3 are taken from the case study. The lognormal distribution function is used in the surgery stage as suggested by several studies, see for example Zhou and Dexter [98]. In the surgery stage, the mean is taken from the actual data obtained in the case study, and variance is modeled by a given level of CV. The proposed methods, however, are not limited to the chosen types of distribution functions.

Two studies are conducted to evaluate the performance of proposed methods. First, the three proposed methods are compared based on the quality of their solutions and the computation time. The quality of solution is assessed based on three criteria of average waiting time, completion time, and number of cancellations. The second study includes comparing the performance of the proposed methods with a set of scheduling rules.

4.4.1. Performance study of proposed methods

Based on the selected factors and the levels for each factor, 18 test problems are generated. Each test problem runs ten times, using a different random seed each time. In total, 180 runs are performed for each method. Each run is limited to 300 function evaluations.

Methods are examined based on their efficiency and effectiveness. The efficiency is measured based on the CPU time (in seconds) required by each method for completing 300 function evaluations; each function evaluation includes a run of simulation model. The completion time is inclusive of computation time of all steps of the considered method and not restricted to

simulation run. The effectiveness is measured based on the average waiting time, completion time, and number of cancellations resulted by each method.

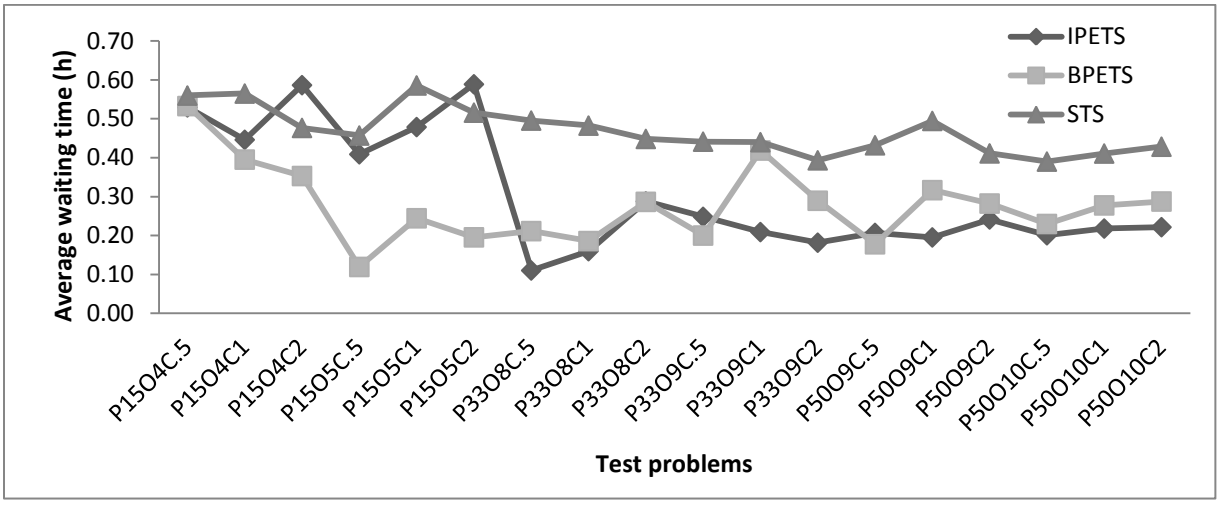

Figure 4.5 Comparison of the average waiting time from the proposed methods.

Figure 4.5 compares the average waiting time of patients yielded by the proposed methods. It is observed that as the size of the problem grows, the gap between the performance of the methods which benefit from a MP model (IPETS and BPETS), and STS becomes more significant, and the methods which use the MP model present superior results. Considering medium and large problems, IPETS delivers better results than STS for all instances, and competitive results compared to BPETS.

Figure 4.6 compares the performance of the proposed methods in terms of completion time. The completion times of the methods enhanced by MP are significantly less than those offered by STS. In this figure, a set of problems with the same number of patients and ORs is contrasted by drawing an oval around them. Similar ovals can be constructed for other sets with the same number of patients and ORs. The result suggests that the CV of surgery service time plays a

significant role in the completion times and increasing the CV significantly increases the completion time.

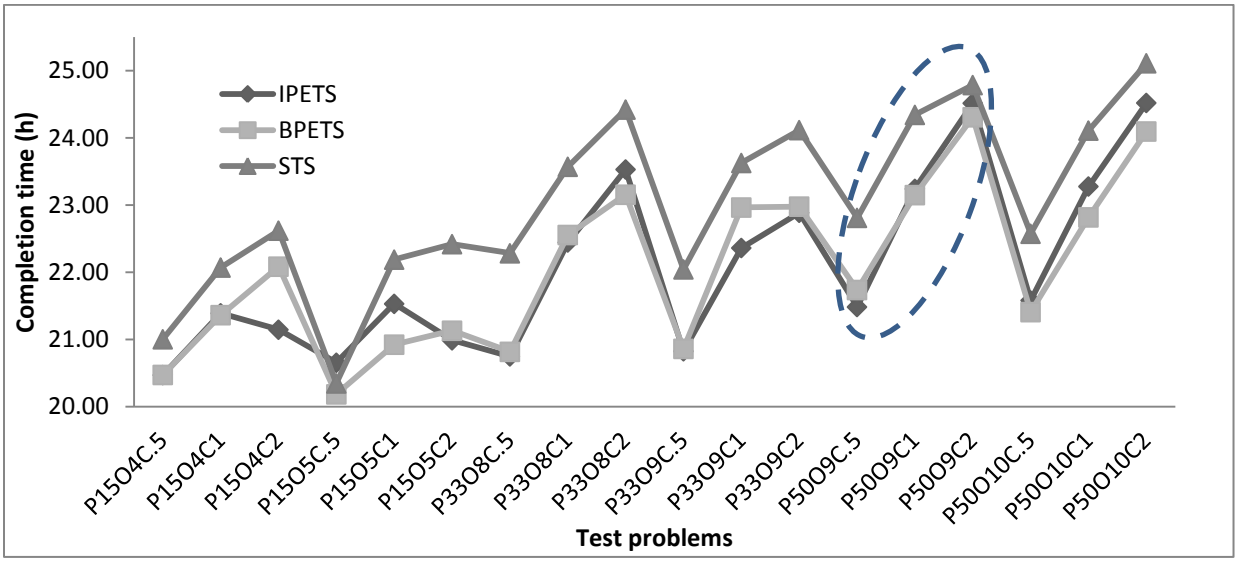

Figure 4.6 Comparison of the completion time of the proposed methods over 18 test problems.

Table 4.3 Comparison of proposed methods in terms of waiting time and completion time.

Test Problem	# of patients	# ORs	CV	95% CI of average waiting time (h)			95% CI of completion time (h)		
				IPETS	BPETS	STS	IPETS	BPETS	STS
P15O4C.5	15	4	0.5	[0.41,0.65]	[0.45,0.62]	[0.46,0.66]	[20.22,20.72]	[20.19,20.76]	[20.73,21.27]
P15O4C1.			1.0	[0.33,0.56]	[0.29,0.50]	[0.41,0.72]	[21.06,21.73]	[21.07,21.65]	[21.76,22.38]
P15O4C2.			2.0	[0.48,0.70]	[0.26,0.45]	[0.35,0.60]	[20.86,21.43]	[21.75,22.42]	[22.34,22.90]
P15O5C.5		5	0.5	[0.32,0.50]	[0.08,0.16]	[0.37,0.55]	[20.34,20.97]	[19.96,20.40]	[20.09,20.60]
P15O5C1.			1.0	[0.39,0.57]	[0.21,0.28]	[0.47,0.71]	[21.17,21.89]	[20.78,21.06]	[21.77,22.61]
P15O5C2.			2.0	[0.53,0.64]	[0.18,0.21]	[0.43,0.60]	[20.62,21.35]	[20.87,21.39]	[22.14,22.70]
P33O8C.5	33	8	0.5	[0.08,0.14]	[0.16,0.27]	[0.43,0.57]	[20.55,20.95]	[20.66,20.98]	[22.04,22.52]
P33O8C1.			1.0	[0.11,0.21]	[0.12,0.26]	[0.43,0.54]	[22.18,22.71]	[22.24,22.87]	[23.31,23.83]
P33O8C2.			2.0	[0.19,0.38]	[0.20,0.38]	[0.37,0.53]	[23.09,23.97]	[22.69,23.62]	[24.11,24.74]
P33O9C.5		9	0.5	[0.15,0.34]	[0.14,0.26]	[0.36,0.52]	[20.53,21.12]	[20.62,21.11]	[21.81,22.28]
P33O9C1.			1.0	[0.12,0.29]	[0.28,0.56]	[0.38,0.50]	[22.08,22.64]	[22.64,23.29]	[23.38,23.87]
P33O9C2.			2.0	[0.12,0.24]	[0.20,0.37]	[0.33,0.45]	[22.62,23.15]	[22.71,23.25]	[23.77,24.47]
P50O9C.5	50	9	0.5	[0.15,0.26]	[0.14,0.22]	[0.38,0.49]	[21.26,21.71]	[21.46,22.01]	[22.52,23.10]
P50O9C1.			1.0	[0.18,0.21]	[0.23,0.40]	[0.44,0.55]	[22.91,23.57]	[22.77,23.54]	[24.02,24.67]
P50O9C2.			2.0	[0.17,0.31]	[0.21,0.35]	[0.34,0.48]	[24.10,24.93]	[23.88,24.74]	[24.49,25.08]
P50O10C.5		10	0.5	[0.13,0.27]	[0.18,0.28]	[0.34,0.44]	[21.31,21.85]	[21.08,21.74]	[22.27,22.88]
P50O10C1.			1.0	[0.17,0.27]	[0.22,0.34]	[0.38,0.44]	[22.90,23.66]	[22.59,23.04]	[23.83,24.39]
P50O10C2.			2.0	[0.15,0.29]	[0.22,0.36]	[0.34,0.51]	[24.14,24.91]	[23.71,24.48]	[24.80,25.43]

Table 4.3 compares three proposed methods in terms of effectiveness classified based on the total number of patients, number of ORs, and CV of the problems. This table contains 95% confidence intervals (CI) on the mean for each test problem based on the results obtained over multiple runs. Considering the result presented in Table 4.3, Figure 4.5, and Figure 4.6, it is concluded that IPETS and BPETS outperform STS in terms of waiting time and completion time, as their CIs do not overlap the CIs of STS for most cases (Table 4.3) while they present better average values (Figure 5, Figure 6).

Figure 4.7 compares the average number of cancellations for proposed methods. It is shown that the number of cancellations delivered by all methods is comparable in small size problems. However, in the large size problems with large CV, the performance of BPETS is superior to the other methods. Overall, the number of cancellations grows as the number of patients and CV increase in the test problems.

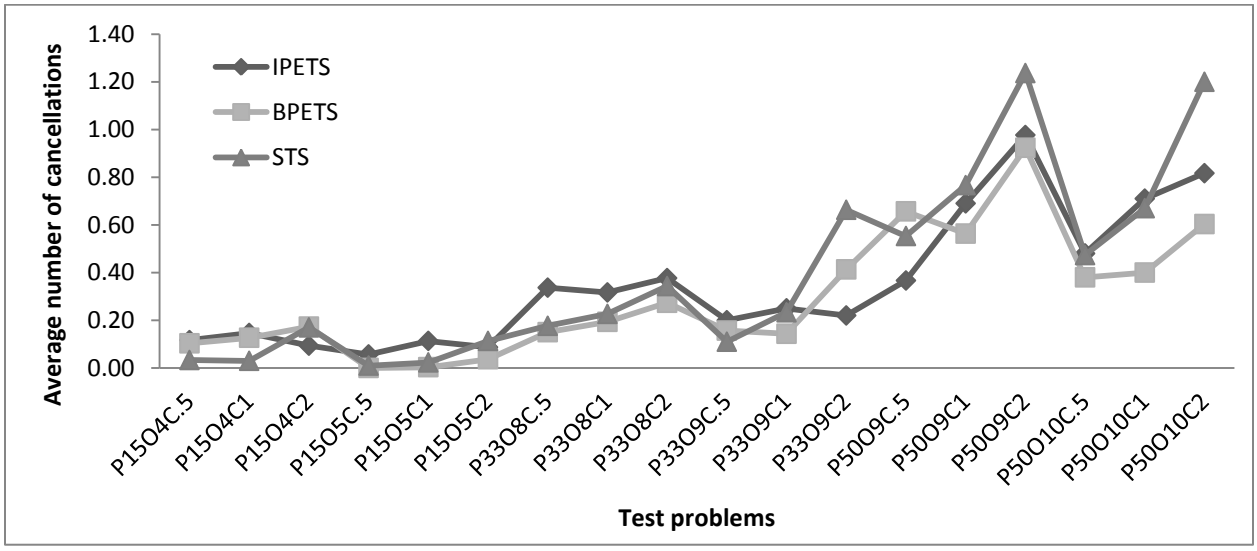

Figure 4.7 Comparison of the average number of cancellations of proposed methods.

Table 4.4 addresses the cancellation for the proposed methods. This table includes the 95% confidence intervals on the mean number of cancellations for each test problem. The number of cancellations rises as the number of patients and CV increase.

Table 4.4 Comparison of CI on the average number of cancellations

Test Problem	# of patients	# ORs	CV	95% CI of the number of cancellations		
				IPETS	BPETS	STS
P15O4C.5	15	4	0.5	[0.00,0.24]	[0.00,0.22]	[0.00,0.06]
P15O4C1.			1.0	[0.08,0.21]	[0.05,0.21]	[0.01,0.05]
P15O4C2.			2.0	[0.06,0.13]	[0.09,0.26]	[0.12,0.22]
P15O5C.5		5	0.5	[0.01,0.10]	[0.00,0.00]	[0.00,0.03]
P15O5C1.			1.0	[0.02,0.20]	[0.00,0.01]	[0.00,0.04]
P15O5C2.			2.0	[0.07,0.11]	[0.03,0.04]	[0.09,0.14]
P33O8C.5	33	8	0.5	[0.10,0.57]	[0.03,0.27]	[0.02,0.34]
P33O8C1.			1.0	[0.05,0.59]	[0.05,0.34]	[0.12,0.33]
P33O8C2.			2.0	[0.13,0.63]	[0.15,0.40]	[0.23,0.45]
P33O9C.5		9	0.5	[0.00,0.55]	[0.02,0.29]	[0.03,0.19]
P33O9C1.			1.0	[0.06,0.44]	[0.00,0.31]	[0.03,0.44]
P33O9C2.			2.0	[0.15,0.29]	[0.18,0.65]	[0.41,0.92]
P50O9C.5	50	9	0.5	[0.24,0.49]	[0.52,0.79]	[0.26,0.85]
P50O9C1.			1.0	[0.53,0.85]	[0.32,0.81]	[0.55,0.98]
P50O9C2.			2.0	[0.76,1.19]	[0.72,1.13]	[0.97,1.50]
P50O10C.5		10	0.5	[0.15,0.81]	[0.15,0.61]	[0.21,0.73]
P50O10C1.			1.0	[0.45,0.97]	[0.31,0.49]	[0.43,0.91]
P50O10C2.			2.0	[0.62,1.01]	[0.47,0.74]	[0.91,1.49]

Table 4.5 shows the 95% CIs on the average of computation times of the proposed methods. Computation time for BPETS and IPETS includes the time needed for solving the MP model and tabu search method. Overall, the computation times of the methods are comparable. For current test problems, it is observed that the integer programming model does not require a long computation time to solve. However, it is expected that if a larger number of patient types is

considered (e.g., if each specialty is broken down into major procedures), the computation time of IPETS would be significantly longer.

Table 4.5 95% CIs on average computation time of proposed methods.

Test Problem	# of patients	# ORs	CV	95% CI of Computation time (s)		
				IPETS	BPETS	STS
P15O4C.5	15	4	0.50	[1030,1045]	[1038,1046]	[1020,1037]
P15O4C1.			1.00	[1024,1032]	[1032,1042]	[1024,1034]
P15O4C2.			2.00	[1022,1037]	[1028,1036]	[1022,1032]
P15O5C.5		5	0.50	[1062,1073]	[1076,1092]	[1061,1073]
P15O5C1.			1.00	[1064,1075]	[1074,1089]	[1061,1076]
P15O5C2.			2.00	[1068,1078]	[1070,1084]	[1058,1071]
P33O8C.5	33	8	0.50	[1376,1385]	[1378,1393]	[1384,1400]
P33O8C1.			1.00	[1375,1398]	[1380,1396]	[1379,1398]
P33O8C2.			2.00	[1377,1401]	[1372,1389]	[1381,1395]
P33O9C.5		9	0.50	[1427,1440]	[1421,1441]	[1421,1439]
P33O9C1.			1.00	[1421,1441]	[1414,1429]	[1419,1440]
P33O9C2.			2.00	[1417,1435]	[1417,1431]	[1409,1435]
P50O9C.5	50	9	0.50	[1624,1646]	[1650,1669]	[1624,1645]
P50O9C1.			1.00	[1626,1649]	[1646,1672]	[1625,1643]
P50O9C2.			2.00	[1622,1649]	[1635,1656]	[1602,1625]
P50O10C.5		10	0.50	[1650,1672]	[1669,1690]	[1646,1666]
P50O10C1.			1.00	[1645,1663]	[1666,1685]	[1651,1671]
P50O10C2.			2.00	[1647,1667]	[1668,1690]	[1638,1656]

Figure 4.8 compares the penalized objective value of the results of IP model and IPETS. The result of the IP model has been evaluated through the simulation model using Expression (1). This figure shows that in most of the test problems, IPETS finds better solutions compared with the initial solutions offered by the IP model.

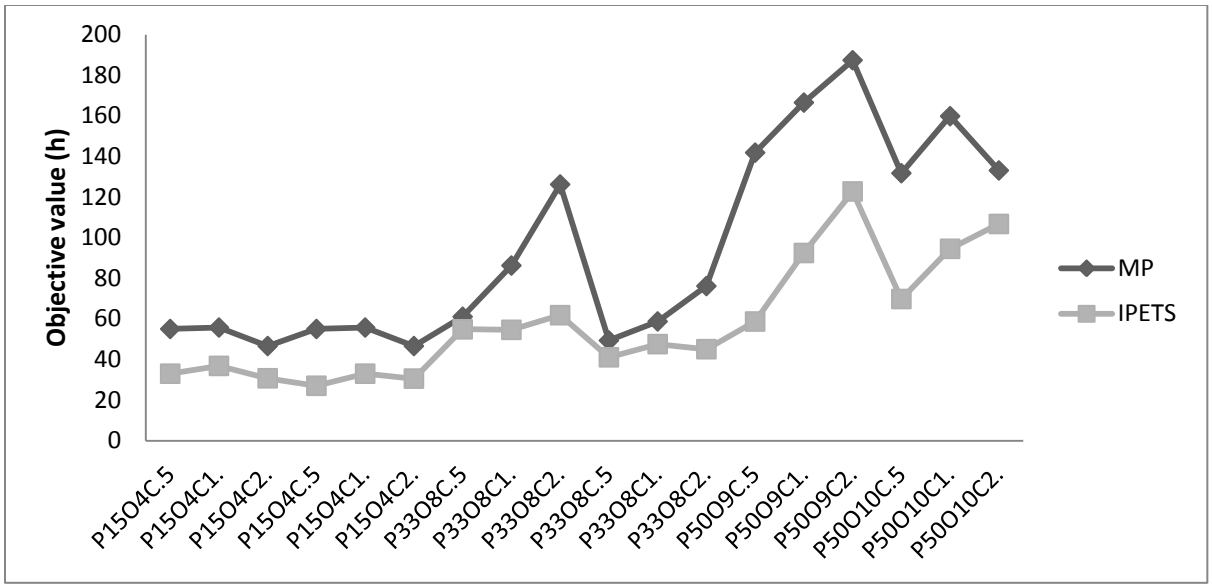

Figure 4.8 A comparison of the result obtained from IP model and IPETS.

In summary, it is observed that using mathematical programming in metaheuristics improves the performance of methods in terms of solution quality. Furthermore, while BPETS may not outperform IPETS in all criteria of solution quality, it yields quality results for large size problems in reasonable computation time.

4.4.2. Scheduling rules

In order to study the performance of the scheduling rules compared with the proposed methods, a test problem with 15 patients is considered. Several scheduling rules are considered along with several hedging factors, α. Scheduling rules include two stages of sequencing patients and determining appointment time. In sequencing step, the proposed method utilizes the mean of service time distributions for scheduling rules (SPT, LPT, DSR, SCV, and SVAR). In addition, the use of adjusted service times in SSPT and SDSR are examined which in total results in seven sequencing rules. Using the increments of 0.05 for the hedging factor (α), 16 values of hedging

factor are determined which ranges from 0 to 0.75 for each scheduling rule. Consequently, 16 schedules for each scheduling rule is delivered.

After the determination of appointment times, the simulation model evaluates the schedule. The method monitors three criteria, namely, average waiting time of patients, completion time, and the number of cancellations. The method reports the average value of these metrics over 10 simulation runs using different random seeds. Each run includes 30 replications of the simulation.

The preliminary experiments show that scheduling rules result in many cancellations, as these approaches do not address the surgeons' time window constraints. A new set of test problems are generated by relaxing the surgeons' schedule in order to evaluate the performance of the scheduling rules. Figure 4.9 presents the performance of scheduling rules on the 15-patient test problem with relaxed surgeon schedule. In this figure, each scheduling rule consists of 16 points corresponding to 16 levels of the hedging factor, α. For each scheduling rule, these points are spread from left (α=0.75) to the right (α=0) in the figure. This figure also includes the results generated by IPETS, BPETS, and STS methods. IPETS, BPETS, and STS are presented each by a single point in the plot because they produce a single solution for the given problem (no hedging factor involved).

Our experiments show that DSR, SDSR, SVAR, and SCV have superior performance among the studied scheduling rules. DSR and SDSR present schedules with short waiting time and long completion time. On the other hand, SCV and SVAR deliver schedules with short completion time and long waiting time values. In terms of cancellations, DSR and SDSR present the lowest number of cancellations. Results suggest that extreme values for the hedging factor might result

in either long completion times or long waiting times. Our experiments indicate that scheduling rules are not a competent approaches when dealing with providers with time window constraint (e.g., surgeons in OR departments). Scheduling rules, however, can be utilized in systems that have relaxed time windows for the service providers.

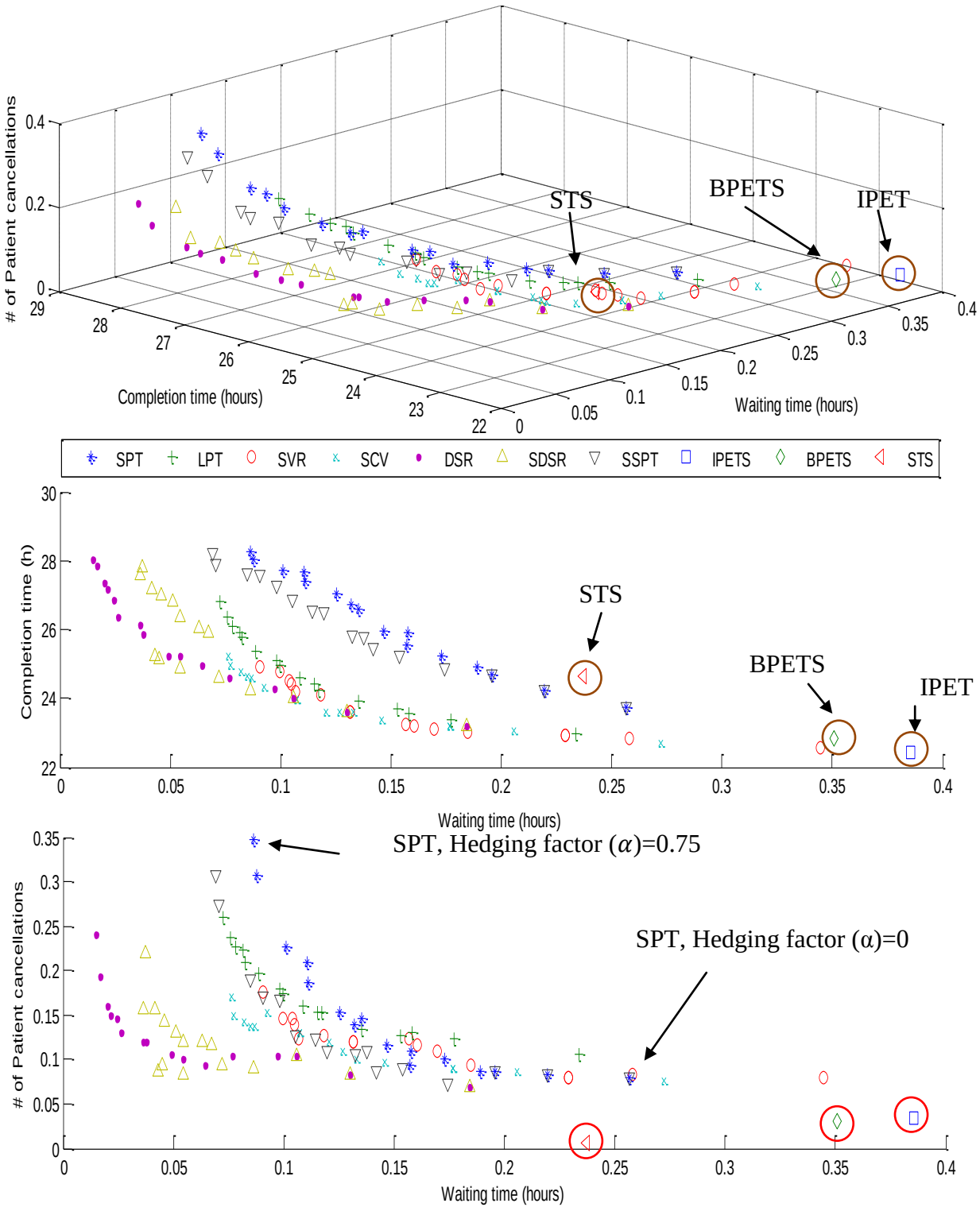

Figure 4.9 Comparison of the performance of multiple scheduling rules with STS, IPETS, and BPETS on a test problem with 15 patients and relaxed surgeon schedule.

Concerning the performance of proposed simulation-based methods, it is observed that STS method presents inferior results in terms of completion time, while delivering schedules with the smallest number of cancellations. Although STS presents promising waiting time and number of cancellations, completion time is significantly larger than those generated by IPETS and BPETS. On the other hand, it is observed that the proposed methods enhanced with MP models deliver significantly better results, especially in terms of completion time. The large completion time for STS makes it an inferior option, as there are scheduling rules (e.g., SVR) leading to comparable waiting time with less completion times. Also, recall that using scheduling rules to schedule the appointment results in many cancellations when the surgeons' time window constraint is considered; Figure 4.9 shows only results when this constraint is lifted. Results of proposed methods are circled.

In summary, it can be concluded that in problems without time constraints, ordinary metaheuristic methods (such as tabu search) may not offer better schedules compared with scheduling rules. This conclusion confirms the insights achieved by Gül et al. [69]. However, using MP models to reinforce the metaheuristics may significantly improve the performance of metaheuristics where availability of surgeons is restricted by time window constraints.

4.5. Case study

In this section, BPETS method is applied, which is found to be promising in previous sections, to a case study OR department. In the previous section, experiments indicated that both IPETS and BPETS present good quality solutions. However, IPETS relies on integer programming and it may present difficulty in solving large problems due to its required large computational efforts.

Our case study is the OR department of a major hospital in Canada, which includes 13 ORs to serve elective and emergency patients. The ORs are used for most procedures and are shared among the specialties. The space of the OR department is separated into three sections including the holding area, ORs, and post anesthesia unit (PACU).

The holding area is used for the preparation of the patients before entry to the OR which includes procedures such as the identification of patients, checking the consent form and lab tests, etc. In addition to a surgeon, usually three nurses along with an anesthetist are required for each surgery. The nurses and anesthetists are assigned to an OR for a complete shift. However, each surgeon is available throughout the whole operation of the patient, which is supposed to fall in the time window of his or her availability to the department. The PACU is the main recovery unit, which can accommodate up to 12 patients. In cases where more beds are needed for recovery, the surgical intensive care unit (SICU) is able to accommodate 10 patients. SICU often serves patients with major surgical procedures. Historical data indicates that on normal days 70%-80% of the cases are elective surgeries, which are assigned to the ORs and resources in the surgery scheduling. The patient processing starts at 7:30 am at the OR department. The PACU unit is open until 12:00 am. However, if a patient is not ready to be discharged, she/he will be sent to SICU.

The activities in a patient's visit are broken down into three stages of pre-operation, operation, and post-operation. Based on the available data, the service time probability density functions of different patient types are developed for each stage. Eleven patient types are considered based on the 11 existing specialties in the OR department. The service time of each patient type at each stage has been determined using available historical data. Table 4.6 presents the service times in this case study for each specialty.

Table 4.6 Processing time of different types of patients in the case study OR department.

Specialty	Distribution	Mean (min)	Standard deviation
Cardiac	LOGN(48.8, 12.4)	48.8	12.4
	391 * BETA(2.94, 4.96)	145.51	63.35
	240 + EXPO(71.5)	311.5	71.5
Vascular	12 + WEIB(31.6, 1.77)	40.12	16.41
	LOGN(98.5, 154)	98.5	154
	LOGN(221, 91.5)	221	91.5
Neurology	TRIA(14.5, 50, 77.5)	47.33	12.89
	490 * BETA(2.16, 6.3)	151.17	69.46
	45 + 270 * BETA(1.76, 4.06)	126.64	47.48
Orthopedic	14.5 + GAMM(17.7, 1.39)	39.10	5.84
	LOGN(132, 200)	132	200
	40 + WEIB(93.2, 1.22)	134.3	71.92
Oncology	6.5 + 65 * BETA(1.98, 3.77)	28.88	11.88
	LOGN(98.1, 201)	98.1	201
	25 + EXPO(73.9)	98.9	73.9
Thoracic	2.5 + 81 * BETA(1.71, 3.73)	27.96	14.81
	LOGN(79.7, 147)	79.7	147
	70 + EXPO(99.7)	169.7	99.7
Urology	12.5 + ERLA(7.93, 2)	28.26	5.63
	EXPO(86.2)	86.2	86.2
	50 + WEIB(88.4, 1.15)	134.13	73.34
Gastro-Intestinal	3.5 + 65 * BETA(3.2, 5.57)	27.21	10.01
	LOGN(111, 163)	111	163
	45 + 505 * BETA(1.72, 8.02)	134.17	58.76
Plastic	1.5 + GAMM(7.67, 3.55)	28.72	9.83
	430 * BETA(0.538, 1.54)	111.32	107.35
	30 + WEIB(89.1, 1.32)	112.04	62.73
Oral/Dental	2.5 + 53 * BETA(2.99, 4.07)	24.94	9.22
	LOGN(81.7, 54.2)	81.7	54.2
	30 + WEIB(55.4, 0.903)	88.18	64.53
Otolaryngology	9.5 + WEIB(19.1, 2.04)	26.42	8.68
	238 * BETA(2.25, 7.59)	54.42	30.35
	TRIA(25, 53.8, 140)	72.93	24.42

The case study OR department is used in order to evaluate the performance of BPETS in comparison with the actual appointment schedules. Experiments include 24 days of studied OR department and apply BPETS method in order to schedule the same set of patients for each day. The performance of BPETS, in terms of completion time, is then compared with the actual completion time of each day extracted from the historical data, and the completion time of the actual schedule of each day reached using the simulation model. Figure 4.10 presents the comparison between BPETS and actual practice in terms of completion time.

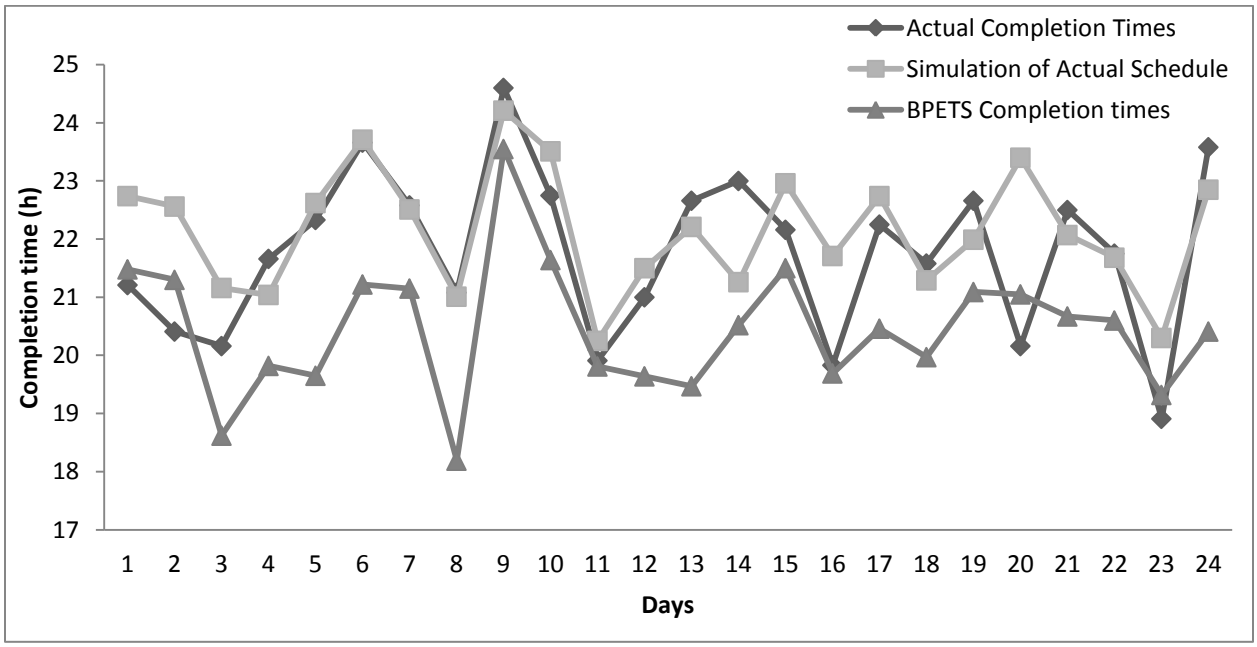

Figure 4.10 Comparison of the completion time of BPETS with the actual schedule in case study OR department.

The comparison of BPETS and simulated actual schedules indicates that the application of BPETS results in a significant reduction of completion time in the studied OR department.

4.6. Conclusion

Our survey on scheduling outpatient surgeries points out the need in existing literature for efficient and effective methods that present competitive optimization schemes while addressing different aspects of complex real world systems. To address this need, three simulation-based tabu search methods have been proposed which benefit from the flexibility of simulation, and the power of mathematical programming optimization to find good quality solutions for outpatient scheduling in OR departments. The proposed simple tabu search (STS) method integrates simulation with optimization. IPETS and BPETS methods improve on STS by incorporating integer and binary programming models, respectively. Three studies have been conducted to examine the performance of proposed methods. First, the performance of the three proposed methods has been compared in terms of waiting time of patients, completion time, number of cancellations, and computation times. Second, the performance of scheduling rules has been studied in appointment scheduling of OR departments and compared them to the proposed simulation-based methods. Finally, this chapter applied the BPETS to a case study OR department. To analyze the performance of the proposed methods, several experiments have been designed over a range of important factors - the number of patients, number of ORs, and coefficient of variability of service times. The range of the factors is determined based on the insights obtained in the case study.

In the performance study of the proposed methods, our observation suggests that although STS, BPETS and IPETS require approximately the same amount of computation time, STS results in consistently inferior solutions as compared to IPETS and BPETS. Thus, STS is not recommended for practical purposes. This further confirms the significant effects of employing mathematical programming in improving performance of metaheuristic approaches.

Comparison of IPETS and BPETS indicates that both methods present quality solutions. However, since IPETS relies on integer programming, it may present difficulties in solving large problems since it requires long computational time. Therefore, applying BPETS is recommended for practical purposes based on its effectiveness and efficiency in solving test problems ranging from small to large.

The experiments indicate that methods based on scheduling rules are not competent approaches when dealing with providers restricted with time-window constraints (e.g., surgeons in OR departments). Application of scheduling rules may result in several cancellations because these rules do not consider the time constraints. However, scheduling rules can be utilized in the systems that include providers with relaxed time-window constraints. DSR, SDSR, SVAR and SCV have superior performance among the studied scheduling rules. DSR and SDSR present schedules with small waiting times and large completion times. This result suggests that the appointment rules that present a dome pattern may result in minimum patients waiting times. On the other hand, SCV and SVAR deliver schedules with short completion time and long waiting time values.

Applying BPETS in the case study suggested that application of metaheuristics enhanced with MP models improve the appointment scheduling of the case study in terms of completion time.

CHAPTER FIVE

5. Appointment scheduling of outpatient clinics with heterogeneous service sequence and multiple objectives

5.1. Introduction

This chapter considers multistage outpatient clinics that serve patients of multiple types with non-identical stochastic service time at each stage, simultaneously considering multiple objectives. It assumes heterogeneous service sequences, i.e. each patient type follows a specific order to visit clinic stages. For instance, a surgical patient may go through stages such as reception, pre-operation, operating room, and post-operation, while a check-up patient may undergo a different sequence of stages. Here, scheduling refers to the determination of the arrival time of each patient to the clinic in order to minimize the waiting time of patients and the overtime of the clinic.

In this chapter, a multiobjective optimization method is developed to provide a Pareto front for the patients' waiting time and the clinic overtime, in contrast to commonly used single objective optimization methods. The Pareto front (also known as Pareto set, Pareto frontier, or set of non-dominated solutions) is the set of choices that outperforms other solutions in the objective space considering all objectives. Solution x_1 is dominated by solution x_2 if for any of objectives

considered in the optimization, x_1 presents worse performance than x_2. A set of solutions that are not dominated by any other solutions is called Pareto front or set of non-dominated solutions.

Our review of the literature relevant to this chapter suggests the following remarks:

1. Several performance criteria such as patient waiting time, clinic overtime, etc. have been studied in the literature, and several studies addressed problems with multiple criteria. However, most of these studies considered a weighted sum of objectives to generate a single function optimization problem. Therefore, a ample room still exists in the literature of methods that provide Pareto fronts for the appointment scheduling problems.
2. To the best of our knowledge, Pham and Klinkert [54] is the only article, which proposed a method that was capable of addressing patients with different service sequences. However, their method only considered the deterministic version of the problem in OR departments. Thus, it is understood that still ample room exists for developing methods that can address multistage facilities serving patients with different service sequences and stochastic service times.

To address these opportunities, this chapter proposes a multiobjective simulation-based tabu search method enhanced by MP model. Our work can be differentiated from the previous works in following ways. First, to the best of our knowledge, this is the first attempt to address appointment scheduling of patients with stochastic service times, and heterogeneous service sequence in a multistage facility in which patients may revisit a stage. Second, our method takes advantage of the flexibility of simulation to model complexities of systems and integrates MP and tabu search to find optimal or near optimal solutions. Third, it offers the Pareto (near) optimal set of schedules, which shows the trade-offs between the factors that influence patients and providers.

This work proposes an optimization method termed multi-agent tabu search (MATS), which simultaneously addresses bi-objectives of minimizing patients' waiting time and the clinic's overtime. MATS uses mathematical programming (MP), tabu search, and simulation modeling. The MP model provides MATS with promising initial solutions. Tabu search then improves the initial solution by searching for optimal schedules by running a number of agents in parallel. The agents seek the non-dominated solutions of the problem, and share information with each other to improve the search performance of the algorithm. In order to capture the complexity of the outpatient clinics, MATS is performed on a discrete event simulation.

In order to evaluate the performance of the proposed method, several test problems are developed with a range of important factors such as the number of patients and patient types, and the coefficient of variance of service time. The performance of MATS is compared with Non-dominated sorting genetic algorithm II (NSGA-II) in terms of quality of solutions and computation time. To measure the quality of solutions, two performance indicators, hypervolume and spacing, are considered which will be explained later in this chapter.

The rest of this chapter is organized as follows: Section two discusses the problem definition. Section three describes the architecture of the proposed approach and the different components of the algorithm. Section four reports the design of experiments (DOE) and analysis of the results. Finally, Section five provides the conclusions.

5.2. Problem description

This chapter addresses clinics in which patients can have different sequences of services. A good example for this kind of clinic is a private clinic that offers a large variety of outpatient services, from checkups to small surgeries. The problem considers the appointment scheduling of patients

of different types with stochastic service times in each stage where each type of patient may not only require different service times, but also has a specific sequence of services (including the possible revisits to a stage). Here, appointment scheduling refers to the determination of the arrival time of each patient at the clinic in order to minimize the average waiting time of all patients, and the overtime of the clinic. The availability of resources, such as doctors and nurses for each stage, are also considered.

This chapter considers a clinic that includes a surgical suite, a diagnostics section, and doctor consultation sections. Figure 5.1 shows the layout of the clinic and the route that a check-up patient and a surgical patient may take. It is possible that some patients need to revisit a stage based on their type. For instance, Figure 5.1 shows that the check-up patient first visits the doctor, goes to X-ray and Lab, and then returns to the doctor stage. For instance, OR stage includes multiple staffed ORs with the service time dependant on the patient's type.

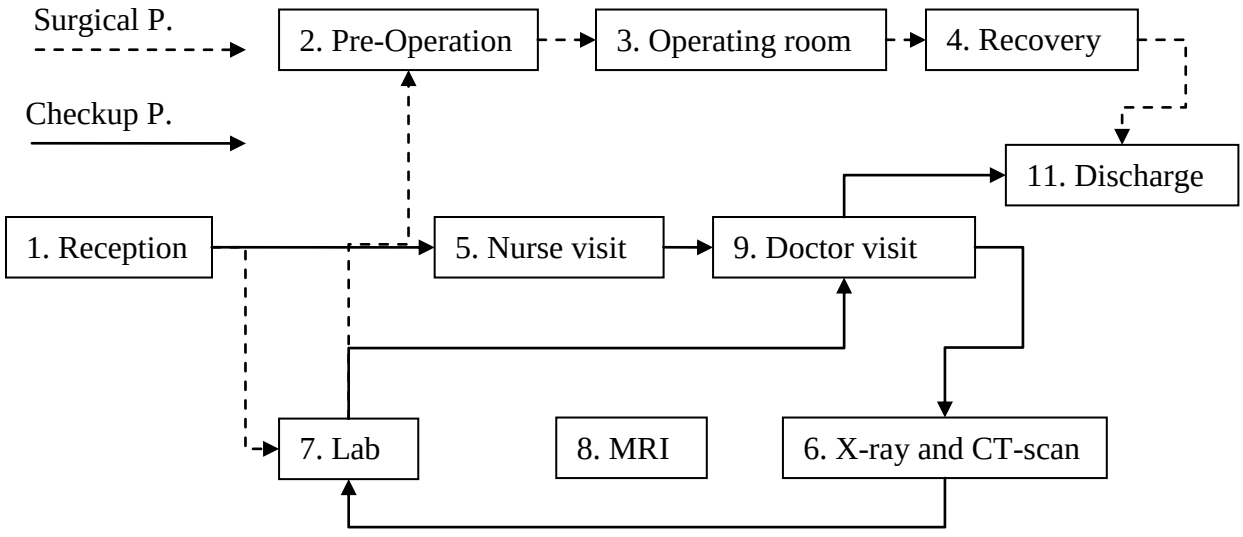

Figure 5.1 Layout of a clinic depicting the service sequences of check-up patients and surgical patients.

A predetermined number of patients of different types are considered to be scheduled in the horizon of one day. Parameters such as, distribution of service time for each patient type, capacity of each stage, and sequence of services are given. The patients arrive according to the schedule, and it is assumed that neither tardiness of arrivals nor no-shows happen in the system. The stages of the clinic (except for the reception stage) work according to the first-come-first-serve rule. The reception stage admits the patients according to the schedule.

5.3. Methodology

MATS employs an MP model, and a simulation model along with the tabu search to generate Pareto (near) optimal schedules that minimize waiting time of patients and overtime of the clinic. The MP model provides MATS with promising initial solutions through solving a deterministic version of the problem. Tabu search then improves the initial solutions by searching for optimal schedules by running a number of agents in parallel. The agents seek the non-dominated solutions of the problem and share information with each other to improve the search performance of the algorithm. Simulation modeling enables MATS to evaluate the schedules by considering stochastic nature of services of the clinic, and applying several resource and operational constraints. Finally, MATS presents a non-dominated set of solutions that contains the most promising appointment schedules.

5.3.1. Mathematical programming model

In the proposed method, MP model provides the MATS with promising initial solutions. The MP model solves the deterministic version of the problem. MATS uses the results of the MP model along with a number of randomly generated solutions to compose the initial Pareto front.

Mathematical model (3):

The notation of the model is presented as follows:

Notation:

t discrete time index, t=1,...,T, where T is the time horizon and number of time grids in each day;

j stage index, j=1, ..., k, k+1, …, M, where M is the number of stages in the department. It is assumed that stage k is the doctor visit stage and the stage (k+1) is dedicated to patients' revisit. Stage M is considered as the discharge stage;

p patient type index, p= 1, ..., P; where P is the number of patient types;

$[j]_{Bp}$ indicates the stage located, before position of stage j in the ordered set of B for patient type p.

Parameters:

$S_{j,p}$ service time of patient type p in stage j;

$I_{j,p}$ the initial number of patients of type p in the line, waiting to be served at stage j;

R_j the number of available servers or operators in stage j at the beginning of the scheduling horizon;

γ_p the penalty coefficient of waiting of a patient of type p for a single time period;

H a large number;

B_p the ordered set of stages that patient type p should follow.

α penalty coefficient for waiting of a patient per time grid;

β penalty coefficient of operating the department per time grid.

Variables:

$x_{j,t,p}$ the number of patients of type p at stage j to start being processed at time t;

$Q_{j,t,p}$ the number of patients type p who are waiting to be served at stage j at time t;

$X_{j,t,p}$ the cumulative number of patients of type p at stage j has been started being processed by time t;

$r_{j,t}$ the number of available idle resources at stage j and time t; each stage has its dedicated resources;

m the last time block, in which all patients have been discharged (makespan of the schedule);

y_t a binary (1 or 0) variable which indicates if there is any discharge at time t.

The MP model is expressed as follows:

- Objective functions:

$$\text{Minimizing } \alpha \sum_{j} \sum_{t} \sum_{p} \gamma_p . Q_{j,t,p} + \beta m \tag{5-1}$$

- Queue balance constraints:

$$Q_{j,t,p} = I_{j,p} - X_{j,t,p} + X_{[j]_{Bp},t-s_{[j-1],p},p} \quad \forall j \in B_p, t, p. \tag{5-2}$$

- Cumulative variables:

$$X_{j,t,p} = \sum_{\tau=1}^{t} x_{j,\tau,p} \qquad \forall j \in B_p, t, p. \tag{5-3}$$

- Capacity constraints:

$$r_{j,t} = R_j - \sum_{(p|j\in B_p)} X_{j,t,p} + \sum_{(p|j\in B_p)} X_{j,t-s_{j,p},p} \quad \forall j \notin \{k, k+1\}, t, \tag{5-4}$$

$$r_{k,t} = R_k - \sum_{p} X_{k,t,p} + \sum_{p} X_{k,t-s_{k,p},p} - \sum_{p} X_{k+1,t,p} + \sum_{p} X_{k+1,t-s_{k+1,p},p} \qquad \forall t. \tag{5-5}$$

- Number of patients whom have to be served:

$$X_{M,T-s_{M,p},p} = \sum_{j} I_{j,p} \qquad \forall p. \tag{5-6}$$

- The makespan indicator constraint:

$$y_t \cdot H \geq \sum_{p}\sum_{j} X_{j,t,p} \qquad \forall t, \tag{5-7}$$

$$m \geq t.y_t \qquad \forall t. \tag{5-8}$$

$x_{j,t,p}$, $Q_{j,t,p}$, $X_{j,t,p}$, $r_{j,t}$ $\forall j,t,p$ are non-negative integer variables; y_t is a binary variable for t.

- Initial conditions:

The values for $x_{j,t,p}$, $I_{j,p}$, $X_{j,t,p}$, $r_{j,t}$ should be pre-specified.

MP model (3) is differentiated from the models presented in the previous chapters by its capability to manage service sequence of each type of patients. That is, the model determines which stages the patients visit, according to the patients' type.

To one's knowledge, Pham and Klinkert [54] is the only article that provided an MP model capable of addressing scheduling of patients with heterogeneous service sequence in a multistage surgery department. While they only considered the deterministic version of the problem, they

focused on minimizing the makespan and scheduling of the patients as soon as possible. Our experiments suggest that this policy may cause significant patient waiting time. Their experiments included up to 26 patients served by six ORs.

Our proposed model is able to represent the flow of patients in the stages of the clinic as well as revisited stages. In addition, the proposed method considers the minimization of bi-objectives of patients' waiting time and the overtime of clinic while considering the resources' availability. Our experiments show that our model can solve large problems in a reasonable amount of time.

5.3.2. Simulation model

Using the Arena™ 12 software, a discrete-event simulation model of the described clinic has been developed. The model includes the eleven stages as shown in Figure 5.1. The commonly used lognormal distribution is adopted to model the service time distribution for each stage (e.g., see [94], [98]). The mean and variance of the service time is determined based on the specified values in each test problem, to be discussed in Section four. In addition, the number of resources at each stage is determined according to the test problem parameters, such as the number of patients, the number of ORs, and so on. For OR stages, it is assumed that the number of surgeons is equal to the number of ORs, and that the ORs are staffed.

The simulation model falls in the category of terminating simulation models [88]. That is, in our simulation model, a predetermined number of patients are served in the scheduling horizon of one day. The patients arrive according to the schedule with no tardiness in arrivals and no no-shows in the system. Patients proceed through the clinic according to their service sequence. The simulation model has been validated by comparing the waiting time and overtime measures with

those from MP when the system is assumed stationary (i.e., the variance of service time is equal to zero).

The simulation model has been used to estimate patients' waiting time and the clinic's overtime for each proposed schedule. Also, the simulation model is used as a platform to compare the performance of MATS with NSGA-II.

5.3.3. Multi agent tabu search (MATS)

Most multiobjective tabu search (MOTS) methods consider multiple solutions that are simultaneously improved towards the Pareto optimal front. Typically, MOTS methods apply two approaches regarding the objective function handling. The first approach considers a weighted sum of the objectives as the objective function. Subsequently, any single objective optimization method can be used to solve the problem. The weights are usually pre-set, and the result of this method is a single solution rather than a Pareto front. The second approach deals directly with finding the Pareto frontier solutions. For instance, Hansen [99] used a modification of weighted sum method in which multiple weighted sum objectives were improved simultaneously considering different sets of weights for objectives. Caballero et al. [100] developed a two-phase TS based algorithm: the first phase involved a tabu search method to generate a non-dominated solution set; the second phase consisted of an intensification method using path-relinking strategies. Jaeggi [101] developed a novel TS method using Hooke-Jeeves direct search methods. They had suggested that NSGA-II might be a better approach for problems with a small number of variables. However, they added that MOTS was a better approach for large and highly constrained problems.

This work proposes a multi-criteria multi agent tabu search algorithm to obtain the Pareto (near) optimal frontier for the appointment scheduling problem. MATS algorithm includes a number of agents that attempt to find members of the non-dominated solution set (also, called approximation set). Here, a stochastic simulation model is incorporated with the tabu search to solve a highly constrained problem (including time and resource compatibility constraints). The time and resource constraints are captured using simulation and MP models. For instance, availability of resources in each stage is managed by variables local to the simulation model. However, this policy requires passing of many parameters to the model. The simulation model, for example, requires the initial values of resources that are provided by MATS algorithm. In addition, a mathematical programming model is developed to provide the initial solutions to the tabu search method. In order to reduce the number of function evaluations, a deterministic scheduling module (DSM) is developed which estimates the waiting time and overtime of a schedule based on the mean of service time. Then a number of schedules with the largest fitness value are selected to be evaluated by the simulation model.

Moreover, agents work in parallel and share information with other agents regularly to improve the algorithm's performance. Each agent functions as a standalone tabu search that seeks optimal solutions for the problem in each iteration. MATS updates the information among the agents after every iteration. Each iteration of MATS includes only a single iteration of an agent.

A solution consists of two parts. The first part is an array of size *n,* where *n* is the number of patients. This array represents the time block at which each patient arrives at the first stage. The second part is of size *T,* where *T* is the number of time blocks. The second part of a solution describes the number of patients who will be served in each time block. Figure 5.2 depicts the steps of MATS, which are described below.

5.3.3.1. Generating initial solutions using the MP model

In order to initialize the algorithm, each agent requires an initial solution. A number of initial solutions are determined using MP as outlined in Section 4.2. The rest of initial solutions are specified randomly by assigning an arbitrary time block to each patient.

5.3.3.2. Fitness computation and frontier points identification

MATS determines the fitness value (G score) of a given set of solutions following the Schaumann [102] approach. Assuming minimization of all objectives:

$$G_i = [1 - \max_{1 \le j \le J, j \ne i}(\min_{1 \le k \le m}(f_{s1}^i - f_{s1}^j, \dots, f_{sk}^i - f_{sk}^j, \dots, f_{sm}^i - f_{sm}^j))]^l \qquad (5\text{-}9)$$

Where, G_i is the fitness value of solution *i; J* is the number of objectives and *m* is the number of solutions in the set; *i* and *j* are two solutions in a given set of solutions. f_{sk}^i is the scaled k^{th} objective value of the i^{th} solution. The max operator is over all solutions except solution *i*. The min operator includes all the objectives. The f_{sk}^i is scaled in range [0,1], hence all objective values are in the same range and comparable. The frontier exponent *l* equals to 1. f_{sk}^i is scaled using Equation (5-10).

$$f_{sk}^i = \frac{f_k^i - f_k^{min}}{f_k^{max} - f_k^{min}} \qquad (5\text{-}10)$$

where, f_k^i is the original value of the k^{th} objective and $f_k^{\max}$ and $f_k^{\min}$ are the maximum and minimum values among original values of the k^{th} objective of all solutions, respectively. The non-dominated solution set is determined based on the G score of all solutions. The solutions with the score greater than or equal to one are identified as non-dominated solutions.

For instance, consider five solutions with 2 objectives of f_1 and f_2: a(0.5,2), b(1,1), c(4,0.5), d(3,1.5), and e(2,3). The scaled objectives of these points over the first objective are as follows: $f_{sa}^1 = 0$, $f_{sb}^1 = 0.1428$, $f_{sc}^1 = 1$, $f_{sd}^1 = 0.7142$, $f_{se}^1 = 0.4285$ $f_{sa}^2 = 0.6$, $f_{sb}^2 = 0.2$, $f_{sc}^2 = 0$, $f_{sd}^2 = 0.4$, $f_{se}^2 = 1$. For sake of brevity, only the calculation relevant to point a is presented. The term $\max_{j=1, j\neq i}^{J} (\min_{k=1}^{m} (f_{s1}^i - f_{s1}^j, ..., f_{sk}^i - f_{sk}^j, ..., f_{sm}^i - f_{sm}^j))$ for $i=a$ is equal to following term by replacing the values: $\max(\min(-0.1428, 0.4), \min(-1, 0.6), \min(-0.7142, 0.2), \min(-0.4285, -0.4))$ which ultimately results in max (-0.1428, -1, -0.7142, -0.4285) and hence the G_a will be equal to *1.1428.*

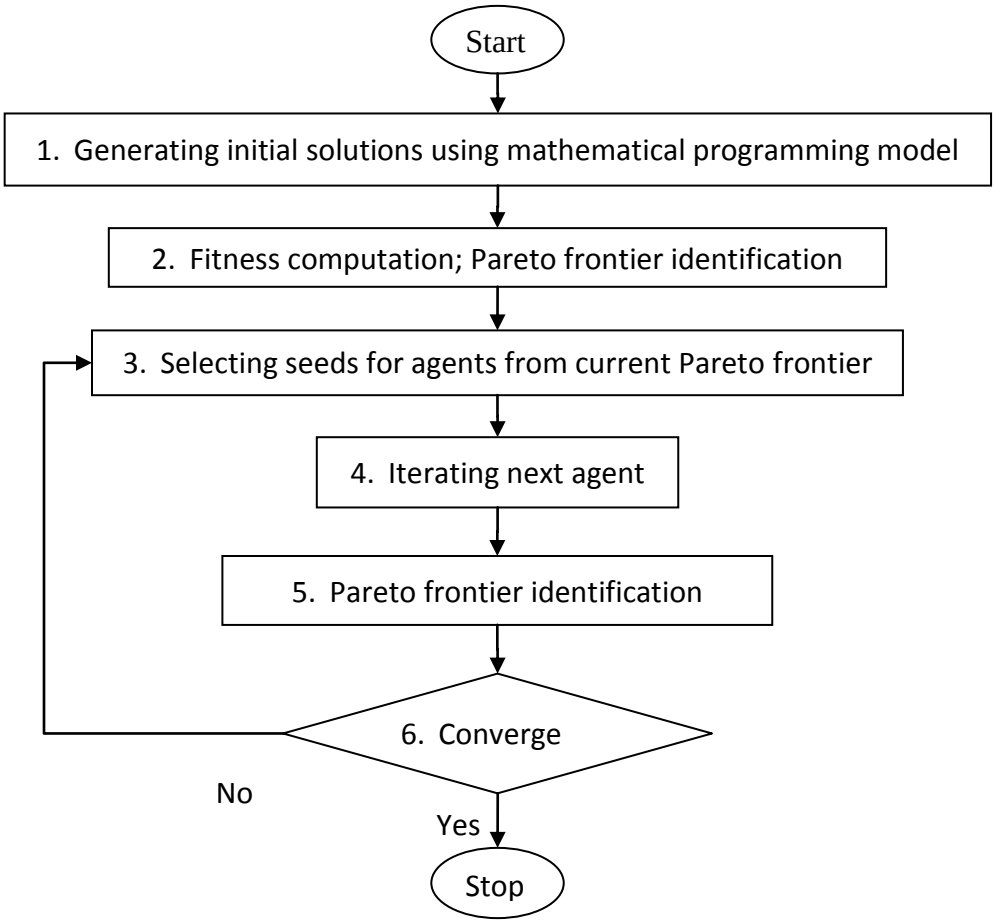

Figure 5.2 Steps of MATS.

5.3.3.3. Selecting seeds for agents from the non-dominated solutions set

In order to achieve the best performance of the algorithm, the agents' seeds are updated every time the Pareto front changes. It is assumed that the better are the agents' seeds, the better results can be expected from the optimization. MATS considers the last iterated agent and updates the next agents' seeds with the solutions that have the highest fitness scores (G_i). This treatment may cause the search to be trapped in a local minimum. To remedy this issue, the crowding distance density estimate by Deb et al.[103] has been used to improve the diversity of agents' seeds. In

order to assign the agents' seed, MATS selects the non-dominated solutions with the highest value of the crowding distance estimator. This estimator has been discussed in Section 5.3.3.5.

5.3.3.4. Iterating next agent

This step includes the iteration of the next agent which encompasses neighborhood generation, fitness evaluation, and selecting the next seed considering the tabu list. This process is described in detail in Section 5.3.4.

5.3.3.5. Pareto frontier identification

This step includes the identification of Pareto front based on the new G scores, which are obtained by including the selected promising solutions from the agents' iteration in the previous step. When an agent iterates, the G scores of the promising solutions within the neighbourhood of the agent are evaluated and compared with the best solutions known in previous iterations. If any of the solutions within the neighbourhood has a fitness value greater than a specific value, $G_0 \in [0,1]$, it is added to the long term memory. The solutions with the largest G score are identified as non-dominated solutions.

5.3.3.6. Convergence condition check

Several convergence conditions can be applied to the proposed method. MATS considers two criteria for algorithm termination, a) the number of stalled iterations, and b) the number of function evaluations. If the Pareto front does not improve during the previous iteration and no better solutions added to it and the set of non-dominated solutions has not been changed, the current iteration is considered as a stalled iteration. MATS converges if a specified number of stalled iterations happen consecutively. In addition, MATS terminates after performing a specified number of function evaluations (solutions evaluated by simulation).

In order to improve the performance, tabu search keeps record of local information by means of different types of memory structures. This local information may include parts of the solutions, or other attributes of the solutions. A short-term memory is used to address the tabu lists of local searches of the algorithm. After each move, the attribute of the move is recorded to avoid cycling in the algorithm. In the proposed method, each local search has a tabu list and tabu tenure of 30 iterations is considered. Based on the swap and insertion, the tabu lists are constructed, which are shared by all agents. The lists contain the history of recent moves. For example, when swapping is applied, a list of moves is recorded. This list prevents any reverse moves.

The long term memory is used to record all best solutions achieved in all iterations. MATS records all the solutions with G score greater than G_0. The long term memory includes all non-dominated solutions at each iteration of the algorithm. In addition, the recorded solutions are used in the calculation of G score for future solutions. The G score of each solution is updated in every iteration.

As an agent performs the search, a number of best solutions are evaluated using simulation. These solutions are then evaluated and ranked based on their G scores. At the end of each iteration, the seeds of agents are updated and replaced with the best solutions recorded in the long term memory. The selection procedure includes two steps: first, all the non-dominated solutions are selected; second, the solutions with the largest crowding distance are selected as new seeds for the agents. This policy intensifies the search in the most promising regions. Contrary to the other MOTS methods which separately improve multiple solutions in parallel, our method uses all the information obtained collectively by all agents. Moreover, each agent uses the information gathered by all agents.

The long term memory is usually used in diversification policies of TS methods to lead the search into the unexplored regions of search space. In single objective optimization problems, a common strategy is penalizing objective function based on the frequency of occurrence of attributes. This work uses the crowding distance density estimator introduced by Deb et al. [103] to maintain diversity of our search method. This method estimates the density of the solutions around a specific point by taking the average distance between this point and its two neighbouring points on each side in the performance space. The distance quantity represents the size of the largest cuboid that encompasses the point excluding any other point from the Pareto front. Figure 5.3 shows the cuboid that is used to determine the crowding distance for point *i* in its front. F_1 and F_2 represent the first and second objective. In addition to guiding the search to unexplored regions, the crowding distance helps increase the range of non-dominated solution set by assigning large values to extreme solutions for each objective function.

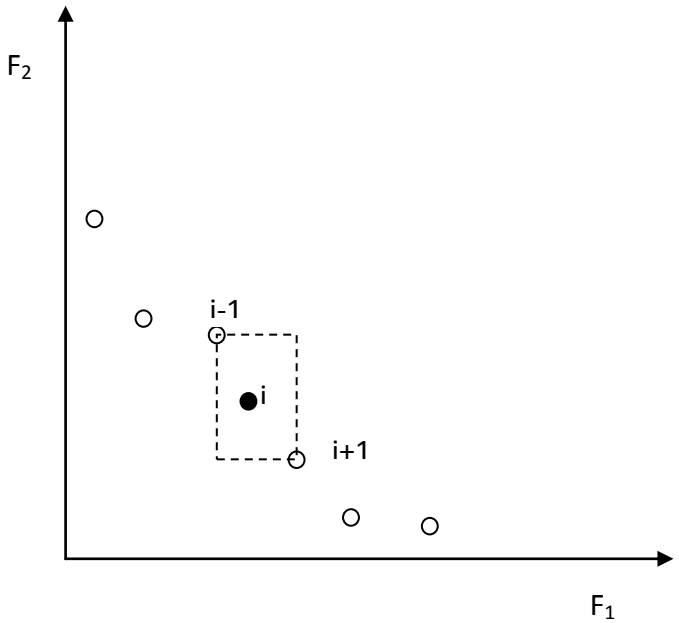

Figure 5.3 The crowding distance of point *i* is the side length of the cuboid surrounding the point.

MATS applies a hash function in order to speed up the search and identify the solutions. Hash functions commonly refer to any algorithm or mathematical function that converts a large amount of data into an integer value. Hash functions are usually used to speed up table lookups or to find duplicated records.

MATS uses the hash function to map an integer value to each solution which enables us to speed up search in the long term memory. Furthermore, it prevents the algorithm from recording duplicate solutions in the long term memory and agents' seed. In MATS, a 32-bit Cyclic Redundancy Check (CRC32) algorithm has been used as the hash function. The Reader can refer to Stigge et al. [104] for more information on its theory and implementation.

5.3.4. Tabu search agent

This section describes Step 4 of the MATS algorithm. This step includes the steps of an agent's iteration. In the proposed algorithm, agents operate according to the tabu search algorithm. Generally, a tabu search iteratively generates the next solution *j* from the current solution *i* through specified steps. A neighbourhood is defined for each current solution, *N(i)*. The next solution is obtained by searching around *N(i)* using neighbourhood search methods.

Tabu search allows for non-improving moves. That is, even if the best solution found in the current neighbourhood is a non-improving one compared to the best-known solution, the non-improving solution will be used in the next iteration. Here, a move is defined as replacing the seed solution of the tabu search with a new solution. The components of the tabu search are described in the following sub sections.

5.3.4.1. Local searches:

Each local search is applied to a specific part of the solution. The same solution structure has been used throughout this thesis. Readers can refer to Section 3.3.4.2 for further description.

Swapping: let *i* and *j* be two positions in a random sequence *s*. By performing a swap-move iteratively, a neighbourhood of *s* is obtained by interchanging the patients in positions *i* and *j*. In the proposed method, swapping is applied to the second part of the solution, which concerns time blocks.

Insertion: let *i* and *j* be two positions in a random sequence *s*. A neighbourhood of *s* is obtained by inserting the patients assigned to position *i* to position *j*, pushing the cells between these positions backward (forward), including the patients of position *j*, if *j* is greater (less) than *i*. This change of positions is performed on the first part of a solution (patients' time blocks). This policy of swap on the second part of the solution and insert on the first part reduces the chance of cycling and trapping in local minima since it separates the domain of the local searches.

5.3.4.2. Deterministic scheduling module (DSM):

In order to reduce the number of function evaluations (evaluating solutions by the simulation model), a deterministic heuristic has been developed. DSM calculates the average waiting time of patients and overtime of each schedule based on the mean of service times. This component enables us to screen solutions before being evaluated by the simulation model. DSM calculates the discharge time of each patient based on the patient service sequence and mean of service times. The completion time of the schedule is the maximum discharge time of all patients. The overtime of the clinic is determined as the difference between the scheduled completion time and the official clinic closing time. The waiting time of each patient is determined under the

assumption that the waiting time is the total time that a patient spends in the clinic subtracted by the sum of service times at different stages. DSM ranks the evaluated solutions according the G score calculated based on deterministic evaluations. A number of solutions with the largest G score will be evaluated using the simulation model. Figure 5.4 illustrates the flowchart of an agent's iteration, which shows details of Step 4 of MATS algorithm.

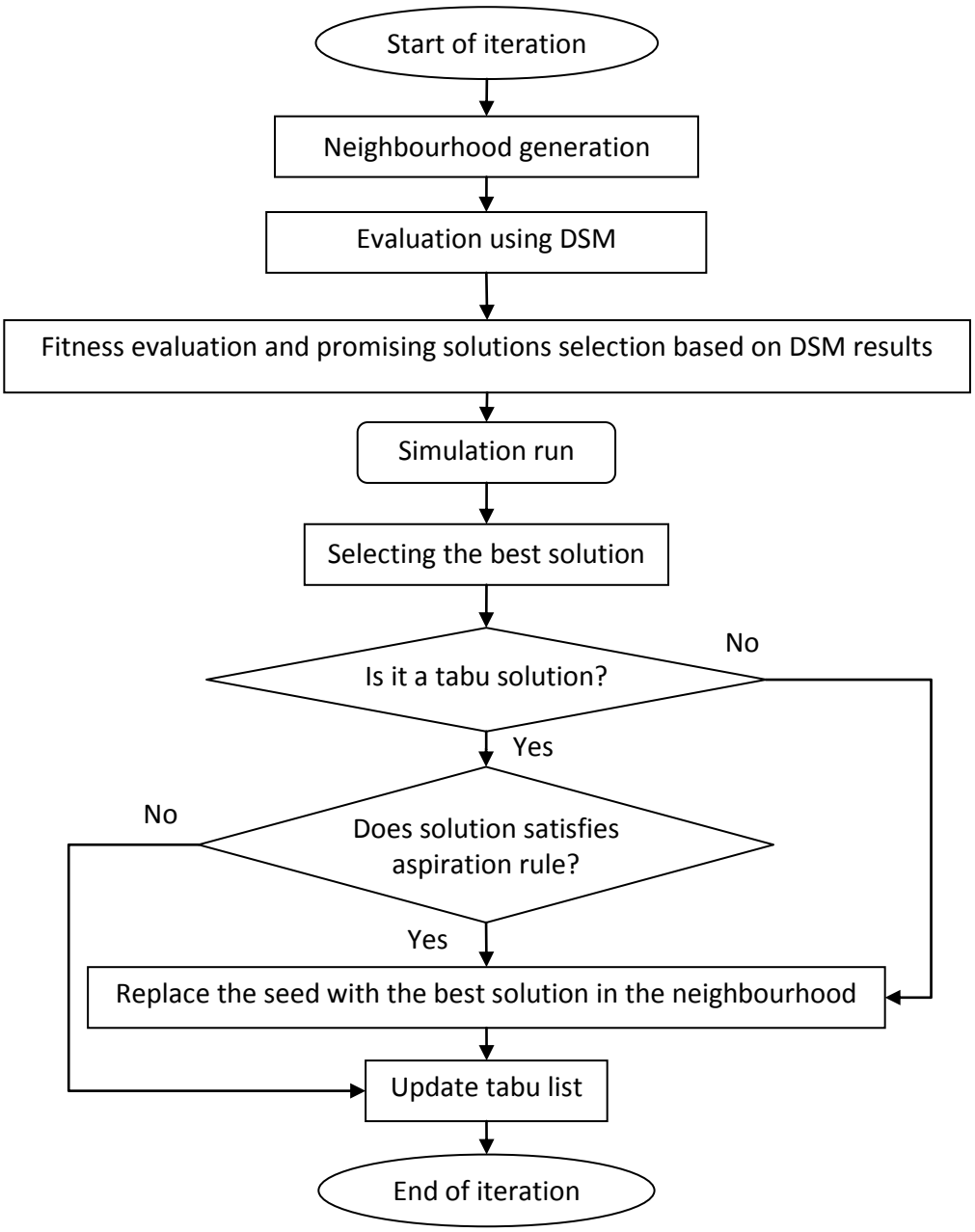

Figure 5.4 Flowchart of an agent's iteration. This flowchart depicts the Step 4 of MATS depicted in Figure 5.2.

The neighbourhood is built based on the current solution using the swap and insertion local search. A random number of solutions in each neighbourhood are evaluated by using the DSM. The solutions are sorted and ranked based on the values assigned to them through the

deterministic evaluation. The agent algorithm selects the promising solution based on the G score of all solutions calculated using the approximate average waiting time and overtime obtained from DSM. The selected promising solutions are then evaluated using the simulation model. The simulation model determines the average waiting time and overtime through multiple replications of the simulation. Thirty replications of simulation are run for each setting in our experiments.

The next step includes the fitness evaluation of the selected solutions with respect to all solutions with high fitness values obtained in previous iterations. In order to do so, the set of new solutions have to be added to and evaluated against the list of solutions with the highest fitness values (non-dominated solution set). The G score of all of the solutions are evaluated based on this combined set.

5.4. Performance study of MATS

5.4.1 Performance measures

To gauge the performance of the proposed MATS for multiobjective optimization, this chapter evaluates the algorithm with the well-known NSGA-II method based on a number of performance measures.

Typically, to study the performance of a multiobjective optimization algorithm, two aspects are examined: effectiveness (the quality of outcome) and efficiency (the required amount of computational resources to generate such an outcome). For effectiveness measurement, various performance measures have been defined in the literature. Addressing the efficiency of an algorithm often includes considering a fixed number of function evaluations or a certain amount of computational time. Interested readers may refer to Zitzler et al. [105] for a review of

performance metrics in multiobjective optimization. Here, three criteria for evaluating performances have been selected: the hypervolume indicator, spacing indicator, and computation time. This section uses the hypervolume indicator to measure the closeness of the non-dominated solutions set to the true Pareto-optimal front. Additionally, in the absence of true Pareto optimal front, it enables us to compare the performance of a number of given Pareto fronts. The spacing indicator has also been utilized to examine the spread of non-dominated solutions. Furthermore, to evaluate the efficiency of the algorithm, this chapter considers a limited number of function evaluations (500 function evaluations), and compares the computation time of algorithms.

5.4.1.1. Hypervolume indicator (HV)

One of the metrics that is employed by many researchers is *hypervolume*, first proposed by Zitzler and Thiele [106]. Hypervolume delivers a single scalar for the closeness of the non-dominated solutions to the true Pareto optimal front. In addition, it can be used to compare algorithms in problems for which the true Pareto optimal front is unknown.

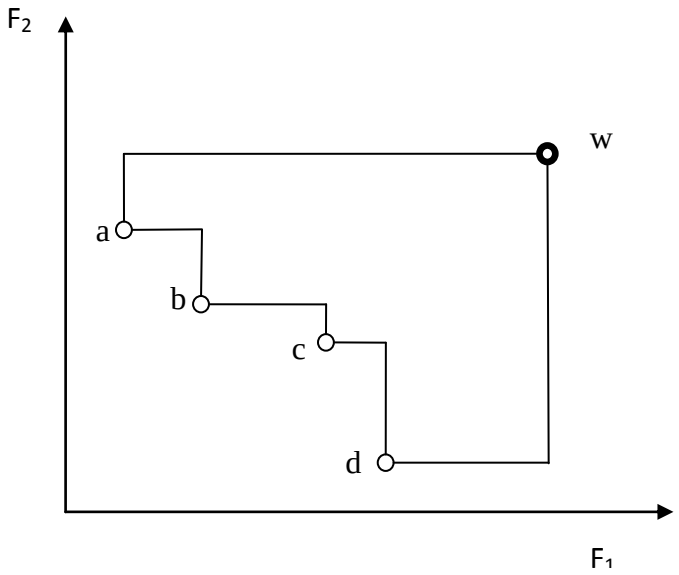

Figure 5.5 Illustration of hypercube indicator

For example, the HV indicator calculates the volume confined by the non-dominated points and a reference point (w) as shown by the shaded area in Figure 5.5. The reference point (w) can be simply determined as a vector of worst possible objective values. The method that results in larger HV indicator values is more desirable. Figure 5.5 presents the calculation of the HV indicator for a bi-objective problem where the objectives are to be minimized.

5.4.1.2. Spacing indicator

The spacing indicator, proposed by Schott [107], measures how evenly the points in non-dominated solutions set are distributed in the objective space. The indicator is the standard deviation of the distance of each point to its closest neighbor, Equation (5-11).

$$\sigma = \sqrt{\frac{\sum_i (d_i - \bar{d})^2}{(n-1)}} \tag{5-11}$$

Where, $d_i = \min_j\left(\left|f_1^i - f_1^j\right| + \left|f_2^i - f_2^j\right|\right), i, j = 1, \ldots, n, \bar{d}$ is the mean of all d_i, and n is the number of solutions in the non-dominated solution set. The less the value of the spacing indicator, the smoother the points are distributed within the objective space. The smooth spread of points in Pareto front is preferred, because it suggests that the method is capable of finding non-dominated solutions in all areas of objective space.

5.4.2. Tests and results

In this work, GAMS™ is used to develop the mathematical programming model and Arena™ 12 for simulation. The CPLEX solver has been used to solve the MP model. The multiobjective tabu search has been coded in Visual C# 2005. Arena object model component is used to establish connection between MATS and simulation model. A PC with 2.53 GHz Intel® Core 2 Duo CPU with 3 GB of RAM has been used to run the experiments. Source code structure and components

involved in MATS implementation are provided in Appendix. MATS parameters have been tuned through extensive experiments to ensure its optimal performance. The parameters include the number of initial solutions, size of candidate solution, tabu tenure, etc.

In order to evaluate the performance of the proposed method, this chapter studies several important factors based on our preliminary analysis. These factors include the number of patients, number of patient types, and coefficient of variation of service time. A set of test problems has been defined with 10, 20, and 40 patients. Up to ten types of patients with different stochastic service times have been defined in the test problems. In the test problems, 4, 6 and 10 patient types are used to examine the effect of this factor on the performance of the algorithm. All patient type levels start with 4 types of patients (types 1-4), which require less numbers of stages, and then we increase the number of patient types to 6 (types 1-6) to include patients who needs surgical procedures. Finally, 10 patient types have been considered where patient type 10 needs revisiting of the doctor stage. Furthermore, each patient type follows a different service sequence in the clinic.

Table 5.1 Specification of patient types.

Patient type	Service sequence	Mean service time of stages (minutes)	Description
1	1,5,9	15,15,105	Patients who need a doctor visit
2	1,7	15,45	Patients who need to do a lab test
3	1,8	15,105	Patients who need to do a MRI
4	1,6	15,120	Patients who need to do a X-ray/CT-scan
5	1,2,3,4	15,105,60,120	Patients with surgical procedures
6	1,7,2,3,4	15,75,15,120,45	Patients who need to do a lab before surgical procedures
7	1,5,9,7	15,75,120,90	Patients who need to do a lab after doctor visit
8	1,5,9,8	15,15,30,120	Patients who need to do a MRI after doctor visit
9	1,5,9,6	15,30,75,105	Patients who need to do a X-ray/CT-scan after doctor visit

10	1,5,9,6,10	15,105,45,90,105	Patients who need to do a X-ray/CT-scan after doctor visit and return for consulting the results with doctor

Table 5.1 shows the specification of patient types. The mean of service time at each stage follows the uniform distribution between 15 and 120 minutes (Uniform (15,120), for each patient type in each stage. The values are then rounded to the closest multiplier of the length of a time block of 15 minutes. Moreover, to study the effect of variability on the performance of the algorithm, the lognormal distribution as well as two different levels of coefficient of variations (CV) of 0.1 and 0.4 have been considered for service time. The lognormal distribution was used since it was commonly used in the literature to represent duration of services in the healthcare (e.g., see[94], [98]). The mean of the distributions was assumed based on patient types.

The algorithm chosen for studying the performance of the proposed method is the non-dominated sorting genetic algorithm (NSGA-II), which is a widely used multiobjective evolutionary algorithm and publicly available ([103]). NSGA-II classifies the individuals into several layers by applying a non-dominated sorting method and a crowding distance operator. It incorporates elitism selection strategies. The source code of NSGA II for global optimization is available. Additionally, Matlab™ offers a toolbox to develop genetic algorithm including NSGAII for different problems. There is no comparable instance of NSGA II readily available in the literature for the current problem. In order to be able to compare NSGA with TS, the implementation of NSGA-II in Matlab™ optimization toolbox is used. NSGA-II was run with a population size of 50 and 500 function evaluations. For the rest of the parameters, default settings are used. The crossover rate of 0.8 (the single point crossover has been adopted), the mutation rate of 0.01 (order changing mutation has been adopted), and tournament selection have been used in experiments.

Table 5.2 presents the result of experiments on the test problems, and compares MATS with NSGA-II based on the three criteria of HV, spacing indicator, and computation time in seconds. The columns one, two, and three indicate the number of patients, the number of patient types, and CV of service times, respectively. The results are based on the average value of indicators over 30 runs for each method.

Table 5.2 Comparison of MTAS and NSGA-II in terms of quality and computational time.

Test problem	# of patients	# of patient types	CV	NSGA-II			MATS		
				HV	Spread	Time (s)	HV	Spread	Time (s)
P10T4C0.1	10 Patients	4 types	0.10	79.49	0.25	353.23	82.20	0.09	196.05
P10T4C0.4			0.40	78.23	0.21	370.98	80.51	0.14	192.66
P10T6C0.1		6 types	0.10	80.39	0.16	507.21	89.48	0.14	182.50
P10T6C0.4			0.40	77.05	0.24	395.63	85.27	0.16	179.95
P10T10C0.1		10 types	0.10	67.83	0.33	366.36	74.65	0.26	224.30
P10T10C0.4			0.40	63.33	0.21	386.05	68.23	0.17	206.40
P20T4C0.1	20 Patients	4 types	0.10	73.23	0.13	383.14	87.05	0.08	203.80
P20T4C0.4			0.40	71.04	0.15	357.47	84.43	0.13	204.22
P20T6C0.1		6 types	0.10	66.53	0.25	396.28	77.65	0.12	225.53
P20T6C0.4			0.40	61.88	0.19	398.58	70.92	0.13	209.39
P20T10C0.1		10 types	0.10	60.07	0.30	488.41	72.45	0.21	236.63
P20T10C0.4			0.40	54.92	0.18	401.63	62.84	0.12	236.50
P40T4C0.1	40 Patients	4 types	0.10	64.27	0.11	397.23	78.32	0.07	232.30
P40T4C0.4			0.40	62.38	0.12	451.53	74.14	0.08	232.50
P40T6C0.1		6 types	0.10	58.49	0.11	429.84	71.57	0.07	230.81
P40T6C0.4			0.40	55.74	0.11	410.19	66.74	0.09	199.83
P40T10C0.1		10 types	0.10	52.58	0.15	396.30	66.71	0.13	240.30
P40T10C0.4			0.40	48.70	0.13	417.53	58.90	0.08	258.68

Studying the effectiveness of algorithms, the results in Table 5.2 show that MATS yields greater values of the HV indicator. It suggests that solutions offered by MATS are closer to the Pareto-optimal front than NSGA-II. The spacing indicator results suggest that MATS presents more

evenly distributed non-dominated solutions than NSGA-II as well. Considering the efficiency of the algorithms, MATS needs less time to complete 500 function evaluations. In summary, experiments suggest that MATS presents better results than NSGA-II in terms of both efficiency and effectiveness.

Figure 5.6 shows the non-dominated solution set of MATS and NSGA-II for the test problem P40T6C0.4 with 40 patients, six patient types, and CV of 0.4. The figure shows the non-dominated solution sets for 30 runs of MATS and NSGA-II. It suggests that MATS and NSGA-II can achieve almost the same results in the solutions with high overtime. However, in the region which addresses lower overtime with greater waiting times, there is a significant difference between the results of each method. The area is shown by an oval. The results suggest that MATS outperforms the NSGA-II in terms of effectiveness and efficiency.

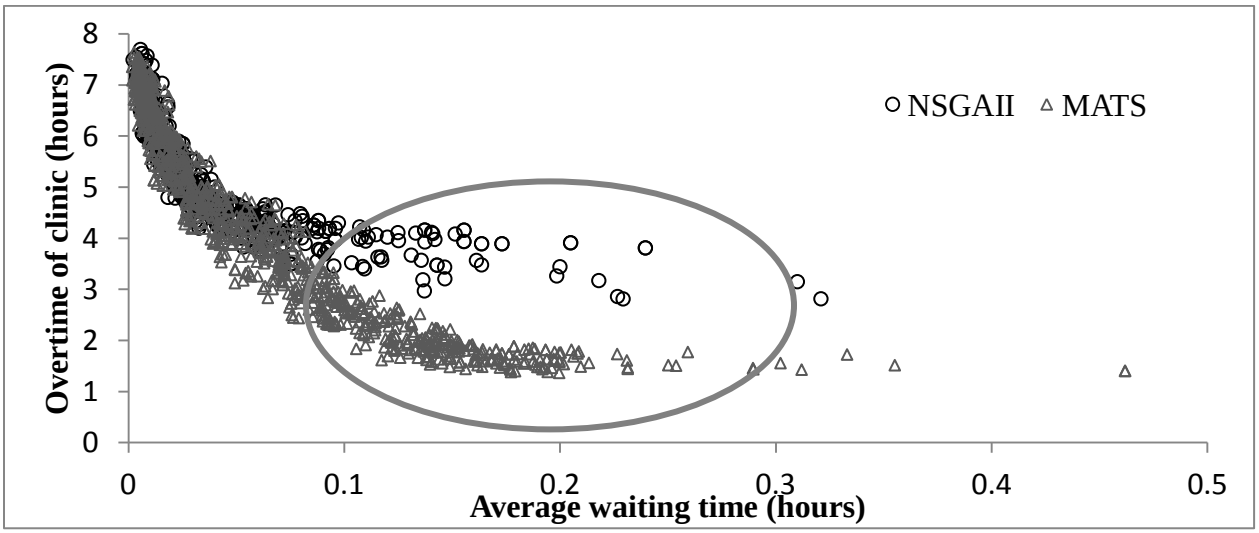

Figure 5.6 Comparison of Pareto sets of MATS and NSGA-II for 30 runs.

5.5. Conclusions

This chapter addresses the appointment scheduling of patients of different types with stochastic service times and heterogeneous service sequences in a multistage outpatient clinic. This chapter proposes a multiobjective simulation-based tabu search method enhanced by MP (MATS), which can take advantage of the flexibility of simulation, and at the same time, efficiently and effectively perform multiobjective optimization. The proposed method integrates the tabu search with mathematical programming (MP) to deliver (near) optimal Pareto fronts for appointment scheduling of patients in order to achieve bi-objectives of minimizing patients wait time and minimizing the clinic's overtime. The simulation model is utilized to address the stochastic nature of the problem and accommodate several constraints and parameters of the system. The performance of the proposed methods has been experimented over a range of scheduling factors - the number of patients, the number of patient types, and the coefficient of variance of service times.

The proposed method is compared with a well-known multiobjective evolutionary algorithm, NSGA-II, based on solution quality and computation time. The quality of solutions has been evaluated by two criteria: the closeness to the optimal Pareto front, and the smoothness of distribution of non-dominated solution in the objective function space. To compare the computation time, the amount of time required by each method to perform 500 function evaluations has been recorded.

Experiments suggest that MATS presents non-dominated solutions which are closer to the Pareto optimal front compared to NSGA-II. Results also indicate that MATS yields frontiers, which cover a larger range of values in the objective space. In addition, it is observed that the solutions presented by MATS are more evenly distributed than those of NSGA-II. Furthermore, MATS

requires less computation time than NSGA-II. In conclusion, MATS shows superior performance in terms of effectiveness and efficiency.

CHAPTER SIX

6. Application of multiagent tabu search in scheduling a case study of an OR department

6.1. Introduction

Scheduling of surgery cases in an OR department is an important and challenging problem; yet it has significant merit to be addressed efficiently as surgery department expenditures account for a significant portion of a hospital operating cost. Furthermore, a larger proportion of population of Canada will be in older cohorts in the near future, which require a larger number of surgical procedures compared with the younger cohorts. This results in a greater demand for healthcare services and more specifically surgical procedures in the coming years.

Significant cost of operating room departments, as well as the expected increase in the demand for surgical procedures, urges healthcare management to seek more efficient and cost effective utilization of their resources. Among these efforts, efficient surgery scheduling policies are adopted in order to minimize costs by reducing the overtime and blockings in OR departments.

This chapter attempts to apply the methodology presented in Chapter five to an OR department of a regional Canadian hospital with around 500 beds. This hospital is the only site in its health authority that offers services in specialties such as thoracic surgery. The OR department of this hospital includes 10 ORs and 20 post anesthesia care unit bays. Annually around 8000 surgeries

are performed in the hospital which 20% of them are emergency cases and the rest are elective cases. However, differing from a trauma center, most of the emergency cases in this hospital can be performed within 72 hours. This enables the surgical program to schedule the emergency and elective cases. This chapter only addresses scheduling of surgeries including emergency and elective cases that can be scheduled prior to the day surgery happens. That is, emergency cases that are not known by the beginning time of the scheduling horizon are out of scope of this study. In addition, patients in the understudy OR department include both inpatient and outpatient cases served in the department. Inpatient patients are treated as the same as the outpatient patients in terms of scheduling.

Chapter four of this thesis also addressed scheduling surgery cases in an OR department. This chapter can be differentiated from Chapter four from the following aspects:

- Chapter four focused on scheduling only outpatient cases, whereas, this chapter includes scheduling of inpatient cases as well as outpatient cases using actual surgical records.
- In Chapter four, it was assumed that all rooms were compatible with all surgery procedures, and there was no constraint on the compatibility of the rooms and patient types. However, in the understudy OR department of Chapter six, prior to the scheduling of cases, patients are assigned to the ORs considering the compatibility of OR/patients' types. The methodology thus differs from Chapter 4 by having a two-step process, i.e., 1) the assignment of patients to ORs using an IP model, and 2) scheduling of surgeries accordingly to results in 1).
- In Chapter four, although the problem addressed multiple objectives, the methodology used a weighted-sum approach which resulted in a single solution. However, the methodology offered in this chapter is capable of providing Pareto fronts rather than

suggesting a single solution. In fact, this feature contributes to the literature of surgery scheduling as only few numbers of studies addressed the problem using Pareto fronts. To the best of our knowledge, Gul et al. [69] approach is the only methodology in the literature of surgery scheduling that is capable of offering Pareto fronts.

- Surgeries carried out in this OR department can be classified into 10 specialties such as orthopedic, general surgery, etc. Each specialty can be further divided into procedures within each specialty. Often due to lack of data and required computational efforts, studies refrained from modeling at the procedure level can be modelled at the speciality level. Similarly, in Chapter four, due to lack of data, the problem was solved at the specialty level rather than at the procedure level. This chapter, however, uses three years of data to develop the simulation modeling of OR department and surgeries at the procedure level.

6.2. Problem description

This chapter addresses scheduling of surgery cases in an OR department of a major Canadian hospital. In this setting, patients undergo three main units of pre-operation, operation, and post operation in the understudy department.

The pre-operation unit includes the preparation of the patient for surgery and checking their documents. The pre-operation tasks usually take place in a holding area. The operation unit is the main unit of the department in which patients are transferred to the ORs and they undergo the surgical procedures. This unit includes 10 ORs, which are considered as stages of the operation unit. Each stage involves a team consisted of a surgeon, an anesthetist, and nurses. Because the nurses and anesthetists work based on salary, they are not considered as a resource constraint of the stage. Nurses and anesthetists are pre-assigned to rooms. Whereas, the availability of

surgeons who are paid based on the performed surgery cases is limited. Therefore, constraints on their availability are specifically addressed in the proposed models. The final unit includes the recovery of the patient stage and mainly happens in post anesthesia care unit (PACU). It is assumed that all stages including ORs and PACU are sufficiently staffed, and are ready to serve patients upon availability. We consider scheduling of a pre-determined number of patients of multiple types with stochastic service time at each stage in order to minimize bi-objectives of facility completion time and average waiting time of patients. This problem consists of two phases:

Phase 1: Assignment of the patient to ORs, and

Phase 2: Determination of patient's arrival time at each stage of the department.

The first phase of this approach forms a problem of assigning each patient to an OR considering the patient's surgery duration, OR's available time, and the compatibility of the OR with the surgical procedure of the patient. In order to cover the impact of variability in surgery durations, a factor called 'surgery duration hedging factor' is used to provide some flexibility in each OR's capacity, which will be explained later in detail. The result of this phase sets the sequence of stages that each patient should take. These sequences are then used at the second phase of our approach.

The second phase tackles the scheduling of patients considering availability of ORs and other resources in each stage, stochastic service time of patients at each stage, and availability of the surgeon assigned to the case with the goal of minimizing bi-objectives of facility completion time and average patients waiting time.

The completion time refers to the time that the last served patient leaves the post anaesthesia care unit (PACU). This measure reflects the total time that an OR department should be operational. Completion times longer than official operating hours incur overtime of the department. The average patients waiting time reflects the blocking at each stage. For instance, if a patient is waiting to receive a PACU bay, this patient is blocking an OR and delaying the future surgeries of that OR.

The number of patients and patient types are pre-determined in a higher-level planning step according to the available resources. The problem assumes patients' punctual arrivals. The stages in the department (except for the first stage) work according to the first-come-first-served rule. The first stage receives patients according to a schedule. Each patient is served by a specific surgeon. Therefore, it is important that the compatibility of resources and patients is considered.

6.3. Methodology

The methodology presented here adopts the approach proposed in Chapter five and tailors its components to address surgery scheduling in a case study OR department. The proposed method consists of two MP models to provide promising initial solutions, a simulation model to evaluate the performance of the solutions proposed by the overall method, and a multiagent tabu search algorithm (MATS) to provide Pareto front of surgery schedules. Figure 6.1 presents the components of the methodology used in this chapter.

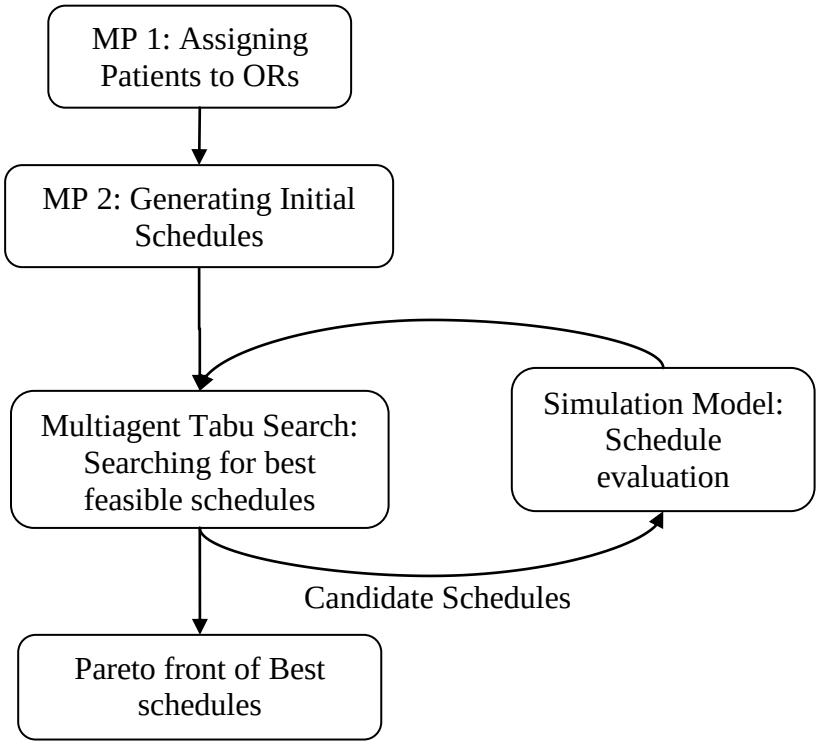

Figure 6.1 components of the proposed method

6.3.1. Integer programming (IP) models

Two MP models are used in this work. These MP are integer programming (IP) models in specific. The IP models play an important role in the methodology presented in this chapter. The IP models serve two purposes: a) assignment of patient to ORs, and b) providing quality surgery schedules which are used as initial solutions for MATS. This differentiates the role of IP models from the previous chapters where the sequence and stages that patient should go through were known in advance. In this chapter, the assignment of OR/patients is not known in advance, and is addressed by the proposed methodology.

The following assumptions are made in IP models:

1. The processing time of each patient type at each stage is deterministic (only in the IP models and not for simulations); the mean of the service duration at each stage is used to populate service durations,
2. The processing time of each patient type at each stage is an integer multiple of the basic time grid length, and
3. Each patient should go through all chosen stages and cannot skip any of those stages.

Notations

t time grid index *1,..., T, T* is the number of time grid;

s stage index 1, …, 13; stage 1 is the pre-operation unit; stages 2 to 11 represent 10 ORs (stages of the OR unit), and stage 12 is the post-operation unit; stage 13 is a dummy stage, which is used to identify patients' discharge;

p patient type index *1, ..., P; P* is the number of patient types;

i patient index *1, ..., I; I* is the number of patients;

d surgeon index *1, ..., D; D* is the number of available surgeons;

B_d set of patients who are served by surgeon d;

$C(s)$ set of stages that are ORs;

U_i the ordered set of stages that patient i should follow;

$[s]_{U_i}$ indicates the stage located before the position of stage s in the ordered set of U for patient type i.

M an arbitrarily large number;

Parameters

π_p surgery duration hedging factor for type p patients; a parameter to handle variability of surgery duration;

ρ_i type of the patient i;

$l_{p,s}$ surgery duration of patient type p at stage s;

R_s initial number of available resources at stage s;

b_d surgeon d available start time;

e_d surgeon d available end time;

o_s available hours of stages mainly used for ORs;

$a_{p,s}$ penalty of operating patient type p at stage s; large penalty values are used to address the incompatibility of patients and ORs;

α penalty coefficient for waiting of a patient per time grid;

β penalty coefficient of operating the department per time grid.

Variables

$x_{s,t,i}$ binary variable indicating if patient i at stage s starts being processed at time t;

$q_{s,t,i}$ binary variable indicating if patient i is waiting to be served at stage s at time t;

$q^{init}{}_{s,i}$ binary variable indicating if patient i is initially at stage s;

$X_{s,t,i}$ binary variable indicating if patient *i* treatment at stage *s* started by time *t*;

$r_{s,t}$ number of available idle resources at stage *s* and time *t*; each stage has its dedicated resources;

m last time block, in which all patients have been discharged (completion time);

y_t binary variable equal to one if at least a patient is discharged at time *t*; 0 otherwise;

$Z_{s,i}$ binary variable equal to one if patient *i* is assigned to OR *s.*

Phase 1:

Minimizing

$$\sum_{i} \sum_{s \in C(s)} z_{s,i} * a_{\rho_i,s} \qquad (6\text{-}1)$$

Subject to:

$$\sum_{s \in C(s)} z_{s,i} = 1 \qquad \forall i \qquad (6\text{-}2)$$

$$\sum_{i} \pi_{\rho_i} z_{s,i} * l_{\rho_i,s} \leq o_s \qquad \forall s \in C(s) \qquad (6\text{-}3)$$

$$z_{s,i} \; are \; binary \; variables \; \forall s, i$$

In Phase 1, the model chooses the best assignment of the patients to the ORs constrained by the compatibility of the patients' type and ORs, and available capacity of OR time. The compatibility of patients' type and ORs is handled by introducing large value penalties to non-compatible pairs of patient type and ORs. Expression (6-1) seeks a set of patient-to-OR assignments which leads to the least amount of penalty.

Expression (6-2) ensures that each patient has been assigned to an OR. An important feature of Phase 1 model is involving a surgery duration hedging factor, which is usually set between 1.0 and 1.5 based on the advice of the OR department management. Applying this factor adds flexibility to the capacity of an OR to deal with the variability of surgery durations. Expression (6-3) addresses the capacity constraint of ORs considering the hedging factor.

After the Phase 1 model is solved, the assignment of patients to ORs is achieved. This result is then used to populate U_i for Phase 2. U_i is an ordered set of stages that the i_{th} patient should go through. For example, in Phase 1, if the i_{th} patient has been assigned to OR2 the U_i will be {1, 3, 12, 13}. Note that the OR2 is the 3rd stage in the model and every patient has to go through Stage 12 (the post-operation unit) and Stage 13 (the dummy stage).

Phase 2 model addresses the scheduling of patients in the OR department considering the OR assignments decided in the first phase.

Phase 2:

Minimizing

$$\alpha \sum_{s} \sum_{t} \sum_{i} q_{s,t,i} + \beta m \tag{6-4}$$

Subject to:

$$X_{s,t,i} = \sum_{\tau=1}^{t} x_{s,\tau,i} \qquad \forall s, t, i \tag{6-5}$$

$$q_{s,t,i} = q_{s,i}^{init} - X_{s,t,i} + X_{[s-1]_{U_i},t-l_{\rho_i,[s-1]_{U_i}},i} \qquad \forall s, t, i \tag{6-6}$$

$$r_{s,t} = R_s - \sum_i X_{s,t,i} + \sum_i X_{s,t-l_{\rho_i,s},i} \qquad \forall s,t \tag{6-7}$$

$$M\left(1 - x_{s,t,i}\right) \geq t - e_d \qquad \forall s,d,t,i \in B_d \tag{6-8}$$

$$M\left(1 - x_{s,t,i}\right) \geq b_d - t \qquad \forall s,d,t,i \in B_d \tag{6-9}$$

$$My_t \geq \sum_i x_{13,t,i} \qquad \forall t \tag{6-10}$$

$$m \geq ty_t \qquad \forall t \tag{6-11}$$

$$X_{13,T,i} = 1 \qquad \forall i \tag{6-12}$$

$$\sum_{s \in C(s)} \left[\sum_{i \in B_d} X_{s,t-l_{\rho_i,s},i} - \sum_{i \in B_d} X_{s,t,i} \right] \leq 1 \qquad \forall t,d \tag{6-13}$$

$x_{s,t,i}$, $q_{s,t,i}$, are binary variables; $r_{s,t}$ are non-negative integer variables.

Initial conditions: the values for $x_{s,t,i}$, $X_{s,t,i}$, $r_{s,t}$ should be pre-specified from the previous scheduling horizon if applicable.

Expression (6-4) presents the objective function of the model that minimizes waiting time of patients and completion time of the facility. Expression (6-5) determines the cumulative variables. Expression (6-6) manages the balance of the patient flow among the stages of the department. It also ensures that each patient follows the selected path determined by U_i. Expression (6-7) controls the resource availability at each stage. Expressions (6-8) and (6-9) stipulate that the surgeries related to each doctor take place at the time that the surgeon is available. Expressions (6-10) and (6-11) determine the completion time of the schedule through controlling the last patient's discharge time block.

Expression (6-12) ensures that all patients will have the surgery and leave the department within the scheduling horizon. Expression (6-13) avoids performing more than one surgery at a time by each surgeon.

6.3.2. Simulation model

The simulation model provides a platform for evaluating the proposed schedules considering the stochastic surgery durations and constraints on resources and surgeons. The model includes the main three units of the OR department: pre-operation, operation, and post operation. Pre-operation unit includes a holding area with a capacity of 10 beds. The operation unit consists of 10 operating rooms. Each operating room is capable of accommodating surgeries of specific types. The post-operation unit consists of 20 PACU bays.

Three years of actual data is used to derive the probability distributions parameters of surgeries and PACU durations. These probability distributions of surgery durations are derived for each procedure, which resulted in 1056 distributions. As suggested by literature and our own experiments, it is assumed that all surgery duration distributions follow lognormal distribution. Pre-operation and PACU duration distributions are gathered based on the specialty of patients, which resulted in twenty distributions equally split for both inpatient and outpatient patients with ten specialties served by the department. In the operation stage, the room setup and cleanup after the surgery are also considered. Based on the suggestion of the department management, for room each setup and cleanup tasks, triangular distributions (triangular (5, 10, 15) in minutes) are used.

The simulation model accommodates several constraints and conditions that exist in the OR department and are challenging to be addressed by other methods. For instance, in simulation model the resources of the current stage of a patient are not released until the next stage is ready to accommodate the patient. This guarantees the safety of patients and mimics the industry practices. In situations where a patient's surgery is done but there is no empty spot available in PACU, the patient remains in the OR till a bay becomes available in PACU.

In order to address variability of the service durations at each stage, 40 replications of simulation runs for each schedule are performed. Forty replications are selected to make an appropriate trade-off between computation time and desired half width of the confidence intervals. The simulation model tallies and reports the results over the 40 replications as the final output of a given schedule.

The validity of the model has been confirmed by comparing the results with actual data of 15 days of the OR department. Measures such as discharge time of each patient, procedure finish time of each patient, and the facility completion time have been checked to ensure validity of the model. Figure 6.2 presents a comparison of patients' procedure finish time with simulated patients' procedure finish time for a sample day.

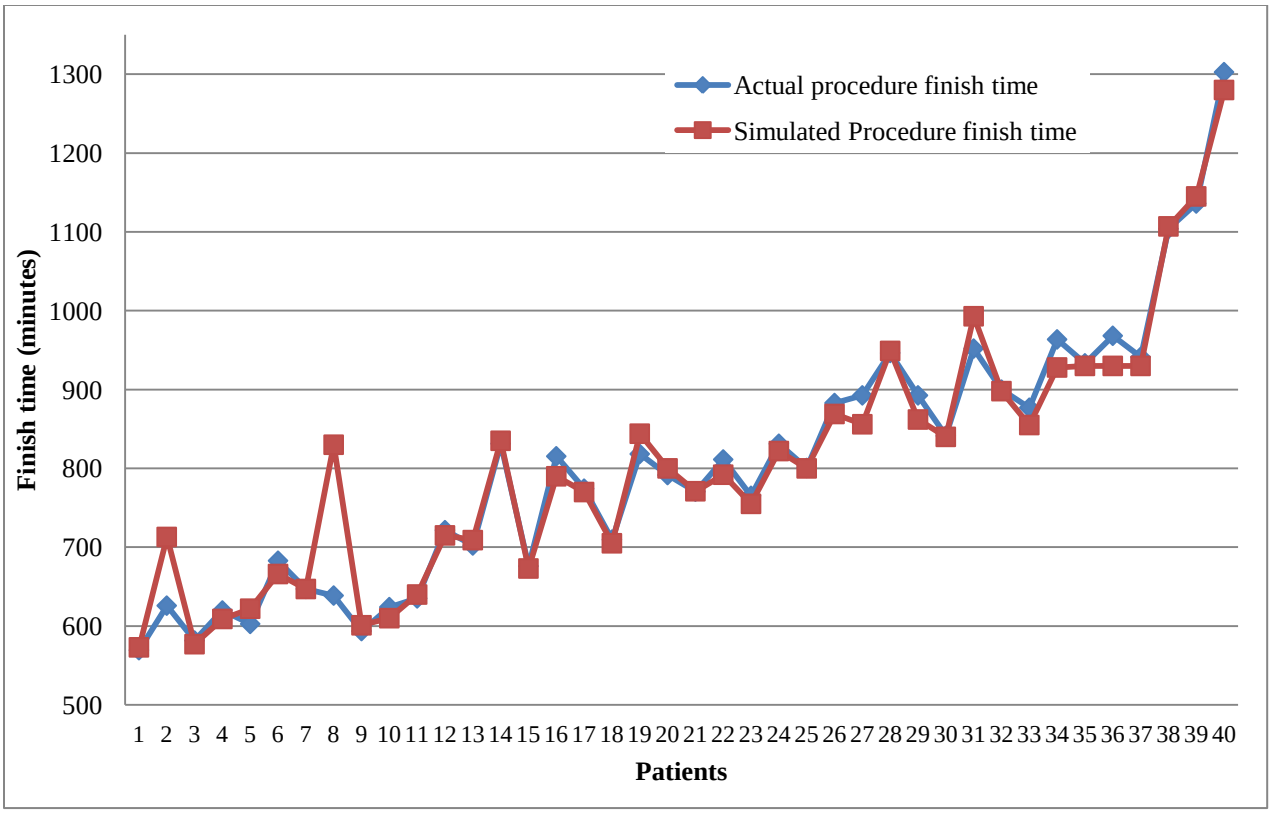

Figure 6.2 Actual patients' procedure finish time versus simulated patients' procedure finish time for a sample day

6.4. Experiments and results

The IP models have been developed using GAMS™, and CPLEX solver has been used to solve the IP models. The simulation model has been built in Arena™ 12 environment. Microsoft Visual C# was used to code the multiagent tabu search algorithm. The connection between MATS and the simulation model has been established using Arena object model component. A PC with i7 2.8GHz Intel® CPU with 8 GB of RAM has been used to run the experiments.

In order to evaluate the performance of the proposed method, five days of the OR department are selected. These days have been suggested by the management of surgery because the detailed data were readily available. The actual data for these five days are used to construct five test problems. Each test problem is implemented in the simulation model and the parameters of IP models and MATS are set accordingly. Table 6.1 presents the specification of test problems.

Table 6.1 Specification of test problems

Test problem	# of Patients	# of Patient Types	# of Surgeons
Day 1	39	30	10
Day 2	38	29	10
Day 3	36	29	9
Day 4	48	35	11
Day 5	32	30	10

Based on the study of the literature and the priorities stated by management, two metrics of completion time and average waiting time of patients have been selected to evaluate the performance of the methodology. For each test problem, five runs of the methodology are considered. Each run provides a Pareto front. The hypervolume (HV) indicator (introduced in Chapter five) for each run is then calculated. Based on the hypervolume values of these five runs, the superior front (the front with the largest HV value) and the inferior front (the front with the least HV value) are chosen. For each day, the best and the worst performance of the algorithm are then compared with the actual schedule retrieved from actual data.

6.4.1. DAY One

Figure 6.3 shows the result of the proposed method on the Day 1 test problem. Results suggest that the proposed method delivers a few non-dominated schedules (superior front) compared with actual schedule. It is observed that the proposed method offers schedules that significantly decrease the completion time while slightly raises patients' average waiting time.

Based on the coordinates of the actual schedule, a gray square has been drawn to represent the dominated area. The dominated solutions fall in the square. Most of the solutions from the front with lower HV value (the inferior) fall in the dominated area. However, the inferior front still presents few non-dominated solutions.

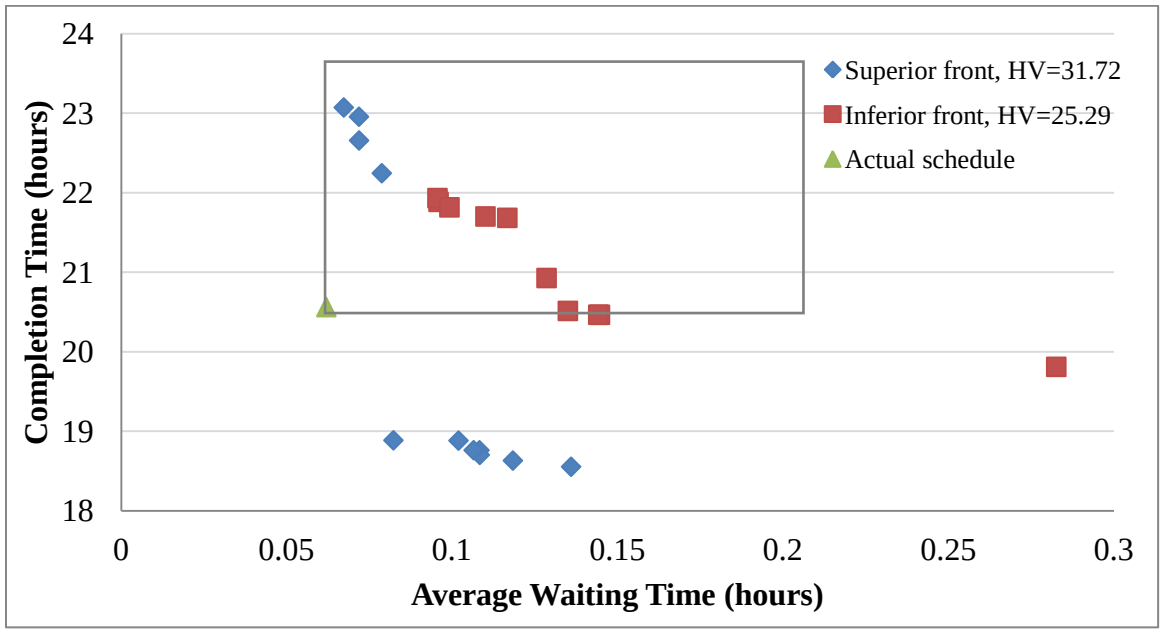

Figure 6.3 Day One; performance of proposed method versus actual schedule

6.4.2. Day Two

Figure 6.4 presents the result of applying the proposed method on test problem of Day two. Results suggest that the proposed method yields two fronts which are not quite distant from each

other. This fact is also reflected in the small difference in the HV values of the two fronts. In addition, both fronts include schedules that are not dominated by the actual schedule performance. However, the superior front with larger HV includes more non-dominated solutions. The dominated solutions are surrounded by a square in the figure.

It is observed that the proposed methodology can offer solutions which significantly better in terms of completion time at the expense of a slight increase in the waiting time of patients.

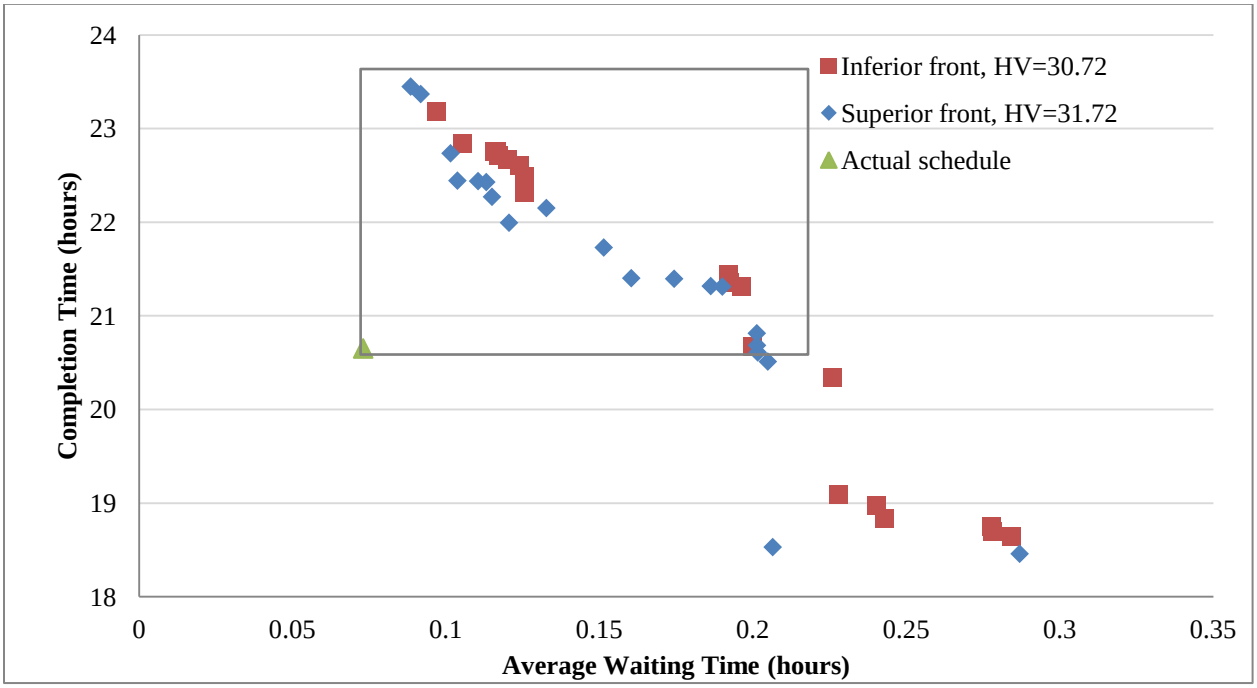

Figure 6.4 Day Two; performance of proposed method versus actual schedule

6.4.3. Day Three

Figure 6.5 depicts the performance of the proposed method compared with the actual schedule. The superior front offers non-dominated solutions compared with the actual schedule. However, the inferior front is entirely dominated by the performance of the actual schedule. The inferior front is surrounded by square which represents the dominated area.

The difference between the performance of the superior and inferior front is significant as reflected by their HV values. Although the superior front offers better solutions compared to the inferior front, it does not cover a wide range of solutions and its solutions are congested in a small area.

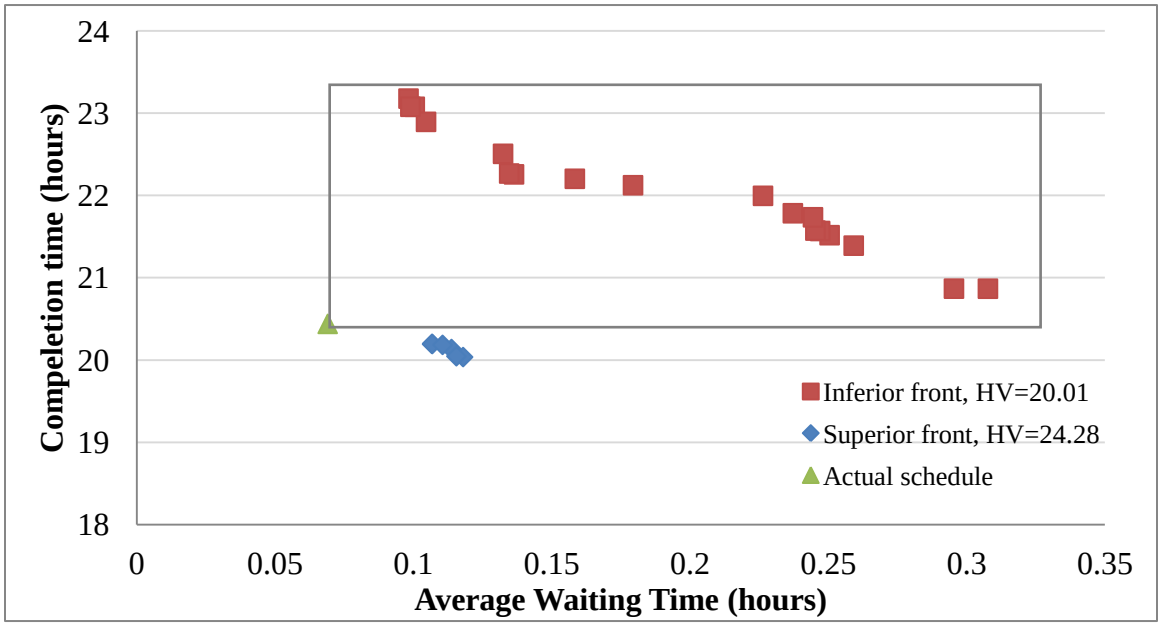

Figure 6.5 Day Three; performance of proposed method versus actual schedule

6.4.4. Day Four

The performance of the proposed model is assessed on the Day Four test problem and showed in Figure 6.6. The proposed offers only few schedules which are not dominated by the performance of the actual schedule. The dominated area by actual schedule is presented with a square which covers entire solutions of the inferior front. While these solutions are not dominated by the actual schedule, they do not provide significantly better completion times compared to the actual schedule.

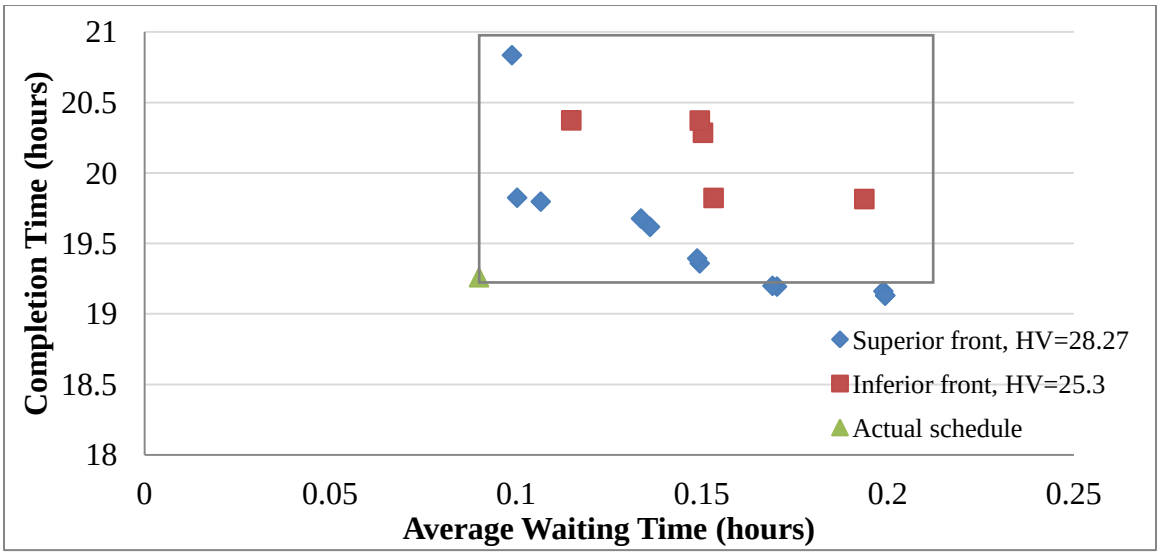

Figure 6.6 Day Four; performance of proposed method versus actual schedule

6.4.5. Day Five

Results of applying the proposed method on Day five test problem are presented in Figure 6.7. The proposed method delivers many non-dominated solutions compared with the performance of the actual schedule. While an apparent difference in the HV values of the inferior and superior fronts, both fronts provide non-dominated solutions which cover a wide range of solutions in terms of both objectives. The superior front completely dominates the actual results and offers better results that decrease the completion time and patients' average waiting time.

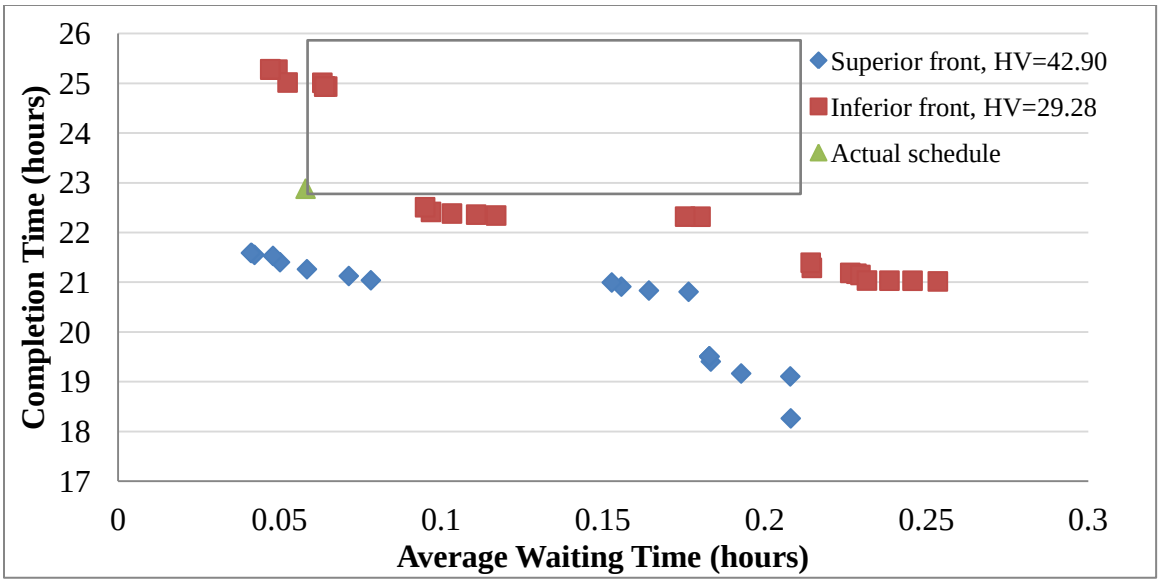

Figure 6.7 Day Five; performance of proposed method versus actual schedule

Considering the test problems and the performance of the proposed method, it is noticed that the strength of the proposed method is more apparent in providing schedule with less completion time in comparison with actual schedules, whereas actual schedules are mainly better in terms of patients' waiting time.

Overall, it is observed that the proposed method offers quality schedules which provide the OR department management with more options to schedule patients. This enables the management to select schedules which are more aligned with their priorities. For instance, if it is important for the management to decrease the completion time to reduce the overtime cost, they may choose the schedules with less completion time at an acceptable expense of patients' waiting time.

6.5. Insights for practitioners

Based on the case study presented in this chapter, and through the collaboration with the OR booking staff of the case study hospital and the health region's OR booking coordinator, invaluable insights have been obtained that can be of interest of both academic and practitioner

readers. In this section, a few observations and simple recommendations are presented which may be useful to improve performance of surgery scheduling in different facilities.

1. Overtime should be avoided whenever possible.

Based on interviews with the OR management and the OR booking personnel, it is observed that aiming for less overtime has more priority in their scheduling than other factors such as minimizing cancellations and OR idleness. Following reasons have been mentioned:

- The financial implication of having overtime suggests against it. The cancelled cases can ultimately be done in the regular hours of future days if clinical safety measures allow.

- Clinical safety measures recommend avoiding long working days for surgeons. This not only leads to less complications due to surgeons' fatigue in OR departments, but also ensures that patients with major procedures are stabilized in the recovery and arrive at surgical wards before evening shifts start.

2. It is recommended to schedule cases with longer durations or larger variability (e.g., inpatient cases) in the beginning or in the middle of the day.

Cases with longer durations often have larger variability. This insight is mainly driven by the fact that scheduling cases with longer duration at the end of the day may result in more overtimes. In contrast, having longer cases at the beginning or middle of the day may result in more cancellations. As mentioned above, the management are more inclined to see less overtime than having cancellations when it comes to tradeoffs. Additionally, applying this recommendation gives the management more flexibility to compensate for delays due to long cases by taking reactive measures.

3. It is recommended to schedule cases of a surgeon in the same room.

Scheduling all patients of a surgeon in a specific room will provide the opportunity of having shorter turnaround time as most equipment for a day of surgery may be placed in the same room and it leads to less transportation and setup time. In addition, it would avoid cancellations and delays for other surgeons' cases if a case goes longer than expected. This is a desirable outcome as the surgeon time is the most expensive and scarce resource in the OR department. Applying this recommendation is subject to availability of enough number of cases for a surgeon to fully utilize an OR.

4. It is recommended to schedule similar cases in the same OR.

This recommendation mainly aims at reducing setup time by placing similar procedures (i.e., cases that require same equipments) in the same room. It particularly pertains to procedures that require special equipment such as open-heart surgeries or orthopedic cases. Applying this recommendation is subject to availability of enough number of similar cases to fully utilize ORs.

5. It is recommended to schedule a mix of procedures with high and low variability on the same resource.

This recommendation targets balancing of the variability among resources (i.e., surgeons and ORs). For instance, the schedule of a surgeon for one day can be built by scheduling few inpatient cases (that often have large variability) along with some outpatient cases (that often have less variability).

6. It is recommended to schedule cases with less flexibility first.

Some cases can be assigned to only one type of resource, whereas others may be flexible and can be assigned to several alternative resources. This recommendation encourages scheduling of less flexible cases prior to more flexible cases to satisfy compatibility requirements. For example, scheduling of a case with an open-heart surgery (that can be performed only in a specific room) should be done in prior to the scheduling of a general surgery case (that can be performed in most rooms).

All these recommendations should be applied in conjunction with the clinical and operational constraints that exist in the OR departments. For instance, usually diabetic and pediatric cases are done in the early mornings to avoid long fasting of patients or usually surgeons' preference and clinical opinion have the final say in the sequencing of the surgeries.

6.5. Conclusion

This chapter aimed at applying the MATS which is presented in Chapter five to scheduling of a case study of an OR department in a major Canadian hospital. The cases study of OR department includes three units of pre-operation, operation, and post-operation which are broken down in multiple stages. The operation unit embodies ten ORs that each can only serve specific type of patients. The scheduling problem in this OR department focuses on a) the assignment of the patients to ORs and, b) the determination of the arrival time of the patients at different stages of the department to minimize bi-objectives of completion time and patients' average waiting time. The OR department serves multiple types of patients with possibly different stochastic service durations at each stage. The problem is constrained by the availability of resources at each stage and compatibility of patient types to the ORs. Additionally, the availability the surgeons for each patient was addressed. Note that each patient can be served with a specific surgeon.

In order to apply MATS to the targeted problem, two components of integer programming and simulation models have to be modified and developed based on the case study of OR department. The IP model first determines the assignment of patients to ORs and then seeks the best schedule for the deterministic version of the problem with the goal of minimizing completion time and average waiting time of patients.

The simulation model mimics the OR department and its stages informed by three years of actual data to determine service durations. Using this abundance of actual data the simulation model not only breakdowns patient by their specialty but also divides them one level further and classifies patients by their surgical procedures.

In order to evaluate the performance of the proposed method, five days of the OR department were selected based on the availability of the detailed data. Five test problems were constructed based on these days. The proposed method was run five times for each test problem. The inferior and superior fronts of the schedules were selected from the five runs based on their hypervolume indicator values.

The experiments suggest that the proposed method was capable of offering solutions that were not dominated by the performance of the actual schedule that had been retrieved from historical data. That is, the proposed method can deliver solutions that are at least as good as actual schedules employed by the practitioner. Additionally, it enables the department management to select the schedules which are more aligned with their priorities.

Moreover, it was observed that the proposed method was more successful in providing schedules with less completion time rather than less average patient waiting time when compared with actual schedules.

CHAPTER SEVEN

7. Conclusions and future research

Patient appointment scheduling has attracted many researchers in the past decades due to its influence on the cost reduction of healthcare and its complexity. Many researchers made attempts to tackle various aspects of this problem by employing operations research techniques. Optimization methods and simulation modeling are among the utilized methods to address this problem.

In outpatient appointment scheduling, several methods have been used to provide an optimal (or near optimal) schedule for patients and providers, based on performance measures chosen by researchers. These methods can be summarized in three main categories: analytical approaches, simulation, and combination of simulation with heuristic methods. Analytical methods include mathematical modeling and queuing theory.

Although analytical methods can usually offer an optimal solution, they have to simplify the real world outpatient scheduling problem due to the problem complexity. As a result of this simplification, only few environmental factors such as multiple stages, no-show rates, patient priority, and resource allocations may be considered by these techniques.

For instance, in queuing theory researchers made strong assumptions about service durations by considering exponential distribution and Markovian behaviour. However, these assumptions do not accurately represent actual clinical environments. In addition, most works in this context

consider the steady state behaviour which is not realized in these systems due to the limitation on both the number of served patients and clinic's operating hours.

On the other hand, simulation can study multiple factors in appointment scheduling. Environmental factors can be easily modeled using simulation while different constraints can be applied. Simulation, however, is usually considered as a tool to evaluate performance of a system and does not usually involve any optimization strategy to provide optimal solutions. This shortcoming encourages combination of simulation with other components such as heuristics to guide the system toward an optimal schedule. To overcome these shortcomings, researchers recently have used a combination of simulation and optimisation methods to leverage the simulation flexibility to model the system's complexities and the power of optimization methods to reach (near) optimal schedules. However, literature still lacks effective and efficient methods that consider 1) multistage multi-server clinics, 2) multiple types of patients, 3) non-identical stochastic service duration at each stage, 4) availability of resources with time window constraints, 5) compatibility of resources, and 6) patient with heterogeneous serves sequence.

Furthermore, although several researchers studied multiple performance criteria in this context, few attempts have been made to develop methods that result in Pareto optimal fronts of schedules. In the past, many studies used weighted-sum function of multiple performance criteria. This approach leads only to a single optimal schedule that is extremely dependent on the selected weights. However, a multiobjective optimization method that delivers a Pareto optimal front of schedules enables the management to select the optimal scheduling decision according to the clinic's conditions.

This thesis proposes simulation-based optimization methods incorporating elements of mathematical programming, simulation modeling, and tabu search. The proposed methods in Chapters three to five focus on the appointment scheduling in a specific healthcare unit from different perspectives.

7.1 Thesis contributions

Overall, contributions of this thesis can be summarized as follows:

- This thesis proposes mathematical programming (MP) models that represent the patient flow throughout the clinic/department and address the challenges of appointment scheduling in such systems. The MP models include constraints on the availability and compatibility of resources. They assist the proposed multiobjective optimal scheduling method to improve its search performance.
- Efficient and effective global and multiobjective optimization methods are developed for appointment scheduling problems in multiple healthcare facilities.
- This study proposes novel appointment scheduling methods to address challenges of real world problems including:
    - Tackling appointment scheduling in multistage, multiserver outpatient and surgery facilities
    - Serving multiple patient types with stochastic service duration at each stage
    - Considering non-identical service time distribution for each patient type at each stage, regardless of the distribution type of the service time.
    - Addressing patients with heterogeneous service sequence for each patient type
    - Optimizing multiple performance criteria and delivering a Pareto optimal front

- The proposed methods in this thesis have been applied to two cases study OR departments and the performance of these methods are evaluated.

Chapter three considers a multistage outpatient clinic which serves multiple types of patients with stochastic service duration including revisiting patients. A clinic has been represented in four stages such as reception, nurse visit, doctor consultation, and lab/X-ray. A proportion of patients requiring testing or X-ray imaging will return to the doctor for consultation following the lab/X-ray visit. The proposed method offers (near) optimal appointment schedules with the goal of minimizing waiting time of patients. The minimization is constrained by the availability of resources at each stage and a limited time for patient appointments.

Chapter three includes the following contributions:

- Two simulation-based optimization methods-enhanced tabu search (ETS) and mathematical programming based enhanced tabu search (MPETS) - have been proposed. ETS is developed based on integration of a simulation model with a tabu search, which uses an auxiliary objective to decrease simulation runs. MPETS enhances the ETS by using a mathematical programming model.
- A mathematical programming model has been developed in order to present an appointment schedule with minimum patient waiting. Patient flow and resource constraints in different stages are included in the model.
- Several test problems have been developed based on the factors that were found to be effective on the performance of algorithms in preliminary experiments. These factors include the patient mix and number of patients, the type of probability distribution function used by the simulation, and the variance of the service duration.

- Box-Behnken design of experiment has been used to perform the experiments. This design provides insights on the impact of influencing factors on the performance of the proposed method as well as its counterpart method, OptQuest.

The comparison of ETS and MPETS reveals that the combination of the mathematical programming model with metaheuristics improves the performance of algorithms. Our experiments further suggest superior performance of MPETS method over OptQuest in terms of both the solution quality and computation time in all tested instances.

In addition, our study shows that the number and mix of patients are the most influential factors affecting the effectiveness and efficiency of the algorithms. The variance of service time distribution is the second most significant factor influencing the performance of the algorithms.

Chapter four focuses on the challenges involved in the appointment scheduling of OR departments. This chapter presents the OR department as a three-stage unit including pre-operation, operation, and post operation. The stages are located in pre-operation holding area, operating rooms, and post anaesthesia unit (PACU). The ORs are represented as separate servers located in the second stage and function in parallel. While patients are assumed to pursue the same path through the stages, they require different service durations at each stage. Furthermore, each patient type requires a determined type of surgeon for the operation which stipulates resource compatibility constraints. Resource availability constraints account for the number of idle and busy resources such as pre-operation beds and PACU beds. Additionally, time-window constraints control the availability of surgeons, which directly influence the appointment schedules of patients within each specialty.

The contributions of chapter four are as follows:

- It examines three main performance criteria of patient waiting time, completion time, and number of case cancellations.
- Each patient type requires a specific type of surgeon for the operation, which stipulates resource compatibility constraints on the appointment scheduling in addition to resource availability concerns.
- Three simulation-based optimization methods have been proposed for the appointment scheduling of the OR department.
- A mathematical programming model has been developed in order to present appointment schedules with minimum patient waiting time and completion time. The model includes features such as patient flow and resource constraints in different stages. Resource constraints address challenges of compatibility of surgeons to patient time as well as time-window constraints on availability of surgeons.
- Heuristics have been proposed in order to remedy the impact of relaxing the integrality constraints in the model.
- To assess the performance of the proposed algorithms, several test problems have been developed considering the effective factors. These factors include the number of patients, number of ORs, and coefficient of variance of service time. An extended range for each factor has been selected based on a case study of an OR department in a major Canadian hospital.
- Application of simple scheduling rules has been examined in the context of surgery scheduling and compared with the proposed methods. This study provides insights for practitioners to improve the performance of appointment scheduling.

- Finally, this chapter presents a case study OR department in a major Canadian hospital where the proposed method has been applied to schedule appointments for 24 days in the OR department.

The experiments confirm the significance of employing mathematical programming in improving the simulation-based optimization results. Comparison of integer programming enhanced tabu search (IPETS) and binary linear programming enhanced tabu search (BPETS) recommends application of BPETS for practical purposes based on its effectiveness and efficiency in solving test problems ranging in size from small to large.

Moreover, the performance study of simple scheduling rules indicates that methods based on simple scheduling rules are not promising approaches when dealing with systems that have providers restricted by time window constraints (e.g., surgeons in OR departments). That is, use of simple scheduling rules may result in several cancellations, because these rules do not consider the time constraints. However, simple scheduling rules can be utilized in the systems that include providers with relaxed time-window constraints.

Overall, chapter four yields two main points in the appointment scheduling of OR departments. First, metaheuristics are recommended to be used in services in which surgeons or resources are restricted by tight time constraints, whereas simple scheduling rules are recommended in services with no time constraints. Second, applying MP models leads to improvements in the performance of metaheuristics in appointment scheduling.

Chapter five involves addressing appointment scheduling in multistage clinics which offer a wide variety of services and serve multiple types of patients with different service sequences and stochastic service durations. The patient types in this type of clinic range from simple doctor

consultation or annual checkups, to minor surgical or ambulatory care procedures. Contrary to the clinics discussed in previous chapters that assumed the same pathway and service sequence for all patient types, this chapter allows for multiple pathways and different service sequences specific to each patient type. The clinic is composed of eleven stages including doctor revisit and discharge of patients. The doctor revisit happens when a patient needs to revisit the doctor after having a labor imaging procedure done. Resource availability is considered for each stage with resources, such as nurses, doctors, ORs, pre-operation beds, PACU beds, etc.

The contributions of Chapter five include:

- It addresses appointment scheduling in multistage clinics that offer a wide variety of services and serve multiple types of patients with heterogeneous service sequences and stochastic service durations.
- A multiobjective simulation-based tabu search method (MATS) has been proposed and enhanced by a mathematical programming (MP) model. The proposed method integrates the tabu search with MP to deliver (near) optimal Pareto fronts in order to minimize the patients' waiting time and overtime of the clinic.
- Contrary to previous works, the proposed method uses a multi-agent method that leads to Pareto front of solutions. Most previous studies targeted at multiple performance criteria by using a weighted-sum objective function which led to a single solution.
- Chapter five proposes a mathematical programming model in order to present appointment schedules. The model includes features such as patients with heterogeneous service sequences and resource constraints in different stages.

- The performance of the proposed method is assessed through extensive experiments over a range of scheduling factors: the number of patients, the number of patient types, and the coefficient of variance of service times.

The proposed method has been compared with NSGA-II method based on solution quality and computation time. Results suggest that the proposed multi agent tabu search MATS delivers non-dominated solutions which are closer to the Pareto optimal front compared to those generated by NSGA-II. In addition, MATS yields frontiers, which cover a larger range of values in the objective space.

Chapter six provides following contributions:

- It includes scheduling of inpatient cases as well as outpatient cases using actual surgical records.
- In the understudy OR department of Chapter six, prior to the scheduling of cases, patients are assigned to the ORs considering the compatibility of OR/patients' types.
- This chapter, uses three years worth of data to develop the simulation modeling of OR department and surgeries at procedure level. This one level more granular than the other studies in the literature and Chapter four.

Chapter six aims to apply methodology proposed in Chapter five to a case study OR department. In order to apply the multiobjective scheduling method, two components of the IP models and simulation model have been developed to mimic the OR department. Multiple days of the case study OR department have been selected and used to construct test problems through which the performance of the proposed method has been evaluated. The experiments suggest that the proposed method is capable of offering multiple schedules that are not dominated by the

performance of actual schedules used by practitioners. Not only the proposed method generates results as good as the actual schedules but it enables department management to choose among the provided schedules based on their priorities.

Overall, In spite of its advantages, the proposed methodology bears its own drawbacks. Due to the application of metaheuristic optimization method, it may not be possible to offer a mathematical proof on the optimality of the generated results. However, the performance of the proposed method has been tested with best available methods used in the literature (OptQuest, NSGAII) and proves to deliver superior results in terms of quality of solutions and computation time. Another challenge in developing and implementing the proposed methodology is the development of the MP model for the clinics. For instance, composing the MP model for systems in the presence of demand uncertainty might be challenging, which can be resolved by using constructive heuristics or dynamic programming.

7.2. Future work

Several directions for future research are apparent from this study. Future research may include applying the proposed method in larger settings such as hospitals where the appointment scheduling affects downstream and logistic departments. For instance, studying the relationship of appointment scheduling in OR departments and the staff planning and scheduling in sterile processing departments is a valuable contribution.

Second, the proposed methods may be extended to address challenges of scheduling a multistage clinic in the presence of uncertainty in the number of patients. This challenge differs from the cancellations incurred due to lack of resources or time which have been addressed in Chapter 4. The patients in Chapter 4, assumed to be punctual (i.e., they do not have tardiness in their arrival

to the clinic) and the number of them is known. Whereas, in the recommended future research, the no-shows are possible.

Furthermore, this research could be adapted to tackle other scheduling challenges such as processes requiring setup times/cost, and resources with time-window constraints while patients have different service sequences.

Additionally, multiobjective scheduling approaches for multistage clinics with resource compatibility constraints can be extended to address dynamic and reactive scheduling challenges. The efficiency and effectiveness of these approaches should be further studied under different scenarios.

Finally the proposed methods in this thesis can be applied to the scheduling on other production system such as manufacturing. The problem tackled in Chapter 4 is comparable with constrained stochastic flowshop scheduling problem in the presence of multiple objectives. Additionally, the problem addressed in Chapters 5 and 6 reflects constrained stochastic jobshop scheduling problem in the presence of multiple objectives. The proposed methods in this chapter can be tailored to these problems. The performance of these methods can be compared with the available methods in these fields.

References

[1] Centers for Medicare & Medicaid Services, “National Health Expenditures,” 2009.

[2] Canadian Institute for Health Information (CIHI), “National Health Expenditure Trends, 1975 to 2010,” 2010.

[3] E. L. Villegas, “Outpatient appointment system saves time for patients and doctor,” *Hospitals*, vol. 41, no. 8, pp. 52–57, Apr. 1967.

[4] P. M. Vanden Bosch and D. C. Dietz, “Minimizing expected waiting in a medical appointment system,” *IIE Transactions*, vol. 32, no. 9, pp. 841–848, 2000.

[5] X. Qu, “Development of appointment scheduling rules for open access scheduling,” PhD Thesis, Purdue University, 2006.

[6] W. L. Johnson and L. S. Rosenfeld, “Factors affecting waiting time in ambulatory care services,” *Health Services Research*, vol. 3, no. 4, pp. 286–295, 1968.

[7] D. V Lindley, “The theory of queues with a single server,” in *Mathematical Proceedings of the Cambridge Philosophical Society*, 1952, vol. 48.

[8] N. T. J. Bailey, “A study of queues and appointment systems in hospital outpatient departments, with special reference to waiting times,” *Journal of the Royal Statistical Society*, vol. 14, no. 2, pp. 185–199, 1952.

[9] R. B. Fetter and J. D. Thompson, “Patients’ waiting time and doctors' idle time in the outpatient setting,” *Health Services Research*, vol. 1, no. 1, pp. 66–90, 1966.

[10] K. J. Klassen and T. R. Rohleder, “Scheduling outpatient appointments in a dynamic environment,” *Journal of Operations Management*, vol. 14, no. 2, pp. 83–101, Jun. 1996.

[11] T. R. Rohleder and K. J. Klassen, “Using client-variance information to improve dynamic appointment scheduling performance,” *Omega*, vol. 28, no. 3, pp. 293–302, Jun. 2000.

[12] N. T. J. Bailey, “Queueing for medical care,” *Applied Statistics*, pp. 137–145, 1954.

[13] M. J. White and M. C. Pike, “Appointment systems in out-patients’ clinics and the effect of patients' unpunctuality,” *Medical Care*, vol. 2, no. 3, pp. 133–145, 1964.

[14] B. Jansson, “Choosing a good appointment system-a study of queues of the type (d, m, 1),” *Operations Research*, pp. 292–312, 1966.

[15] M. Brahimi and D. J. Worthington, “Queueing models for out-patient appointment systems. A case study,” *Journal of the Operational Research Society*, vol. 42, no. 9, pp. 733–746, 1991.

[16] C. J. Ho and H. S. Lau, “Minimizing total cost in scheduling outpatient appointments,” *Management Science*, vol. 38, no. 12, pp. 1750–1764, 1992.

[17] C. D. Pegden and M. Rosenshine, “Scheduling arrivals to queues,” *Computers & Operations Research*, vol. 17, no. 4, pp. 343–348, 1990.

[18] A. Soriano, “On the problem of batch arrivals and its application to a scheduling system,” *Operations Research*, vol. 14, no. 3, pp. 398–408, 1966.

[19] E. J. Rising, R. Baron, and B. Averill, “A systems analysis of a university-health-service outpatient clinic,” *Operations Research*, vol. 21, no. 7, pp. 1030–1047, 1973.

[20] C.-J. Liao, C. D. Pegden, and M. Rosenshine, “Planning timely arrivals to a stochastic production or service system,” *IIE Transactions*, vol. 25, no. 5, pp. 63–73, 1993.

[21] L. Liu and X. Liu, “Block appointment systems for outpatient clinics with multiple doctors,” *Journal of the Operational Research Society*, vol. 49, no. 12, pp. 1254–1259, 1998.

[22] L. Liu and X. Liu, “Dynamic and static job allocation for multi-server systems,” *IIE Transactions*, vol. 30, no. 9, pp. 845–854, 1998.

[23] P. M. Vanden Bosch, D. C. Dietz, and J. R. Simeoni, “Scheduling customer arrivals to a stochastic service system,” *Naval Research Logistics*, vol. 46, no. 5, pp. 549–559, 1999.

[24] B. Denton and D. Gupta, “A sequential bounding approach for optimal appointment scheduling,” *IIE Transactions*, vol. 35, no. 11, pp. 1003–1016, Nov. 2003.

[25] P. P. Wang, “Static and dynamic scheduling of customer arrivals to a single-server system,” *Naval Research Logistics*, vol. 40, no. 3, pp. 345–360, 1993.

[26] P. Wang, “Optimally scheduling N customer arrival times for a single-server system,” *Computers & Operations Research*, vol. 24, no. 8, pp. 703–716, Aug. 1997.

[27] P. P. Wang, “Sequencing and scheduling N customers for a stochastic server,” *European Journal of Operational Research*, vol. 119, no. 3, pp. 729–738, 1999.

[28] L. Robinson and R. Chen, “Scheduling doctors’ appointments: optimal and empirically-based heuristic policies,” *IIE Transactions*, vol. 35, no. 3, pp. 295–307, Mar. 2003.

[29] T. Cayirli and E. Veral, “Outpatient scheduling in health care: a review of literature,” *Production and Operations Management*, vol. 12, no. 4, pp. 519–549, 2003.

[30] E. Marcon and F. Dexter, “Impact of surgical sequencing on post anesthesia care unit staffing,” *Health Care Management Science*, vol. 9, no. 1, pp. 87–98, 2006.

[31] E. Marcon and F. Dexter, “An observational study of surgeons’ sequencing of cases and its impact on postanesthesia care unit and holding area staffing requirements at hospitals,” *Anesthesia & Analgesia*, vol. 105, no. 1, pp. 119–126, 2007.

[32] G. Westman, S. O. Andersson, S. Ferry, and P. Fredriksson, “Waiting room time in the assessment of an appointment system in primary care,” *Scandinavian Journal of Primary Health Care*, vol. 5, no. 1, pp. 35–40, Feb. 1987.

[33] X. M. Huang, “Patient attitude towards waiting in an outpatient clinic and its applications,” *Health Services Management Research*, vol. 7, no. 1, pp. 2–8, Feb. 1994.

[34] B. Cardoen, E. Demeulemeester, and J. Beliën, “Operating room planning and scheduling: A literature review,” *European Journal Of Operational Research*, vol. 201, no. 3, pp. 921–932, 2010.

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

[36] J. D. C. Little, “A Proof for the Queuing Formula: L = W,” *Operations Research*, vol. 9, no. 3, pp. 383–387, May 1961.

[37] P. T. VanBerkel and J. T. Blake, “A comprehensive simulation for wait time reduction and capacity planning applied in general surgery,” *Health Care Management Science*, vol. 10, no. 4, pp. 373–385, 2007.

[38] M. Lamiri, X. Xie, A. Dolgui, and F. Grimaud, “A stochastic model for operating room planning with elective and emergency demand for surgery,” *European Journal of Operational Research*, vol. 185, no. 3, pp. 1026–1037, 2008.

[39] S.-C. Kim and I. Horowitz, “Scheduling hospital services: the efficacy of elective-surgery quotas,” *Omega*, vol. 30, no. 5, pp. 335–346, 2002.

[40] M. Brahimi and D. J. Worthington, “The finite capacity multi-server queue with inhomogeneous arrival rate and discrete service time distribution- and its application to continuous service time problems,” *European Journal of Operational Research*, vol. 50, no. 3, pp. 310–324, 1991.

[41] D. Gross and C. M. Harris, *Fundamentals of Queueing Theory (3rd ed.)*. Springer-Verlag, 1985.

[42] A. Mercer, “A queueing problem in which the arrival times of the customers are scheduled,” *Journal of the Royal Statistical Society.Series B (Methodological)*, pp. 108–113, 1960.

[43] A. Mercer, “Queues with scheduled arrivals: a correction, simplification and extension,” *Journal of the Royal Statistical Society.Series B (Methodological)*, pp. 104–116, 1973.

[44] B. Zeng, A. Turkcan, J. Lin, and M. Lawley, “Clinic scheduling models with overbooking for patients with heterogeneous no-show probabilities,” *Annals of Operations Research*, vol. 178, no. 1, pp. 121–144, Jun. 2009.

[45] S. A. Erdogan and B. T. Denton, “Surgery Planning and Scheduling,” in *Wiley Encyclopedia of Operations Research and Management Science*, John Wiley & Sons, Inc., 2010.

[46] W. E. Spangler, D. P. Strum, L. G. Vargas, and J. H. May, “Estimating procedure times for surgeries by determining location parameters for the lognormal model.,” *Health Care Management Science*, vol. 7, no. 2, pp. 97–104, May 2004.

[47] D. P. Strum, J. H. May, and L. G. Vargas, “Modeling the uncertainty of surgical procedure times: comparison of log-normal and normal models.,” *Anesthesiology*, vol. 92, no. 4, pp. 1160–1167, Apr. 2000.

[48] B. E. Fries and V. P. Marathe, “Determination of optimal variable-sized multiple-block appointment systems,” *Operations Research*, vol. 29, no. 2, pp. 324–345, 1981.

[49] R. D. Baker and P. L. Atherill, “Improving appointment scheduling for medical screening,” *IMA Journal of Management Mathematics*, vol. 13, no. 4, pp. 225–243, Oct. 2002.

[50] V. N. Hsu, R. De Matta, and C.-Y. Lee, “Scheduling patients in an ambulatory surgical center,” *Naval Research Logistics*, vol. 50, no. 3, pp. 218–238, 2003.

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

[52] I. Ozkarahan, “Allocation of surgical procedures to operating rooms,” *Journal of Medical Systems*, vol. 19, no. 4, pp. 333–352, Aug. 1995.

[53] D. Sier, P. Tobin, and C. McGurk, “Scheduling Surgical Procedures,” *Journal of the Operational Research Society*, vol. 48, no. 9, pp. 884–891, Oct. 1997.

[54] D. Pham and a Klinkert, “Surgical case scheduling as a generalized job shop scheduling problem,” *European Journal of Operational Research*, vol. 185, no. 3, pp. 1011–1025, Mar. 2008.

[55] A. Testi and E. Tànfani, “Tactical and operational decisions for operating room planning: Efficiency and welfare implications,” *Health Care Management Science*, vol. 12, no. 4, pp. 363–373, Dec. 2008.

[56] D. Min and Y. Yih, “Scheduling elective surgery under uncertainty and downstream capacity constraints,” *European Journal of Operational Research*, vol. 206, no. 3, pp. 642–652, Nov. 2010.

[57] M. Lamiri, F. Grimaud, and X. Xie, “Optimization methods for a stochastic surgery planning problem,” *International Journal of Production Economics*, vol. 120, no. 2, pp. 400–410, Aug. 2009.

[58] S. Batun, B. T. Denton, T. R. Huschka, and A. J. Schaefer, “Operating Room Pooling and Parallel Surgery Processing Under Uncertainty,” *INFORMS Journal on Computing*, vol. 23, no. 2, pp. 220–237, Jul. 2010.

[59] J. D. Welch and N. T. J. Bailey, “Appointment systems in hospital outpatient departments,” *Lancet*, vol. 259, no. 6718, pp. 1105–1108, May 1952.

[60] C. Ho, “Evaluating the impact of operating conditions on the performance of appointment scheduling rules in service systems,” *European Journal of Operational Research*, vol. 112, no. 3, pp. 542–553, Feb. 1999.

[61] J. Vissers, “Selecting a suitable appointment system in an outpatient setting,” *Medical Care*, vol. 17, no. 12, pp. 1207–1220, 1979.

[62] H. H. Schmitz and N. K. Kwak, “Monte Carlo simulation of operating-room and recovery-room usage,” *Operations Research*, vol. 20, no. 6, pp. 1171–1180, 1972.

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

[64] J. C. Lowery, “Simulation of a hospital’s surgical suite and critical care area,” in *Proceedings of the 24th Conference on Winter Simulation, Arlington, VA*, 1992, pp. 1071–1078.

[65] J. C. Lowery and J. A. Davis, “Determination of operating room requirements using simulation,” in *Proceedings of the 31st Conference on Winter Simulation, Phoenix, AZ*, 1999, pp. 1568–1572.

[66] D. C. Tyler, C. A. Pasquariello, and C. H. Chen, “Determining optimum operating room utilization,” *Anesthesia & Analgesia*, vol. 96, no. 4, pp. 1114–1121, 2003.

[67] T. R. Huschka, B. T. Denton, S. Gul, and J. W. Fowler, “Bi-criteria evaluation of an outpatient procedure center via simulation,” in *Proceedings of the 39th conference on Winter Simulation, Washington, D.C.*, 2007, pp. 1510–1518.

[68] K. J. Klassen and R. Yoogalingam, “Improving performance in outpatient appointment services with a simulation optimization approach,” *Production and Operations Management*, vol. 18, no. 4, pp. 447–458, 2009.

[69] S. Gul, B. T. Denton, J. W. Fowler, and T. Huschka, “Bi-Criteria scheduling of surgical services for an outpatient procedure center,” *Production and Operations Management*, vol. 20, no. 3, pp. 406–417, May 2011.

[70] B. Jerbi and H. Kamoun, “Multiobjective study to implement outpatient appointment system at Hedi Chaker Hospital,” *Simulation Modelling Practice and Theory*, vol. 19, no. 5, pp. 1363–1370, May 2011.

[71] M. C. Fu, “Optimization for simulation: theory vs. practice,” *INFORMS Journal on Computing*, vol. 14, no. 3, pp. 192–215, 2002.

[72] J. R. Swisher, P. D. Hyden, S. H. Jacobson, and L. W. Schruben, “A survey of simulation optimization techniques and procedures,” in *Proceeding of the 32nd conference on Winter simulation, Orlando, FL*, 2000, vol. 1, pp. 119–128.

[73] D. Gray and D. Goldsman, “Indifference-zone selection procedures for choosing the best airspace configuration,” in *Proceedings of the 20th conference on Winter simulation, Washington, D.C.*, 1988, pp. 445–450.

[74] H.-C. Chen, L. Dai, C.-H. Chen, and E. Yücesan, “New development of optimal computing budget allocation for discrete event simulation,” in *Proceedings of the 29th conference on Winter simulation, Atlanta, GA*, 1997, pp. 334–341.

[75] H.-C. Chen, C.-H. Chen, J. Lin, and E. Yücesan, “An asymptotic allocation for simultaneous simulation experiments,” in *Proceedings of the 31st conference on Winter simulation Simulation, Phoenix, AZ*, 1999, vol. 1, pp. 359–366.

[76] D. J. Morrice, J. Butler, P. Mullarkey, and S. Gavirneni, “Sensitivity analysis in ranking and selection for multiple performance measures,” in *Proceedings of the 31st conference on Winter Simulation, Phoenix, AZ*, 1999, vol. 1, pp. 618–624.

[77] S. Andradóttir, “A global search method for discrete stochastic optimization,” *SIAM Journal on Optimization*, vol. 6, pp. 513–530, 1996.

[78] J. Banks and J. S. Carson, *Discrete-event system simulation*. Prentice Hall Upper Saddle River, NJ, 2001.

[79] R. W. Eglese, “Simulated annealing: A tool for operational research.,” *European Journal of Operational Research*, vol. 46, no. 3, pp. 271–281, 1990.

[80] M. Fleischer, “Simulated annealing: past, present, and future,” in *Proceedings of the 27th conference on Winter Simulation, Arlington, VA*, 1995, pp. 155–161.

[81] F. Glover and M. Laguna, *Tabu search*. Norwell, MA: Kluwer Academic, 1997.

[82] G. E. Liepins and M. R. Hilliard, “Genetic algorithms: Foundations and applications,” *Annals of Operations Research*, vol. 21, no. 1, pp. 31–57, 1989.

[83] E. M. Manz, J. Haddock, and J. Mittenthal, “Optimization of an automated manufacturing system simulation model using simulated annealing,” in *Proceedings of the 21st conference on Winter simulation, Washington, D.C.*, 1989, pp. 390–395.

[84] M. R. P. Barretto, L. Chwif, T. Eldabi, and R. J. Paul, “Simulation optimization with the linear move and exchange move optimization algorithm,” in *Proceedings of the 31st conference on Winter Simulation, Phoenix, AZ*, 1999, vol. 1, pp. 806–811.

[85] J. Zeng and J. Wu, “DEDS (discrete event dynamic systems) simulation-optimization algorithm using simulated-annealing combined with perturbation analysis,” *Zidonghua Xuebao Acta Automatica Sinica*, vol. 19, no. 6, pp. 728–731, 1993.

[86] T. Brady and B. McGarvey, “Heuristic optimization using computer simulation: a study of staffing levels in a pharmaceutical manufacturing laboratory,” in *Proceedings of the 30th conference on Winter Simulation, Washington, D.C.*, 1998, pp. 1423–1428.

[87] M. C. Fu, F. W. Glover, and J. April, “Simulation Optimization: A Review, New Developments, and Applications,” in *Proceedings of the 37th conference on Winter Simulation, Orlando, FL*, 2005, no. 1, pp. 83–95.

[88] J. Banks, J. S. Carson, B. L. Nelson, and D. M. Nicol, *Discrete-Event System Simulation.*, 4th ed. Upper Saddle River, N. J: Prentice-Hall, 2005.

[89] W. D. Kelton, R. P. Sadowski, and D. T. Sturrock, *Simulation with ARENA*. United States: McGraw-Hill Higher Education, 2003.

[90] B. Adenso-Diaz and M. Laguna, “Fine-tuning of algorithms using fractional experimental designs and local search,” *Operations Research*, vol. 54, no. 1, pp. 99–114, 2006.

[91] T. Bartz-Beielstein and M. Preuss, “Experimental research in evolutionary computation,” in *Proceedings of conference on Genetic and Evolutionary Computation, London, England*, 2007, pp. 3001–3020.

[92] B. Bettonvil, E. del Castillo, and J. P. C. Kleijnen, “Statistical testing of optimality conditions in multiresponse simulation-based optimization,” *European Journal of Operational Research*, vol. 199, no. 2, pp. 448–458, Dec. 2009.

[93] P. Jula and R. C. Leachman, “Coordinated multistage scheduling of parallel batch-processing machines under multiresource constraints,” *Operations Research*, vol. 58, no. 4, pp. 933–947, Apr. 2010.

[94] J. H. May, D. P. Strum, and L. G. Vargas, “Fitting the lognormal distribution to surgical procedure times,” *Decision Sciences*, vol. 31, no. 1, pp. 129–148, Mar. 2000.

[95] R. H. Myers, D. C. Montgomery, and C. M. Anderson-Cook, *Response surface methodology: process and product optimization using designed experiments*, vol. 705. Wiley, 2009.

[96] F. Glover, “Tabu search-part I,” *INFORMS Journal on Computing*, vol. 1, no. 3, pp. 190–206, 1989.

[97] E. J. Yellig and G. T. Mackulak, “Robust deterministic scheduling in stochastic environments: The method of capacity hedge points,” *International Journal of Production Research*, vol. 35, no. 2, pp. 369–379, Feb. 1997.

[98] J. Zhou and F. Dexter, “Method to assist in the scheduling of add-on surgical cases-upper prediction bounds for surgical case durations based on the log-normal distribution,” *Anesthesiology*, vol. 89, no. 5, pp. 1228–1232, 1998.

[99] M. P. Hansen, “Tabu search for multiobjective optimization: MOTS,” in *Proceedings of the 13th International Conference on Multiple Criteria Decision Making, Cape Town, South Africa*, 1997, pp. 574–586.

[100] J. Caballero, R. Gandibleux, X. Molina, “MOAMP—A generic multiobjective metaheuristic using an adaptive memory,” Technical report, University of Valenciennes, France, Technical report, University of Valenciennes, Valenciennes, France, 2004.

[101] D. Jaeggi, G. Paprks, T. Kipouros, and P. Clarkson, “The development of a multi-objective Tabu Search algorithm for continuous optimisation problems,” *European Journal of Operational Research*, vol. 185, no. 3, pp. 1192–1212, Mar. 2008.

[102] E. J. Schaumann, R. J. Balling, and K. Day, “Genetic algorithms with multiple objectives,” in *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis & Optimization*, 1998, pp. 2114–2123.

[103] K. Deb, S. Agrawal, and A. Pratap, “A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II,” in *Parallel Problem Solving from Nature – PPSN VI*, 2000, pp. 839–848.

[104] M. Stigge, H. Plötz, W. Müller, and J. P. Redlich, "Reversing CRC– theory and practice," Technical report, Humboldt University, Berlin, 2006.

[105] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, Apr. 2003.

[106] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[107] J. R. Schott, "Fault tolerant design using single and multicriteria genetic algorithm optimization," Master thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1995.

Appendix: MATS source code structure

The implementation of MATS method includes over 20 classes. This section attempts to provide a description of the most important classes in order to ease the understanding and future application of the source code. Figure App.III.1 presents the class structure of MATS. These classes can be divided into four categories: multi agent tabu search, simulation, MP model, and graphical user interface and miscellaneous. The names of classes are in italic.

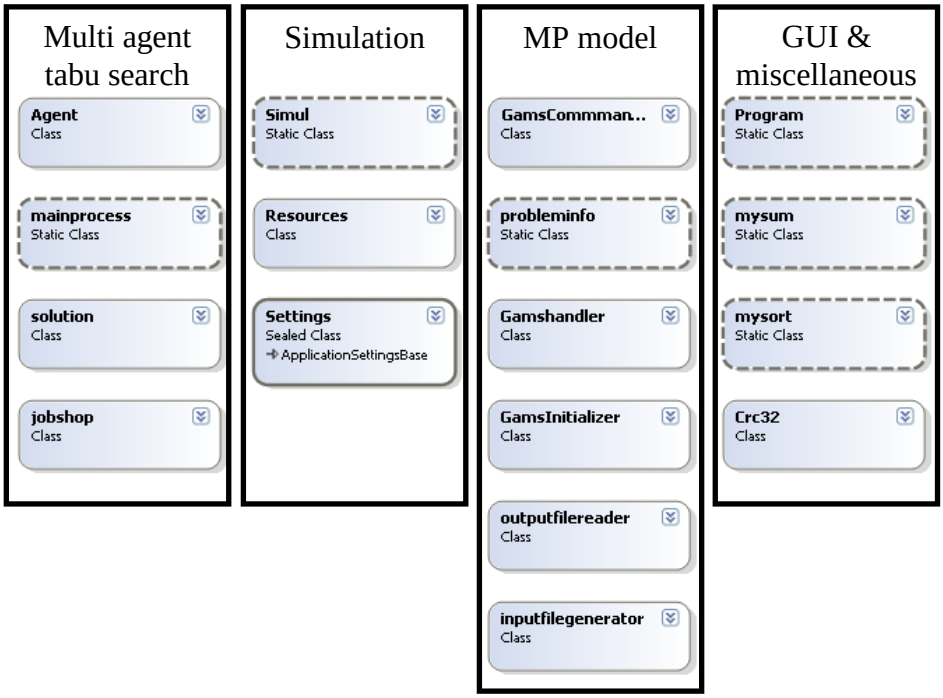

Figure App.1 Classes of MATS.

Classes in multiagent tabu search provide the tabu search main loop and its critical components. *Solution, Agent, main process, and jobshop* are the main classes in this group. *Mainprocess* is a static class that provides the main loop of the proposed tabu search. It defines and manages variables and objects involved in each step of the tabu search algorithm. This class also controls

the number of agents and the transaction of information among them. The *Agent* class defines composition of each agent in MATS and handles variables and structures within them. The *Solution* class accounts for the definition and presentation of candidate solutions in the proposed method. The detailed description of solution presentation has been provided in Section 3.3.4.2. The *Jobshop* class is the implementation of deterministic scheduling module (DSM), which has been described in Section 5.3.4.2. It provides a deterministic evaluation of candidate solutions before they are evaluated by simulation model.

Classes in the simulation category create a bridge between the proposed method and the simulation program, Arena. The *Simul* class takes solution objects as the input and returns an array which contains objective function values. Initialization and manipulation of the variables within the simulation model are done with *Simul* class. This class also controls the simulation model critical activities such as creating components, running models, and generating report.

MP model classes manage transactions between the MP model and the proposed method. The *Probleminfo* class contains the information and parameters of each test problem; the parameter values and test problem information are shared with the rest of code using this class. *Gamshandler* controls the process of creating the MP model input file, running the model, and converting the result of the MP model to initial solutions of MATS. *GamsCommand* implements all the applicable commands of GAMS software tool which is used by MATS in order to construct the input file of MP model. *Inputfilegenerator* and *Gamsintializer* generate the input file and run the MP model using GAMS. *Outputfilereader* accesses the output file following the MP model run and converts the results to the initial solutions of MATS. Figure App.I-2 presents the flow chart of the MATS.

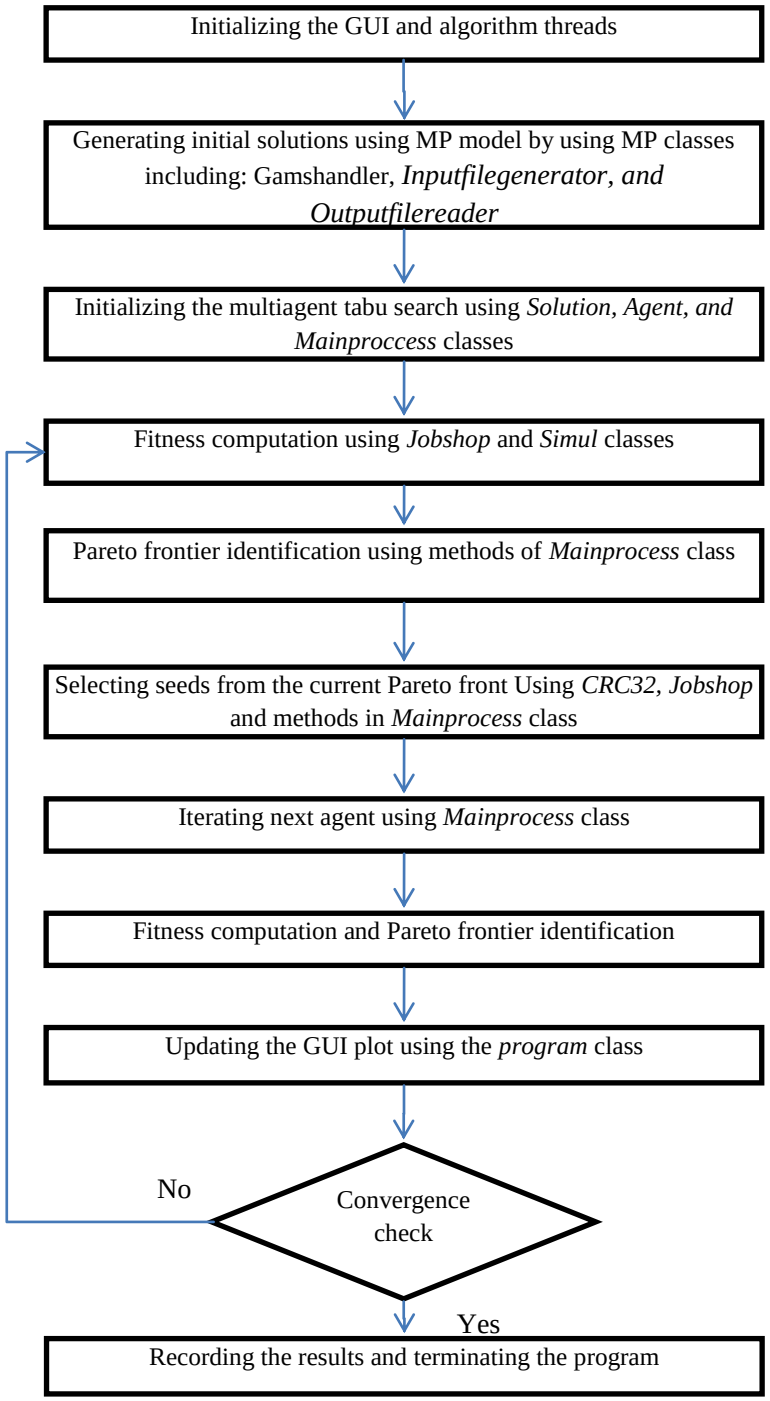

Figure App.2 Flowchart of MATS.

The *Program* class manages the multithreading in the program. The MATS code contains two process threads: GUI thread and algorithm thread. The GUI thread handles the GUI and a plot

component which presents the progress of optimization. The algorithm thread manages the MATS algorithm and its components in the background. The miscellaneous classes such as *mysort* and *CRC32* provide mathematical, ranking, and other peripheral functions which are shared throughout the program.