Novel Approaches to Radiotherapy Treatment Scheduling

Pedro Leite-Rocha, BSc, MSc

Thesis submitted to the University of Nottingham for the degree of Doctor of Philosophy

December 2011

nemo nisi per amicitiam cognoscitur

Abstract

Radiotherapy represents an important phase of treatment for a large number of cancer patients. It is essential that resources used to deliver this treatment are used efficiently. This thesis approaches the problem of scheduling treatments in a radiotherapy centre. Data about the daily intake of patients are collected and analysed.

Several approaches are presented to create a schedule every day. The first presented are constructive approaches, developed due to their simplicity and low computational requirements. The approaches vary the preferred treatment start, machine utilisation reservation levels, and the frequency and number of days in advance with which schedules are created.

An Integer Linear Programming (ILP) model is also presented for the problem and used in combination with approaches similar to the ones above. A generalisation of the constructive utilisation threshold approach is developed in order to vary the threshold level for each day according to how far it is from the current day. In addition, the model is evaluated for different sizes of the problem by increasing the rate of patient arrivals per day and the number of machines available. Different machine allocation policies are also evaluated.

An exact method is introduced for finding a set of solutions representing the whole Pareto frontier for integer programming problems. It is combined with two robust approaches: the first considers known patients before they are ready to be scheduled, while the second considers sets of predicted patients who might arrive in the near future. A rescheduling approach is also suggested and implemented. A comparison is made amongst the best results from each group of approaches to identify the advantages and disadvantages of each. The robust approaches are found to be the best alternative of the set.

Part of the work presented in this thesis has been published (or is currently under review) and presented at international and home conferences and workshops.

Papers published, submitted or in preparation:

Petrovic, S. & Leite-Rocha, P. (2008) "Constructive approaches to radiotherapy scheduling." In S. I. Ao, C. Douglas, W. S. Grundfest, L. Schruben & J. Burgstone (Editors), *World Congress on Engineering and Computer Science (WCECS'08)*, 22–24 October, pp. 722–727. San Francisco, USA: Newswood Limited. (ISBN: 978-988-98671-0-2) – Conference paper. The paper was awarded a Certificate of Merit, which was given to two of the twenty papers presented at its stream. This was the second highest award, the first being "best paper", given to only one paper in the stream.

Petrovic, S. & Leite-Rocha, P. (2008) "Constructive and GRASP Approaches to Radiotherapy Treatment Scheduling." *World Congress on Engineering and Computer Science, Advances in Electrical and Electronics Engineering – IAENG Special Edition of the*, 192–200. Los Alamitos, CA, USA: IEEE Computer Society. – Book chapter. The paper is an extension of the above, and was selected to be included in a book with the best papers presented in the conference. A total of 34 out of the 376 papers presented were chosen for this book.

Burke, E. K., Leite-Rocha, P. & Petrovic, S. (2010) *"Automated Radiotherapy Treatment Scheduling."* Handbook of Abstracts of the Annual Conference of the Operational Research Society (OR52), Royal Holloway, 7–9 September – Abstract.

Burke, E. K., Leite-Rocha, P. & Petrovic, S. (2010) *"An Integer Linear Programming Model for the Daily Radiotherapy Treatment Scheduling Problem."* – Journal paper (Under review of the Journal of the Operational Research Society - JORS)

Burke, E. K., Leite-Rocha, P. & Petrovic, S. (2010) *"Automated Scheduling to Decrease Delays in Radiotherapy Treatments."* – Journal paper (Under review of OR Spectrum)

Burke, E. K., Leite-Rocha, P. & Petrovic, S. (2011) *"Multi-objective Robust Approaches to Radiotherapy Treatment Scheduling."* – Journal paper (Under preparation)

Conferences and workshop talks:

"Scheduling of Patients on Linear-accelerator Machines" – Talk given at the Radiotherapy Workshop, Coventry, 27 February 2008

"Constructive approaches to radiotherapy scheduling." – Talk given at the World Congress on Engineering and Computer Science (WCECS'08), San Francisco, 22–24 October 2008

"Automated Radiotherapy Treatment Scheduling." – Talk given at the Annual Conference of the Operational Research Society (OR52), Royal Holloway, 7–9 September 2010

Acknowledgements

I would like to thank my first supervisor Sanja Petrovic for the opportunity of doing my PhD in the University of Nottingham. My most sincere gratitude goes to my second supervisor Edmund K. Burke, who guided and helped me in the first miles of the bumpy road of the academia path.

My deepest thanks go to the members of the staff of the radiotherapy centre of the Nottingham University Hospitals, NHS Trust, specially Allison Wallace, who was a great help and very patient when answering hundreds of questions about her work.

I would also like to thank my parents, my sister and all the rest of my family for their unconditional love and continuous support during this time spent literally an ocean apart from them.

My gratitude goes to Paweł L. Widera, Sam D. Allen, Sven Groenemeyer and Tiago Cardal Pais, who are likely the best friends I have made during my stay here. Although they are not the only ones, I could not mention all the others without exceeding the word limit of 100,000.

A big thank you also goes to the good friends of Eternal Spirit, who played a major part in helping me maintain my sanity at acceptable levels. Their contribution was critical to this work, even if they do not know it.

This work has been supported by the Engineering and Physical Sciences Research Council grant EP/C549511/1. In addition, it would not have been possible to run all the experiments for this thesis if not for the High Performance Computing facility from the University of Nottingham.

No bytes were harmed in the making of this thesis, to the best of the author's knowledge.

Contents

List of Figures

List of Tables

"He who has begun has half done. Dare to be wise; begin!"

Quintus Horatius Flaccus (65 BC–8 BC)

Chapter 1

Introduction

Nowadays, resources are becoming more and more constrained in nearly every professional field of society, increasing the necessity for efficiency. Specifically in healthcare, with the increasing number of patients, efficient use of resources is critical (Alexopoulos et al., 2008).

Cancer incidence is increasing within the general population, specifically in the United Kingdom. The Department of Health, UK, has released several publications establishing strategies to improve cancer treatment, properly tackle the disease, and set new waiting time targets for cancer patients (Department of Health, 2000, 2004; Williams, 2008). Since then, research on decreasing treatment waiting times has intensified. Even though cancer care has improved recently within the UK, radiotherapy capacity is still an important factor that has not received an adequate amount of attention (Dodwell & Crelin, 2006). Several audits conducted in the UK show that the waiting times for cancer treatment are not yet satisfactory (Board of the Faculty of Clinical Oncology, 1998; Spurgeon et al., 2000; Ash et al., 2004; Summers & Williams, 2006; Drinkwater & Williams, 2008; Williams et al., 2008). New targets set by the Department of Health (2007), a program devised to enhance the effectiveness of cancer treatment in the country, make radiotherapy scheduling a very important problem.

Outpatient scheduling (scheduling of patients who are not hospitalised) has been the subject of many studies in many cases and scenarios in the literature, ranging from clinics

(Alexopoulos et al., 2008; Moore et al., 2001), ambulatory (Cayirli et al., 2006) and radiology services (Lev et al., 1976), surgery scheduling (Gerchak et al., 1996) and others.

The radiotherapy treatment scheduling problem shares a few common characteristics with the other outpatient scheduling problems normally found in the literature, such as:

- A stochastic arrival of patients;
- Different levels of urgency of patients, where emergency and urgent patients may need to be treated immediately;
- The objective of optimising some measure of the quality of service, usually a function of the time patients spend waiting before being seen by a doctor;
- Patients may not need to be informed of time of their appointment immediately after requesting it, making it possible to wait for the accumulation of patients before generating a schedule (however, this is not frequently considered when scheduling primary care patients);
- Patients do not show up or cancel their appointments (although this becomes less frequent when the severity of the case increases).

However, there are some key differences as well. Outpatient scheduling problems usually consist of scheduling a single doctor appointment for a patient, which often has a stochastic duration. When follow-up sessions exist, they may be scheduled on the fly. Radiotherapy treatment scheduling consists of booking several sessions of deterministic duration such that consecutive sessions of the same patient are scheduled a pre-determined number of days apart. If a treatment is missed or cannot be scheduled, the treatment could be compromised (Ragaz et al., 2005).

These features make the problem not only hard to solve, but also hard to model. Developing a good decision support tool (a piece of software) to help the radiographer in quickly generating high quality schedules is a worthwhile effort, as it provides a series of benefits:

Reduced expenses: A common way for radiotherapy centres to increase the quality of their schedules is acquiring additional equipment and/or increasing the number of working hours or size of the radiotherapy staff. A decision support tool might be able to find a better schedule without need of these extra resources.

Decreased radiographer workload: In many centres, the schedule is generated by hand by one or more of the radiographers. This is a very time consuming task which may keep radiographers away from their normal obligations. By using a piece of software for creating schedules, it is possible to relieve the radiographers from this chore.

Increased reliability of treatments: Good quality schedules ensure that patients receive their treatment without excessive waiting time, thus increasing the possibility of treatment success. This is specially true in radiotherapy, because of the adverse effects caused by prolonged delays (Mackillop, 2007).

1.1 Collaboration

The work presented in this thesis has been conducted in collaboration with the radiotherapy centre located at the City Hospital Campus of the Nottingham University Hospitals, NHS Trust (2011), which treats an average of 2,400 patients per year.

A total of 31 meetings were made in the period from March 2006 to March 2011 with members of the hospital staff. They were arranged in order to help us properly understand the problem and its characteristics, as well as to show the progress achieved and the state of the developed software at the time of each meeting.

The radiotherapy process considered in the hospital uses external beam radiation and is consisted of two main phases: pre-treatment and treatment. In the pre-treatment phase, the patient goes through a series of examinations to determine a series of parameters, such as the dose of radiation, number of beams, angles between them, etc. These examinations must be scheduled in a specified order, and there might be a minimum required time interval between some pairs of examinations. For example, depending on the site of the tumour, the patient might require a moulded shell to help them remain immobile during the delivery of the radiation. It usually takes the staff one hour to produce the shell, but an additional 24 hours are required before the shell can be used.

In the treatment phase, the patient attends a number of sessions on the radiation machines in order to receive the prescribed radiation dosage. Patients may require from 1 up to 37 sessions, which must take place on consecutive days for most patients. However, a larger interval may be necessary in a few cases. The duration of each session depends on the tumour site and if more than one site are being treated at a time, meaning it can vary greatly from one patient to the other. In addition, the first session of each patient is longer than the others so that the patient's data can be fed into the machine, and the patient can get used to the equipment and to the staff. Furthermore, there are restrictions on which days of the week each session can be scheduled on. These will be explained in detail in the further chapters.

The work presented in this thesis focuses on the scheduling of the treatment phase. The greatest challenge in radiotherapy treatment scheduling resides in the uncertainty of the number and type of future patients. Although finding the (locally) optimal schedule for patients who have already arrived may not be an issue in some cases, it may be impossible to guarantee a globally optimal schedule, since it is usually impossible to predict exactly the number and type of patients that will arrive in the future. Therefore, schedules must be created in such a way as to minimise the waiting time of patients who already have arrived, while leaving enough space for patients who must be treated urgently and are still to arrive.

1.2 Research Objectives

The research presented in this thesis focuses on developing and evaluating models and algorithms used to automatically create radiotherapy treatment schedules. The main objectives of this thesis are:

- Review the literature available not only on patient scheduling, but also on machine scheduling in order to identify common characteristics with the problem tackled in this work.
- Analyse the data supplied by the hospital and try to identify patterns and other details which might help in designing approaches for creating schedules.
- Develop robust methods capable of creating a real-world schedule in an acceptable amount of time.
- Analyse the performance of the new methods by a detailed and well structured set of experiments.
- Create a graphical interface for the proposed methods to be used by the radiotherapy centre of the hospital staff at the end of the project.

In order to achieve these objectives, three main hypothesis are elaborated and will be tested in this thesis.

Hypothesis 1: *An Integer Linear Programming approach can improve the schedule created by constructive approaches.* An approach which finds the optimal schedule considering for patients who have already arrived may improve the schedule found by constructive approaches. However, this improvement may or may not result in a better schedule overall, since it does not include future patients. This will be investigated in this thesis.

Hypothesis 2: *Robust advanced approaches can improve the quality of the schedule found by simpler myopic approaches for the radiotherapy treatment scheduling problem.* Myopic approaches are those that consider only patients who have already arrived and make no attempt at predicting patients who will arrive in the near future. Robust approaches which take such patients into consideration have found good schedules for similar problems (Sadki et al., 2010b).

Hypothesis 3: *Rescheduling approaches can find schedules of higher quality than robust approaches at an acceptable number of rescheduled patients for the radiotherapy treatment scheduling problem.* Allowing the algorithm to change schedules already made can also increase the quality of the resulting schedule. Although there is a downside that a number of patients will need to be rescheduled, this number might be acceptable depending on the quality of the generated scheduled.

1.3 Thesis Structure

The contents of the thesis are structured as follows:

- Chapter 2 examines different categories of patient scheduling problems. The majority of the literature relevant for this work is discussed. Papers are classified according to the methods used.
- Chapter 3 details the problem approached as it is currently faced by the hospital. Two sets of data are given by the hospital staff, and an analysis of the data supplied is also performed in order to help the author and the hospital staff in gaining insight into the problem. This analysis also helps in the interpretation of some results and in the design of the more advanced methods presented in this thesis. Furthermore, an algorithm is proposed to combine these two data sets in order to help generate a combined data set and prepare for generating problem instances.
- Chapter 4 presents the first approaches developed to tackle the problem and their evaluation with sets of experiments. The main goal of this chapter is to investigate the quality of schedules produced when using simple methods that are easy to implement. If the more advanced methods proposed in the future chapters cannot improve the results from the simple methods, there will be no point to them. Four approaches are investigated in this chapter. The first approach defines a preferred start date for each patient and tries to schedule each patient as close as possible to their preferred start. The second approach determines an utilisation threshold for a category of patients, such that when the utilisation on a machine reaches this threshold for a certain day, no more sessions of patients of this category are scheduled on that machine for that day. The third approach limits the number

of times per week when schedules are created for patients of each category in order to allow the accumulation of unscheduled patients and increase the search space. The fourth approach consists of postponing the creation of patients' schedules until their release date becomes closer. The algorithm developed to generate the problem instances used in this work is also presented in this chapter.

- Chapter 5 presents an Integer Linear Programming model developed to represent the problem of scheduling patients for a given day, and shows how this model can be used with a commercial solver. The model is evaluated when used in conjunction with the third and fourth approaches presented in Chapter 4. An extension of the utilisation threshold approach, where the threshold varies with time, is also developed and investigated. An analysis is made about the correlation of the values found for each objective function when using different values for the utilisation threshold. Furthermore, the variable threshold approach is evaluated with a larger rate of patient arrival, a larger machine set, and changes to the scheduling policy to increase the quality of created schedules.

- Chapter 6 introduces a Pareto-type multi-objective algorithm developed for integer programming problems capable of finding a set of solutions representing the whole Pareto frontier. Two robust approaches are proposed in order to achieve more robust schedules. The first considers patients who have already arrived but are not scheduled yet due to their release date still being far away. The second approach consists of trying to predict which patients will arrive in the near future and consider these predicted patients when creating a schedule. In addition, a rescheduling approach is also proposed, where already created schedules of patients can be modified if their start day is at least a number of days away and the schedule has not been modified yet or has been modified less than a specific number of times. Finally, a comparison is made including the best combination of approaches and parameters found in each chapter to determine which approach should be recommended to the hospital.

- Chapter 7 gives a summary of the conclusions reached throughout this thesis. The impact of this work, as well as an analysis of the hypotheses presented are also given. Future research directions are suggested in the end of this chapter.

"All things I thought I knew; but now confess
The more I know, I know, I know the less."

John Owen (1616–1683)

Chapter 2

Literature Review

There is a comprehensive amount of literature on outpatient scheduling. To the best of the author's knowledge, the earliest work is that of Bailey (1952), where queue theory is used to minimise the waiting time of patients and the idle hours of consultants in a hospital appointment system. Since then, outpatient scheduling has appeared in many cases and scenarios in the literature. Although outpatient scheduling is a problem frequently discussed in the literature, only a few papers deal with the problem of scheduling radiotherapy treatments.

In order to better understand the available literature, a classification scheme based on the works by Cayirli & Veral (2003) and Gupta & Denton (2008) is presented in this chapter. Section 2.1 describes a classification according to the environment the problem deals with. Section 2.2 lists the most common characteristics in outpatient scheduling and how they are more commonly found in each environment. The most common performance measures are described in Section 2.3. Section 2.4 briefly introduces the problem approached in this thesis. Common methods used to tackle outpatient scheduling problems are introduced in Section 2.5. Papers which investigate methods to predict demand and model patient arrival are presented in Section 2.6. Finally, Section 2.7 makes a comparison between outpatient and production scheduling problems.

2.1 Outpatient Scheduling Environments

Outpatient scheduling systems can be classified according to the environment for which they are designed. Three main types of scheduling environments can be defined (Gupta & Denton, 2008):

Primary care: Patients arrive to a clinic or hospital, usually to see a general practitioner (GP);

Specialty care: Patients are usually referred from primary care to a specialty clinic in order to receive treatment specific to the patient's diagnoses;

Elective surgeries: Scheduling of chosen and planned in advance surgery procedures in operating rooms.

These environments are described below in more detail.

2.1.1 Primary Care

This refers to clinics and hospitals which deal with patients when they first come to see a doctor, usually a general practitioner (GP). Scheduling systems designed for primary care commonly have a fixed number of slots with a pre-determined length. If patients require a session longer than the length of a slot, more than one slot can be assigned to them.

In the majority of environments, patients call in advance to schedule an appointment. The scheduler then books a time and day for the patient's appointment. There may also be walk-in patients, who must be accommodated sometime during the day. Some of these may be emergency or urgent patients and may need to be seen immediately.

Although patients are booked in fixed length slots, the actual duration of sessions can be stochastic. Some appointment systems can also consider both stochastic and deterministic session durations at the same time, where the duration of the first session of a patient is stochastic and the duration of follow-up sessions are deterministic when sessions are mostly routine check-ups (Gupta & Denton, 2008).

2.1.2 Specialty Care

Specialty clinics focus on specific, often complex, diagnoses and treatments. For the majority of specialty clinics, patients must be referred to them by a GP before they can ask for an appointment. However, a few types of clinics (such as paediatric and obstetric clinics) are designed to accept patients without need for a referral.

Session durations are usually deterministic in this environment. They may also have a large variation due to different diagnoses and, therefore, fixed length slots are not usually adopted. Instead, each session is booked to use only the amount of time it requires.

Resources in specialty clinics are often very expensive, either in the form of equipment and machinery, or in the form of specialised doctors. For this reason, achieving high levels of utilisation in these environments is highly desired in order not to waste resource time. Patients commonly have different levels of urgency, where normal patients can be scheduled well in advance and emergency or urgent patients can arrive with short notice and need to be treated immediately. The main challenge in this environment usually resides in reserving enough capacity for patients of high urgency, while maintaining a high utilisation of the resources.

2.1.3 Surgery Scheduling

The duration of surgeries is not known a priori. Complications can occur during surgery, which can greatly increase their duration. This stochastic duration with large variation makes surgery scheduling one of the most complex patient scheduling problems. Cardoen et al. (2010) and Magerlein & Martin (1978) present reviews of scheduling algorithms tailored for this environment.

Similarly to primary and specialty care, there are different levels of urgency for surgeries. Elective surgeries can be scheduled with large number of days in advance, as they do not need be performed immediately. Emergency and urgent patients can arrive at any time and need to be treated right away. Such surgeries are not booked, but accommodated during the day. In order to deal with emergency procedures, time in operating rooms can be reserved. It is also possible to try to spread the remaining time available equally throughout the week.

Surgeries require a greater variety of resources to be allocated to them when compared to the primary and specialty care environments. In addition to the operating room, where the actual surgery will take place, they may require specialised equipment to be taken to the room (e.g. pressurised environment for hyperbaric surgery), and different types and levels of specialisation from the nursing staff. Other specialised equipment may not be mobile and may be present only in a small set of rooms. These can result in complex resource allocation constraints. Surgery patients may also require post-operative care before being discharged, which may call for additional specialised equipment, follow-up appointments and/or home care visits.

2.2 Characteristics of Outpatient Scheduling

In this section, outpatient scheduling problems are further classified according to three characteristics: the mapping of the patient arrival process, the definition of service times, and the existence of patient and/or provider preferences. The most common types in each classification for the environments described above are also identified.

2.2.1 Patient Arrival

Appointment systems are designed for a specific mapped patient arrival process, which can be different from the actual process. For example, in the environment of most primary care clinics, each decision is made at the moment that a patient calls asking for an appointment. In this case, the mapped patient arrival process is the same as the actual process. For specialty care, it is common to wait for patients' booking requests to accumulate before creating a schedule, and the mapped process can be considered as blocks of arrivals at regular intervals.

Surgical appointment systems may consider an arrival processes with two steps. In the first step, when a patient first asks for an appointment, a decision is made to establish a time window (e.g. a week) in which the surgery will take place. This decision can be made as soon as the request is received, making the mapped arrival for this decision the same as the actual arrival, similarly to most primary care environments. In the second step, the precise date and time for the appointment are decided. This is usually done once all elective appointment requests for that time window have arrived. For this decision, the mapped arrival process consists of blocks of arrivals which should be scheduled in the time window.

The patient arrival processes can be classified in one of the following groups (Gupta & Denton, 2008):

Unit process: Mapped and actual arrival are considered to be the same. Patients are considered one at a time and inter-arrival times are considered random. For these cases, patient arrival is usually modelled as a Poisson process. This arrival process is most common to primary and specialty care environments.

Periodic process: Time is divided in periods of equal length. At the end of each period, schedules are created for all patients who arrived during this time period. Inter-arrival times are considered to be constant, but the number and type of patients who arrive in a time period may be random. This process is usually present in specialty care and surgery scheduling environments.

Single batch: Schedules are created only after observing all demand for a time period. The inter-arrival times are irrelevant in this case. This can be seen as a special case of the periodic process where the time period covers the whole planning horizon. However, the methods used to tackle each of them can be very different, hence they are treated as two different processes. As periodic process, single batch arrival is usually present in specialty care and surgery scheduling environments.

Additional factors which may be present regarding the arrival process include unpunctuality of patients, unpunctuality and interruption levels of doctors, presence of no-shows, walk-ins and/or companions (Cayirli & Veral, 2003).

2.2.2 Service Times

Service times are considered as the time during which a patient is claiming the consultant's attention, or at least preventing the consultant from seeing the next patient (Bailey, 1952). They may be deterministic, stochastic or a mixture of both. For example, as previously stated, in a primary care environment it is possible to consider the first session of each patient as having random duration and the follow up sessions as having deterministic duration. Generally, authors make the assumption that service times are independent from the arrival patterns. However, this may not be always true in practice. For example, doctors may try to decrease service times if there are many patients in the waiting area.

In the specialist care environment, service time is usually deterministic. However, it can have a large variation depending on patients' diagnoses (Lev et al., 1976; Vermeulen et al., 2007).

In the surgery scheduling environment, service times are usually stochastic and can have a large variation, specially when dealing with emergency patients. However, in the service time of elective surgeries, the variation of the service times is not as large (Cardoen et al., 2010).

2.2.3 Patient and Provider Preferences

Patient and provider preferences can be modelled as constraints or as objective functions. Patients may prefer to be treated as soon as possible or on a specific day of the week and/or at a specific time. They usually do not mind the wait between the request and the appointment if the later is at a more convenient time for them.

Patients may also have preference for a particular consultant, specialist or surgeon. For example, a patient may prefer to be treated by the general practitioner (GP) that the patient is registered with or by a specialist who has been referred to by a friend or another doctor.

Providers may also impose restrictions on the way their time is spent. For example, doctors in primary care may restrict the number of physical exams they perform per day, or reserve slots for patients who are registered with them. Surgeons may also specify on which time

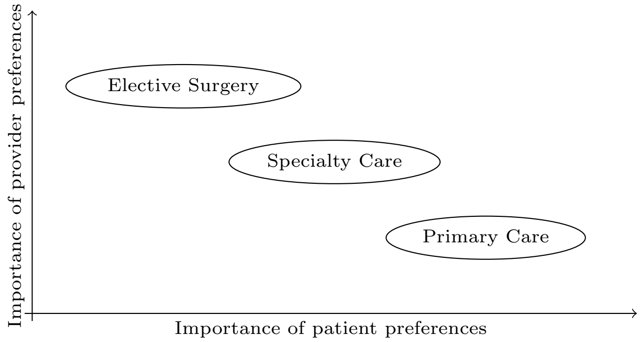


Figure 2.1: Importance of provider preferences versus patient preference (Gupta & Denton, 2008)

of the day and days of the week they would prefer to perform surgeries. Furthermore, providers may allow a certain amount of overtime or no overtime at all.

Patient and provider preferences have different degrees of priority depending on the scheduling environment. Patient preference usually has a higher priority in the primary care environment, while provider preference has a higher priority in specialist care and surgery scheduling. This is illustrated in Figure 2.1.

Some patient preferences may cause the mathematical model of the problem to be intractable. They also often indicate that optimal policies are not trivial or simple to implement (Gupta & Wang, 2008).

2.3 Performance Measures

There are many possible performance measures used in the literature to evaluate appointment systems. Most of them use a function of the time patients spend waiting for their appointment or of the time doctors remain idle. Five main types of performance measures are enumerated by Cayirli & Veral (2003) and detailed below:

Time-based measures: cover mainly the waiting time of patients and the idle and overtime of doctors and resources. Patient waiting time can be further classified as *indirect (or virtual)*, defined as the time between the request for an appointment and the time scheduled for the appointment, or *direct (or captive)*, defined as the time between the scheduled time of the appointment and the time the consultation actually starts. In situations where direct waiting time is calculated and the arrival time of patients is also considered, a common approach is to use the *"true"* waiting time, defined as the difference from the time the consultation starts to the latest time between the arrival of the patient and the scheduled appointment time. This excludes the delay caused by tardiness of late patients and the voluntary waiting time of early patients, since both these delays are the patient's own fault and not a consequence of the appointment system.

Idle time can be defined as the amount of time a resource is available but not used. Overtime can be seen as the extra time after normal closing hours that a resource is kept busy.

Cost-based measures: are generally a linear mapping of the time-based measures to monetary cost. However, it should be noted that a schedule where many patients have small waiting times may be better than a schedule where one patient has excessive waiting, even if the total waiting time in both schedules is the same (Klassen & Rohleder, 1996). When considering more than one performance measure, it is usually enough to establish a relationship between the costs of each measure in order to make a decision. For example, if the objective is to minimise the direct and indirect waiting times of patients, it is possible to provide the ratio of the cost of direct waiting over the cost of indirect waiting. Estimating this ratio may be easier for the service provider, instead of finding the actual monetary costs of each.

Congestion measures: may also give an idea of how "good" a system is. Examples of congestion measures include the average number of patients in the queue in a given time period.

Fairness measures: are usually considered as the degree of uniformity of performance across patients. Examples of fairness measures include measures of the average direct waiting time for each patient in the order of appointments (average waiting time of the first patient, of the second, etc.) (Bailey, 1952), and measures of the uniformity of patient direct waiting times, proposed by Yang et al. (1998).

Other measures: may include the average number of patients treated in a clinic session, resource utilisation, and any other measure which does not fit into the classifications above.

2.4 The Radiotherapy Treatment Scheduling Problem

The problem approached in this thesis can be classified as a specialty care problem with the arrival of patients following a periodic process and deterministic service times. Each patient has a number of sessions which must be scheduled on the radiation machines, such that consecutive sessions of the same patient are scheduled on consecutive days. This applies to the majority of patients, but there are also other requested frequencies of sessions. A minority of patients may require that consecutive sessions are scheduled either further apart (e.g. every other day) or closer to each other (e.g. three sessions per day). The session's duration of patients may differ from each other, and the first session of each patient is slightly longer than the others. In addition, there are restrictions on which days of the week each patient can have their first session.

Four measures of quality (objective function) are considered: three are based on waiting time targets established by UK governmental and academic bodies, while the fourth aims at decreasing the waiting time in general and increasing fairness. The problem is defined in more detail in Chapter 3.

2.5 Methods Found in the Literature

Many methods can be used to tackle patient scheduling problems. The problem can be treated myopically, where an exact method is used to optimise the schedule of currently available patients (Conforti et al., 2008). If an exact method is computationally too expensive, a heuristic may be used in its place (Kapamara & Petrovic, 2009). For problems where patients have different priorities, using such schedule on its own is not a good practice, as it often causes the earliest appointment slots to be always used first, leaving no space for future high priority patients. However, this effect can be counteracted by combining the myopic method with a number of other approaches. For example, it is possible to implement resource reservation in the form of constraints, so that reservations must always be respected (Vermeulen et al., 2009), or in the form of an objective of minimising reservation violation. Demand forecasting can also be used to try to predict which patients may arrive in the near future, so that these patients are considered when creating the schedule (Sadki et al., 2010b). Another approach is to maximise the number of patients scheduled in a finite horizon, e.g. one week, and delay the scheduling of patients who cannot start within this horizon (Conforti et al., 2010a).

Simulations are also commonly used to model patient scheduling problems. They can be used to better study each specific case, identify bottlenecks, as well as estimate the effect of proposed changes on the scheduling policy, and evaluate different scheduling algorithms (Kapamara et al., 2007). The following subsections discuss these methods.

2.5.1 Exact Methods

The main advantage of exact methods is that they can guarantee the optimality of the solution found. In patient scheduling problems, this often translates as a myopically optimal schedule, i.e. a schedule which is optimal considering only patients who have already arrived. As previously stated, exact methods can be combined with different approaches in order to consider future patients as well.

Integer and mixed integer programming (Nemhauser & Wolsey, 1989) models are commonly used to approach this or other similar problems. Conforti et al. (2008) define mathematical models for the scheduling of radiotherapy treatment. The objective in their proposed model is to schedule as many patients as possible in a short period of time (e.g. one week). They consider a block system, where a workday is split into a fixed number of time blocks/slots. In a subsequent paper, the same authors extend the model to take patient availability into account, and run more extensive experiments with real world data (Conforti et al., 2011). Conforti et al. (2010a) then consider a non-block system, where the session time may vary from one session to another. They observe that uniform appointment blocks do not represent real workload properly, since the treatments can take either more or less time than the chosen block time. Conforti et al. (2010b) generalise the model for other problems, not only radiotherapy, where the objective is to maximise the weighted number of patients admitted during one week. However, the models do not consider all constraints present in real-world radiotherapy scheduling, such as machine eligibility, release dates different from the booking requests and patients who require multiple sessions per day.

Kaandorp & Koole (2007) present a patient scheduling problem in a medical clinic. Patients have stochastic service times and call in advance to arrange an appointment, which can be scheduled to a specific time slot, such that more than one patient can be assigned to the same slot. The goal is to design an algorithm which decides the time of an appointment at the time the patient calls in order to minimise three objective functions: the mean waiting time, idle time and overtime (referred to as “tardiness” in the paper). A local search is proposed and the authors prove that it finds an optimal solution by proving that the objective function is multimodular.

2.5.2 Heuristics and Meta-heuristics

In the case where the problem instances are too large for exact methods, heuristics and meta-heuristics can be used instead. As with exact methods, they should also be combined with different approaches to consider future patients, such as resource reservation or demand forecasting.

Sadki et al. (2010a,b) present a good example of using an efficient myopic heuristic, while including future patients predicted in a rolling horizon in order to achieve more robust schedules. The first paper introduces a three-step heuristic based on mixed integer programming to balance the bed load in a chemotherapy ambulatory care unit. The heuristic is used myopically to find a good schedule considering patients who have already arrived. In the succeeding paper, the authors use the same heuristic, but add predicted possible future patients to the problem instance. They conclude that including predicted future patients can result in more robust schedules, which greatly improve the quality of schedules created in the future. An extension of this approach is presented in Chapter 6.

Patrick et al. (2008) investigate a patient scheduling problem in a diagnostic facility. The problem is modelled as a discounted infinite horizon Markov Decision Process, and an equivalent linear program (LP) is solved via approximate dynamic programming. Although this approximation may lead to sub-optimal solutions, the authors highlight that the proposed method generates better schedules than the method currently used in practice. This paper is a good example of stochastic modelling applied to healthcare. The authors conclude that it might be possible to adapt the approach to a radiation treatment problem, but a complication that will arise is that the method is tailored for problems where only one appointment is scheduled for each patient, while radiation treatment requires multiple appointments at specific time intervals.

Genetic algorithms (GAs) (Reeves & Rowe, 2002) are also used to tackle patient scheduling problems. Podgorelec & Kokol (1997) propose a GA for patient scheduling for a problem with a single batch patient arrival, where machine learning is used to determine the GA parameters. They consider patients as actors, therapies as activities, and therapists and therapeutic devices as two different types of resources. A novel multi-dimensional chromosome is used to represent a solution, where each dimension corresponds to either a position in the order of activities, an actor, or one type of resource. Methods for generating the initial population, crossover, mutation and selection are proposed in such a way as to maintain solution feasibility. After the investigation of one case study, the authors conclude that the proposed method is very effective, and that by using actors, resources and activities, the representation of other problems with complex constraints should pose no difficulty. Although this algorithm has been applied to a problem with single batch patient arrival, it could be easily adapted to a problem with periodic patient arrival. If not combined with another approach, such as machine reservation or rolling horizon, it would likely result in very full schedules with little or no room for higher priority patients who arrive with short notice.

Vermeulen et al. (2007) present an adaptive algorithm for scheduling patients on a CT-scan. Patients are divided in groups according to their urgency and other characteristics, where urgent patients have a much shorter time window to get their scan than other patients. The algorithm makes reservations for each type of patient and adaptively modifies the reserved slots when they are not used by non-urgent patients. Results for different scenarios of patients arriving per week are also presented by the authors in a succeeding paper (Vermeulen et al., 2009). This is a good example of resource reservation which is updated as time goes by depending on the quantity of the resource available in the short term.

Kapamara & Petrovic (2009) present a steepest hill climbing method for a radiotherapy patient scheduling problem found in The Arden Cancer Centre radiotherapy department in Coventry, UK. The schedule is first generated by constructive heuristics, where a different dispatching rule is used for each stage of pre-treatment. This schedule is then improved by a hill climbing heuristic which tries to bring each appointment forward to the earliest day possible. In order to evaluate the algorithm, the simulation proposed by Kapamara et al. (2007) is used. The authors find that the hill climbing method is capable of improving the schedule for palliative and radical patients at the cost of slightly worse schedules for emergency patients. The algorithms consider only patients who have already arrived, which often results in a very full schedule with little or no room for higher priority patients who arrive with short notice.

Petrovic et al. (2009) develop a genetic algorithm (GA) which considers both pre-radiotherapy and radiotherapy treatment. The algorithm aims at minimising the normalised sum of two objectives: the average indirect waiting time of patients and the average delay of patients currently being scheduled with respect to their waiting time targets. In a subsequent paper, Petrovic et al. (2011) develop two additional GAs to give a higher priority to emergency patients. The first is called KB-GA and has tailored initialisation, crossover, mutation and selection procedures that implement this higher priority. The second is called Weighted-GA and uses weights with the same purpose. In both papers, the GA's are evaluated in the simulation model created by Kapamara et al. (2007). The authors find that the KB-GA performs better on the experiments. However, the algorithms presented are used only myopically, and no attempt is made at resource reservation or at predicting future arrival of patients. As said previously, this is not ideal and results in very full schedules with little or no room for higher priority patients.

2.5.3 Simulation

As previously stated, simulation is often used to get a better understanding of the problem and to evaluate approaches. Kapamara et al. (2007) develop a discrete-event simulation model for a radiotherapy scheduling problem present in the Arden Cancer Centre, UK. The aim of this paper is to gain a better understanding of scheduling process used in the clinic, and to identify bottlenecks. Their experiments suggest that the largest bottlenecks are *linacs* (linear accelerator machines used to deliver the radiation). In addition, doctors are required for some steps of the treatment, but are present only at limited times at the centre, which causes a sporadic crowding on doctor queues. This model is then used by Kapamara & Petrovic (2009) and Petrovic et al. (2009, 2011) to evaluate their proposed scheduling methods.

Proctor et al. (2007) propose a discrete-event simulation model for the radiotherapy department of the Arden Cancer Centre, UK (however, it was called Walsgrave Hospital at the time the paper was written) in order to investigate the effects of changes in service demand and how to account for them. The authors analyse two strategies to decrease patient waiting times: 1) acquiring more equipment, such as a simulator and/or linac and 2) changing the working policy, such as not requiring radiographers to treat the same patients for all sessions, extend working hours of the linacs. They conclude that both strategies are able to achieve similar levels of improvement.

Cayirli et al. (2006) investigate the effect of different sequencing rules in scheduling an ambulatory care service by using two types of rules: sequencing rules, which define the order in which patients are scheduled in appointment slots, and appointment rules, which define the number of patients assigned to each slot and their length. To evaluate each set of rules, a simulation model is used with real-world data from a healthcare clinic in a New York metropolitan hospital. The authors use patient and doctor-based measures to evaluate each combination of rules and shows that sequencing rules have a much higher impact on performance than appointment rules.

Lev & Caltagirone (1974) categorise the problem of patient scheduling in a diagnostic radiology department as a classic job shop machine scheduling problem, and develop a discrete event simulation model of patient flow. The model is used to evaluate the performance of seven different scheduling rules according to 4 performance measures: waiting time prior to examination, total time in the system, distributions of waiting and total times, and the number of patients in the system at the end of working hours. The two rules prioritise the patients in the queue based on their expected session duration for a resource achieve the best results. The authors recommend one of these two rules. However, they acknowledge that, of the evaluated rules, the two best are the only rules which would require a computer to perform the scheduling (the other rules could be performed manually).

2.6 Demand Estimation and Modelling

Work has also been conducted developing methods to estimate demand and properly model the problem. To the best of the author's knowledge, one of the first works involving simulation and patient scheduling is presented by Fetter & Thompson (1966), where an outpatient simulator is developed for the primary care scenario. The main objective is to study the effect of certain variables, such as patients' arrival pattern and number of no-shows, on patients' waiting time and doctors' idle time.

Thomsen & Nørrevang (2009) introduce a model to predict utilisation levels in a radiotherapy department in Denmark. The model is motivated by a recent adoption of different waiting time targets for different types of patients. Based on average duration of sessions, number of sessions per day and number of new patients per day, two utilisation curves are calculated: 1) the Maximum Booking Curve (MBC), which determines what should be the maximum utilisation on a given future day in order to have enough space for future patients of short waiting times, and 2) the Lower Limit Curve (LLC), which determines the minimum utilisation on a given future day so that the machine achieves 100% utilisation when that day arrives. These two curves are shown to the radiographer via MS Excel in order to help him/her schedule the patients who have arrived on the current day. The schedule is then constructed by hand by the radiographer, who uses the predicted curves as visual guidance. The authors conclude this method improves the transparency of the booking process and is a good method for managing a booking system where patients have different waiting time targets. Although it seems an interesting method which is being used in practice, no reports are given on whether or not the targets are being met. In addition, the method still relies on the radiographer to do the booking by hand.

Thomas (2003) presents a model based on a Monte-Carlo distribution to calculate the percentage of spare capacity required to keep waiting times to treatment short. He analyses the outcome of the model if some parameters are changed, such as no treatment on bank holidays, the number of machines, etc. He concludes that, in order to meet waiting time targets of high priority patients, it is necessary to aim at a 90% average utilisation.

Alexopoulos et al. (2008) propose a modelling strategy for patients arrival in community clinics. They suggest that the usual modelling methods are not very precise (those are using a Poisson process for modelling arrival of unscheduled patients and a normal distribution for calculating the tardiness of scheduled patients). They perform experiments with several models and distributions, and conclude the Johnson CDF (Johnson, 1949) has a better fit than the normal distribution for the problem.

2.7 Comparison with Production Scheduling

Some similarities can be identified between outpatient and production scheduling. In both types of problems, the objective is often to schedule a number of tasks (patient appointments or job operations) to a limited number of resources in order to optimise a function of the completion time of each task. Depending on the characteristics of the problem, it is possible to consider outpatient scheduling as a classic production scheduling problem, for example, as done by Lev & Caltagirone (1974). In such cases, an algorithm designed for production scheduling can be adapted and used to find a schedule of good quality.

Kapamara et al. (2006) analyse the radiotherapy patient scheduling problem, considering both pre-treatment and treatment, concluding that it is similar to a dynamic stochastic job shop problem with recirculation. Although this might be true for the pre-treatment stage, modelling the treatment stage investigated in this thesis as a production problem is not straight forward. The main difficulty lies in the constraint that treatments must be held on days with specific intervals between them. It was not possible to find a particular production scheduling problem reported in the literature which can be directly translated into this constraint. A possibility is to interpret it as a special type of *no-wait* job shop problem, where a no-wait constraint indicates that an operation of a certain job must take place immediately after the previous operation. In production scheduling, this situation can arise when a wait would cause the material to cool down, and it is important that the material remains in the current temperature for the next operation (Hall & Sriskandarajah, 1996; Pinedo, 2008).

Although it is possible to model the problem investigated in this thesis as a dynamic flexible job shop with recirculation, machine eligibility and a special type of no-wait constraint, most of the literature on production scheduling considers only one of these constraints and adapting their proposed methods to the problem investigated in this thesis would likely be ineffective or even impossible. Furthermore, in Chapter 5, a mathematical programming model tailored for radiotherapy treatment scheduling is presented. The model requires very little computational resources and is able to find the myopically optimal solution. Therefore, modelling the radiotherapy treatment scheduling problem as a production problem would not only be very complicated, but would likely result in very little or no advantage at all.

Furthermore, if a few simplifications could be made to the problem investigated in this thesis, it would be possible to interpret it as a production problem such that existing algorithms could be directly "plugged in" to solve the problem. For example, if it was possible to assume that 1) all sessions of all patients have the same duration, 2) no restrictions are made on which days of the week patients can be scheduled on, 3) all sessions are required to be scheduled on consecutive days, and 4) the total machine capacity is the same for any day, the problem would become equivalent to an identical parallel machines problem, where each patient corresponds to a job of duration equal to their number of sessions and the number of appointment slots available per day corresponds to the number of machines. However, these simplifications do not properly reflect the problem faced in the real world.

Instead of directly modelling outpatient scheduling problems as a production problem, It is also possible to tackle the dynamic nature of the problem in similar fashion as reported in the production scheduling literature where jobs arrive continuously and can have different priorities. The majority of literature available on dynamic production scheduling problems adopts one of two approaches: dispatching rules or rescheduling. In dynamic problems, dispatching rules allow the scheduler to choose which job should enter a machine when it becomes available (Suresh &

Chaudhuri, 1993; Kutanoglu & Sabuncuoglu, 1999; Rajendran & Holthaus, 1999; Dominic et al., 2004). The job is chosen from a set of jobs which are ready to start, and the decision is usually made at the moment the machine becomes available or shortly before that. It is easy to see that this is not easily applied to outpatient scheduling, as it would require keeping all patients in the hospital, effectively making them inpatients. However, some authors use a different interpretation of dispatching rules, and apply them to healthcare problems the same way they are used in static production problems, where a schedule is constructed by sorting all available jobs according to one or more dispatching rules, and scheduling them on the earliest possible slot following that order (Kapamara & Petrovic, 2009).

Rescheduling can also be done in patient scheduling, but not in a straight forward manner as in production problems. In a production environment, jobs do not have other plans or lives, and usually do not mind being rescheduled. Patients, however, tend to dislike being told that the time for their appointment has been changed. Rescheduling patients also demands additional work from the hospital staff, who must call the rescheduled patients in order to confirm they are available at the new times. It is possible that a specific patient cannot be found to confirm the change, or that they are not available at the new time. In addition, when rescheduling many patients at the same time, it is possible for any of them to have a problem with the new schedule, making it necessary to create a new one. For these reasons, rescheduling outpatients is frowned upon by hospital staff.

Nonetheless, rescheduling patients can still be done if a few guidelines are followed:

1. The number of times that a patient is rescheduled plays an important role. If the appointment for a patient has already been rescheduled on a previous day, changing it again should be avoided if possible. This can be implemented by imposing a constraint on the number of times which patients can be rescheduled, or including a larger penalty for rescheduling a patient who has already been rescheduled.

2. The number of days between the current date and the time of the appointment should be considered. Patients whose appointment date is very far away from the current date are less likely to be upset if rescheduled than patients who are scheduled for an appointment

the next day. This can also be implemented by imposing a constraint on the minimum number of days in advance with which a patient can be rescheduled, or including a larger penalty for rescheduling appointments which are closer to the current date.

In the production scheduling scene, there are a few papers that are of relevance to this thesis. Erdelyi & Topaloglu (2009) present a stochastic approximation to find good levels of machine reservation for a general machine scheduling problem, where the objective is to minimise holding costs. Since the purpose of the machine reservation presented is to "protect" a portion of the machine capacity from lower priority jobs, in order to reserve it for future jobs of higher priority, they are referred to as "protection levels". Experiments are conducted using two algorithms: calculating the protection levels on the first day of the experiment and using them for the remaining days, or recalculating the protection levels every five days. They compare these two algorithms with a first-in-first-out (FIFO) principle and a rolling horizon approach, and find that the best results are found when the protection levels are recalculated every five days. This algorithm can be applied to outpatient scheduling if the objectives are interpreted as holding costs. However, it should be noted that the algorithm proposed also includes the possibility of rejecting jobs, which usually is not possible in healthcare problems. Also, the specific requirements of radiotherapy treatment scheduling would be very hard to model.

An example of rescheduling in a production environment is the meta-heuristic presented by Liu & Ong (2002, 2004) and Liu et al. (2005). Their initial work investigates a flow shop problem, where the problem is represented by a disjunctive graph (Roy & Sussman, 1964) and a neighbourhood based on the critical path in this graph is defined. The neighbourhood is combined with three meta-heuristics: Simulated Annealing (Kirkpatrick et al., 1983), Threshold Accepting (Dueck & Scheuer, 1990) and Tabu Search (Glover, 1986, 1989, 1990). In the succeeding paper, the methods are generalised for mixed shop problems, where there is a combination of flow, open and job shop jobs. In their later paper, the methods are generalised for dynamic mixed shop problems, considering breakdowns and new jobs which are continuously arriving. To deal with breakdowns and new jobs, they approach the problem as a static one, where the arcs for the operations which already took place are fixed and cannot be changed. All other arcs remain disjunctive, meaning all operations which have not taken place yet can be rescheduled. This method could be used in an outpatient scheduling problem, as long as it is possible to represent it as a mixed shop problem and the rescheduling is made while following the suggested guidelines.

Apart from production scheduling, there is no other classical optimisation problem that we are aware of which radiotherapy scheduling can be mapped to.

2.8 Summary

In this chapter, the available literature on outpatient scheduling is classified according to the environment they deal with, different characteristics of the pattern of patient arrivals, service times and patient and provider preferences, and according to the performance measures used. The most relevant literature is listed according to the methods used to approach the problem.

Rescheduling is not often used in outpatient scheduling literature. Although it should not be applied to outpatient scheduling the same way as in production scheduling problems, it can still be done if the presented guidelines are followed. Depending on the average number of patients rescheduled and on the improvement achieved, rescheduling can be a viable and interesting approach.

As it is shown in the following chapters, one of the most interesting aspects of the radiotherapy treatment scheduling problem to be explored is the large difference between the dates when the radiotherapy centre becomes aware of the patient (referred to as "decision to treat date" in the subsequent chapters) and the day the patient is actually able to start treatment (referred to as "release date" in the subsequent chapters). Although outpatient scheduling has been subject of many papers in the literature, not many of those explore this aspect. This is one of the main aspects explored in this thesis.

"Modern technology has become a total phenomenon for civilisation, the defining force of a new social order in which efficiency is no longer an option but a necessity imposed on all human activity."

Jacques Ellul (1912–1994)

Chapter 3

The Radiotherapy Treatment Scheduling Problem

This chapter presents the description of the radiotherapy treatment scheduling problem which is considered in this work. The arrival process of patients is described, as well as the time when schedules are created for each patient, the attributes of patients and machines, and the large set of constraints which make this a very challenging problem.

In order to better understand the problem and better design a solution, patient data are collected from the hospital and a data analysis is performed. This analysis is presented in this chapter and it has also been shown to the hospital to help them in gaining insight into the problem. A good knowledge of the nature of data and their characteristics is also important for the design of the more advanced algorithms present in this thesis, as well as for a correct interpretation of the results achieved by each method presented in the succeeding chapters.

Two sets of data are collected from the hospital. However, neither set has enough data on its own to enable a proper understanding of the patient intake, much less to be used in experiments. With that in mind, an algorithm to combine the two data sets is also presented in this chapter.

3.1 Problem Definition

The process before starting radiotherapy treatment consists of several phases. A patient is referred to an oncologist if a general practitioner (GP) suspects cancer. When visiting the oncologist, the patient enters the diagnostic phase, where several examinations are made to confirm the cancer, determine the size and stage of tumour and decide what form of treatment should be used. These may include radiotherapy, surgery, chemotherapy and hormone therapy among others. At the end of the diagnostic phase, the oncologist makes a decision to treat with one or more forms of treatment with the consent of the patient.

If radiotherapy is chosen, the patient enters the pre-treatment stage, where a deeper analysis of the location and shape of the tumour is made to properly calibrate the radiation machines. This stage includes deciding on the radiation parameters, such as the dose of radiation, number of beams, angles between them, etc., and verification of the plan on a simulator. The patient should remain immobile during the delivery of radiation, which, in some cases, will require a cast to be produced during pre-treatment.

Radiation therapy (or radiotherapy) treatment involves exposing the patient to beams of accelerated subatomic particles with the intent of destroying the cancer tumour while minimising damage to the surrounding organs. The intensity and direction of the beams will depend on characteristics of tumour and patient. To allow enough time for the healthy organs to recover, the radiation is divided in fractions (Barendsen, 1982; Thames et al., 1982), which are received on treatment sessions scheduled for the patient at the rate of one fraction per session.

The radiotherapy treatment scheduling problem can be defined as the problem of scheduling a number of radiotherapy sessions for a number of patients on linear accelerator machines (referred to as *linacs*). It is considered as a daily scheduling problem in which a number of patients who must be scheduled on linacs enter the booking system, which are partially booked with previously scheduled patients. At the end of each day, the radiographer creates a schedule for the patients who arrived on that day.

Depending on tumour site, patients can require a specific type of radiation from the linacs, where the available radiation types are high energy photon, low energy photon and electron. This imposes a linac eligibility constraint, since not all linacs can emit all types of radiation.

Each linac can attend only one patient at a time. The capacity of each linac, measured by the number of working hours of the hospital staff, must not be exceeded on any given day.

Patients require a number of radiotherapy sessions, and the duration of sessions can differ from one patient to another, or even amongst the sessions of the same patient. Commonly, the first session of each patient is longer due to calibration and verification procedures (Turner & Bing, 2002).

The dates on which patients are allowed to have treatment sessions are also strict. Patients can start treatment on their release date which, in most cases, is the date when their pre-treatment is finished. Some patients have radiotherapy as an adjuvant treatment to increase the chances of success of a previous treatment. In these cases, the previous treatment must also be finished and the patient may need additional time to recover before starting radiotherapy.

Patients may require 1, 2, 3 or 5 session days per week, where a session day is a day when the patient is required to go to the radiotherapy centre to receive one or more fractions of the treatment. Patients who may be treated on weekends can require up to 7 session days per week. Sessions must be scheduled with strict numbers of days between them, such as:

- patients who have 1 session day per week must have all sessions on the same day of the week for consecutive weeks,
- patients who have 2 session days per week must be scheduled either on Mondays and Thursdays consecutively or on Tuesdays and Fridays consecutively,
- patients with 3 session days per week must be scheduled on Mondays, Wednesdays and Fridays consecutively,
- patients with 5 session days per week must have them on consecutive days excluding weekends, and

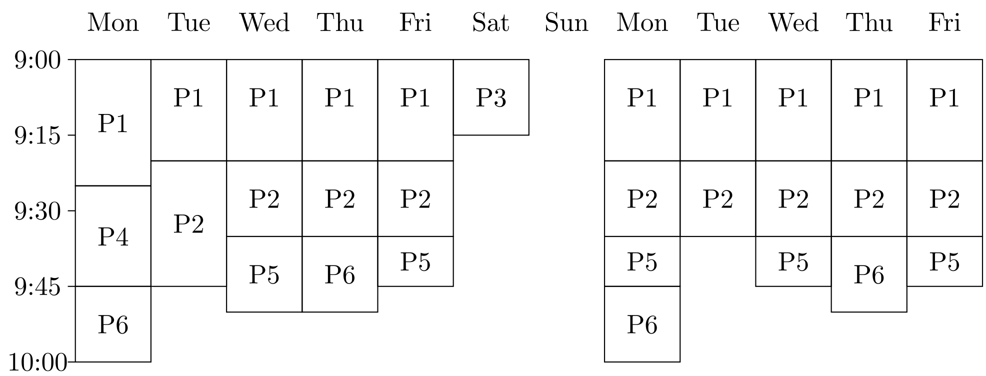


Figure 3.1: An example of a typical schedule where the duration of the first session of each patient is slightly longer than the others. Patient P1 has 10 sessions, 5 per week; patient P2 has 9 sessions, 5 per week; patient P3 has 1 session on Saturday; patient P4 has 1 session; patient P5 has 5 sessions, 3 per week; and patient P6 has 4 sessions, 2 per week

- patients with 7 session days per week must have them on consecutive days including weekends.

In addition, the presence of the doctor is required for the first session of some patients. Since each doctor is available in the radiotherapy centre on only a few days of the week, this imposes one more eligibility constraint.

Some patients must have a minimum number of sessions before the first weekend in order to prevent the tumour from growing back after the first session. For example, some palliative patients must have at least 2 sessions before the first weekend, and therefore, cannot start their treatment on a Friday. Some patients with 5 or less sessions are required to have them all on the same week on consecutive days, without interruptions.

The majority of patients have only 1 session per day. The exceptions are CHART patients (Continuous Hyper-fractionated Accelerated Radiotherapy Treatment), who require 3 fractions per day for 12 consecutive days with treatment starting on a Monday. Figure 3.1 shows an example of a schedule for one linac where the opening times are set as from 9:00 to 10:00 for simplicity.

Patients are grouped into different categories based on their waiting list status (WLS): emergency, urgent or routine. The WLS of a patient is decided by considering the site and the level of advancement of the tumour. Patients are also grouped according to their treatment intent as radical (with the intent to cure) and palliative (with the intent to alleviate the symptoms and improve the patient's quality of life). This classification is also used in the most recent audits by Drinkwater & Williams (2008) and Summers & Williams (2006).

Three due dates are set for each patient. The first is established by the Department of Health (2004, 2007). It states that each patient must start their treatment no later than 31 days from the date when the decision to treat with radiotherapy was made. If the patient had an urgent referral from the general practitioner (GP) and radiotherapy is used as the first treatment, the radiotherapy start must also be no more than 62 days from the GP referral. The earliest of these is referred to as the *breach* date. The UK Cancer Network evaluates each radiotherapy centre according to the number of patients that breach this due date, thus minimising this number is the primary objective in this research.

It is possible to use radiotherapy as an adjuvant treatment, where a different treatment is performed first (e.g. surgery or hormone therapy) and radiotherapy is used to increase the chance of success in destroying the tumour. In these cases, the date the patient had a CT scan during the pre-treatment is used to calculate the breach date instead of the date of the decision to treat. This is the same procedure adopted by the UK Cancer Network to evaluate the performance of radiotherapy centres.

The other two due dates have been established by the Joint Council for Clinical Oncology (1993) (JCCO). They determine the good practice and the maximum acceptable waiting times from the date the patient is first seen for suspected cancer to the first session of treatment for each category of patients. Table 3.1 shows the JCCO waiting time targets which have been adjusted to the nomenclature used in this work, as has been suggested by Drinkwater & Williams (2008) to better reflect the nomenclature currently in use in hospitals. The JCCO due dates are acknowledged by the majority of radiotherapy centres in the UK (Ash et al., 2004; Summers & Williams, 2006) and considered as a secondary objective in this research. The calculation of the target dates is illustrated in Figure 3.2.

	emergency		urgent		routine	
	palliative	radical	palliative	radical	palliative	radical
Good Practice	24 hours	24 hours	48 hours	2 weeks	48 hours	2 weeks
Max. Accept.	48 hours	48 hours	2 weeks	4 weeks	2 weeks	4 weeks

Table 3.1: Waiting time due dates established by the JCCO adjusted to the nomenclature used in this work

In addition to these three targets, in this work the minimisation of waiting time from the decision to treat to the start of treatment is considered. The hospital aims at minimising the waiting time while distributing it as evenly as possible among patients of the same category. To measure it, we compute the weighted sum of the squared waiting times. This criterion can be frequently seen in the literature for machine scheduling (Bagchi et al., 1987), often used when large deviations of completion time from the due date are undesirable.

To illustrate how the sum of squared waiting times can be applied to our problem and how it differs from other frequently used criteria such as the sum of waiting times or the maximum waiting time, let us consider the following example: on a given day, 3 patients arrive at the radiotherapy centre to be scheduled. Let us suppose that in one possible schedule, patients 1, 2 and 3 have a waiting time of 1, 3 and 3, respectively, while in a second schedule, the waiting times are 2, 2 and 3. The hospital would prefer the second schedule since it distributes the waiting time more evenly among the patients. However, if either the sum of waiting times or the maximum waiting time are used, the value of the objective function for each schedule will be the same for both schedules (7 and 3 respectively) making them indistinguishable. If the sum of squared waiting times is used, the value of the objective function will be 19 and 17 for the first and second schedule respectively, enabling the algorithm to correctly choose the second schedule, which would be preferred by the hospital.

A weight is also assigned to each patient, which defines the relative importance of that patient with respect to the set of JCCO waiting time due dates and the squared waiting time. In this approach it is set as 10 for emergency patients, 3 for urgent and 1 for routine. These values were suggested by the hospital staff that collaborated with us.

To summarise, each patient will have the following attributes:

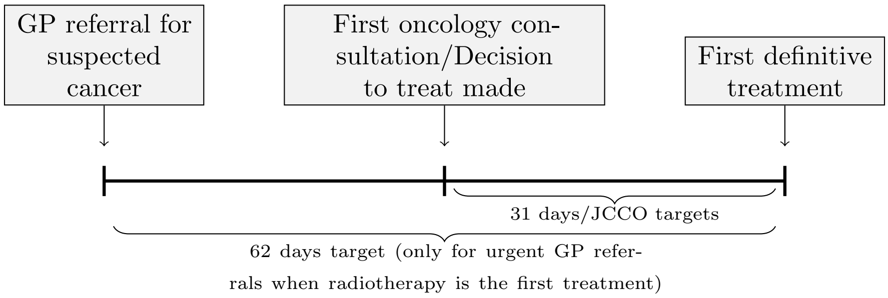


Figure 3.2: Time-line and waiting time targets of patients diagnosed with cancer

1. The waiting list status (emergency, urgent or routine);

2. The required radiation type for all treatment sessions (high energy photon, low energy photon or electron);

3. The intent of treatment (palliative or radical);

4. The date when the decision to treat is made;

5. The release date (date when the patient finishes the pre-treatment and is available to start the treatment);

6. The breach (due) date by which the patient should start the treatment as established by the Department of Health (2005);

7. The maximum acceptable due date by which the patient should start the treatment as established by the Joint Council for Clinical Oncology (1993);

8. The good practice due date by which the patient should start the treatment as established by the Joint Council for Clinical Oncology (1993);

9. The number of required sessions;

10. The number of required sessions per day;

11. The number of required session days per week;

12. The set containing the days of the week when the patient is allowed to have the first session;

13. The duration of each session.

Currently, the schedule in the hospital is created by hand by the booking clerk for each patient. Schedules are created a few days before the patient's release date and no re-scheduling is made. A similar approach is presented in Section 4.5.

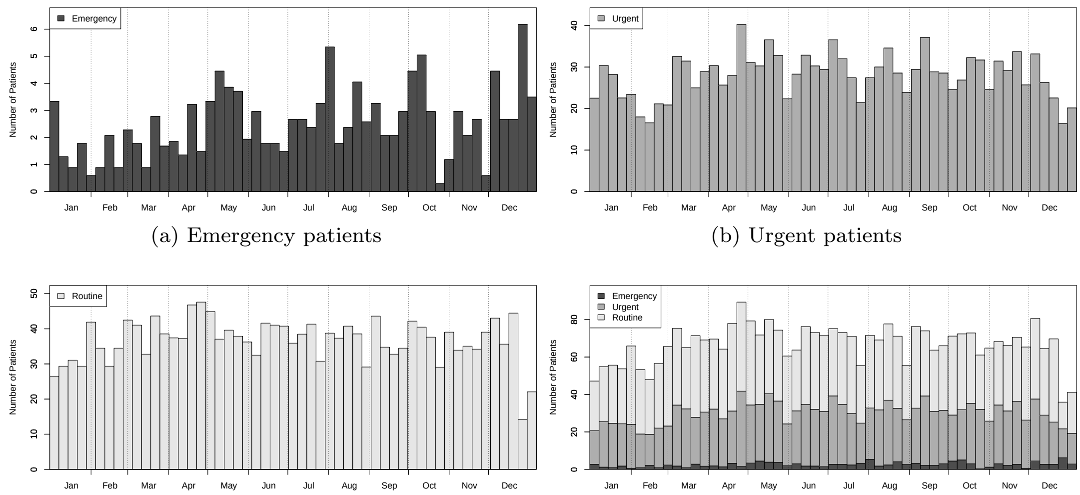
3.2 Collected Real-world Data

The hospital provided us with two sets of patient data. The first set contains the data about over 12,000 patients treated in a period of five years from 2001 to 2005. These data contain the following attributes of each patient: waiting list status, intent, required radiation type and date when the decision to treat has been made. However, it should be noted that this data set is incomplete in the sense that several attributes necessary for creating a schedule are missing.

The second set contains data about 173 patients treated in a period of roughly one month around June/2009 and has all the attributes from each patient necessary to build a schedule. It was manually collected by the hospital staff using forms prepared by us. An example of the forms used is given in Appendix A.

It is not possible to use either set on its own, since many attributes are missing from the first set and there are not enough patients in the second set to yield meaningful results. Therefore, both sets must be taken into consideration when making the analysis of patient intake.

The first aspect to be analysed is the number of patients that arrive per day, where the arrival date of a patient is considered to be the day when the decision to treat with radiotherapy is made. For this analysis, only the first data set is used. Two types of seasonality are identified in the arrival of patients. The first is according to the week of the year. Figure 3.3 shows the average number of patients of each waiting list status for each week of the year. It is possible to observe that early in the year during the winter, the number of patients arriving each day is smaller than the year average. It slowly increases in the next months, coming to a few peaks of patient arrivals in April and May. There is little variation in the next months, ending with



(a) Emergency patients

(b) Urgent patients

(c) Routine patients

(d) All patients

Figure 3.3: Average number of patients for each week of the year

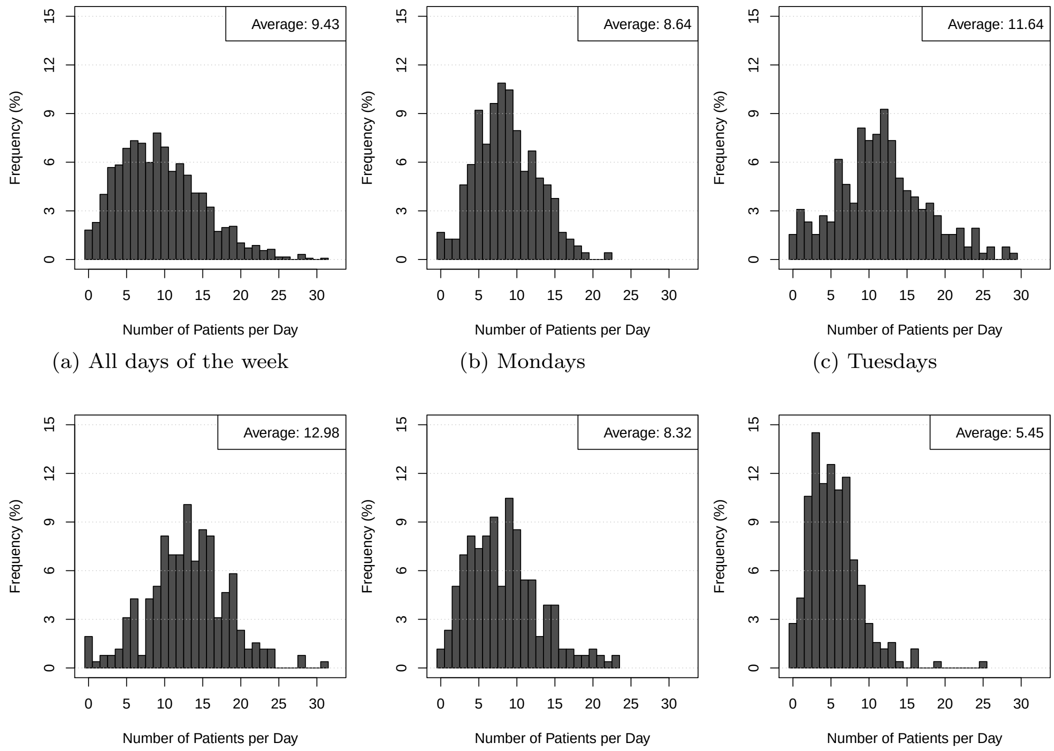
a steep drop in patient arrivals in the last two weeks of December. However, this variation is slightly different for each waiting list status. The number of emergency patients has a smaller variation in the period January-April, while the drop in the last two weeks of the year is not as steep for emergency and urgent as it is for routine patients.

The second type of seasonality is according to the day of the week. Figure 3.4 shows histograms of the number of patient arrivals per day considering all days and separately for each day of the week. It is possible to see that the number of patient arrivals is usually high on Tuesdays and Wednesdays, slightly low on Mondays and Thursdays, and very low on Fridays.

The next aspect to be analysed is the frequency of each type of patient. Patients are classified according to three attributes: waiting list status, treatment intent and required radiation type. Figure 3.5 shows the frequency of each combination of these patient attributes for each data set.

It is possible to notice some key differences between the two data sets by comparing Figures 3.5a and 3.5b:

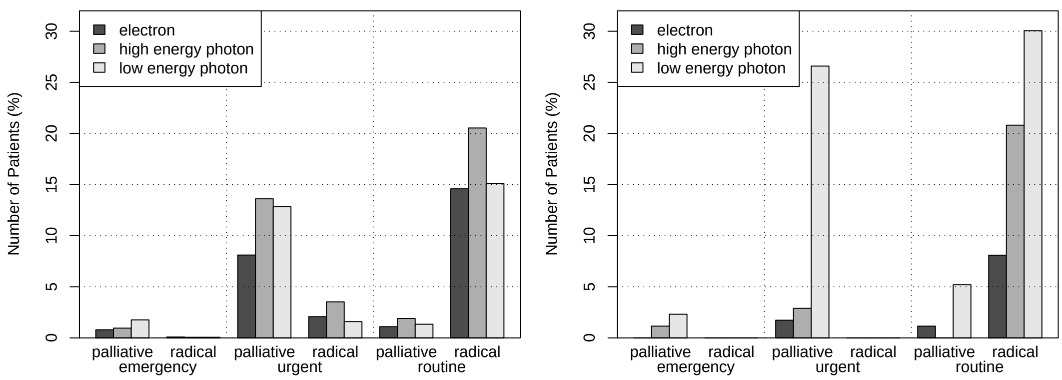
- There are some emergency palliative patients who require electron radiation in the first data set, but none in the second;



(a) All days of the week (b) Mondays (c) Tuesdays

(d) Wednesdays (e) Thursdays (f) Fridays

Figure 3.4: Histograms of the number of patient arrivals per day considering all days and separately for each day of the week



(a) First data set (b) Second data set

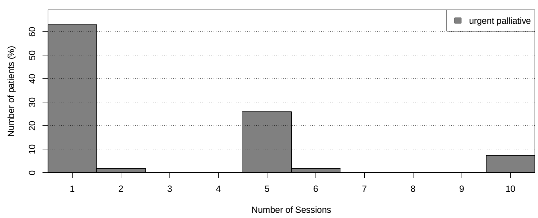
Figure 3.5: Frequency of patient types in each data set

- There are some emergency and urgent radical patients in the first data set, but none in the second;
- There are some routine palliative patients who require high energy photon radiation in the first data set, but none in the second;
- There is a much larger proportion of patients who require low energy photon radiation in the second data set than in the first.

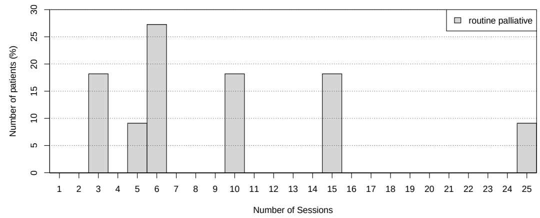
The hospital staff was consulted on each difference and indicated which data set better reflects reality. The attribute combinations mentioned above that appear in the first data set and not in the second are likely to give us wrong information and should not be taken into consideration. On the other hand, the time period when the second data set was collected seems to be an atypical one with regard to the high number of patients requiring low energy photon radiation, and this information is better taken from the first data set.

Other interesting aspects of the data are present only in the second data set and include the length of pre-treatment, the number of sessions and the frequency of sessions per week. Figure 3.6 shows bar charts of the number of fractions for different waiting list status/intent combinations of patients according to the second data set. It is interesting to see that there seems to be a strong correlation between the number of sessions of a patient and the waiting list status/intent. All emergency and around 63% of urgent patients have only one fraction. Routine patients usually have a very high number of sessions, with an average of 21 sessions for each patient. Also, around 64% of the patients who have more than 1 session have a number of fractions multiple of 5, showing a preference for treatments which take a round number of weeks.

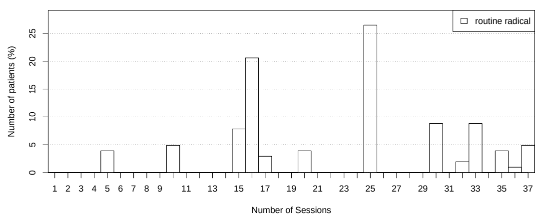
Figure 3.7 shows bar charts of the length of the time period between the decision to treat and the release date for each waiting list status/intent combination from the second data set. During this time, the patient goes either through the pre-treatment phase or a different treatment. One can also notice a strong correlation between the length of this time period and the waiting list status/intent of the patients. Emergency patients have the shortest time period between the decision to treat and the release date of 1 day in average. In contrast, urgent patients have



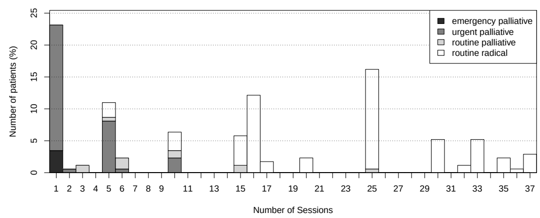
(a) Urgent palliative patients



(b) Routine palliative patients



(c) Routine radical patients



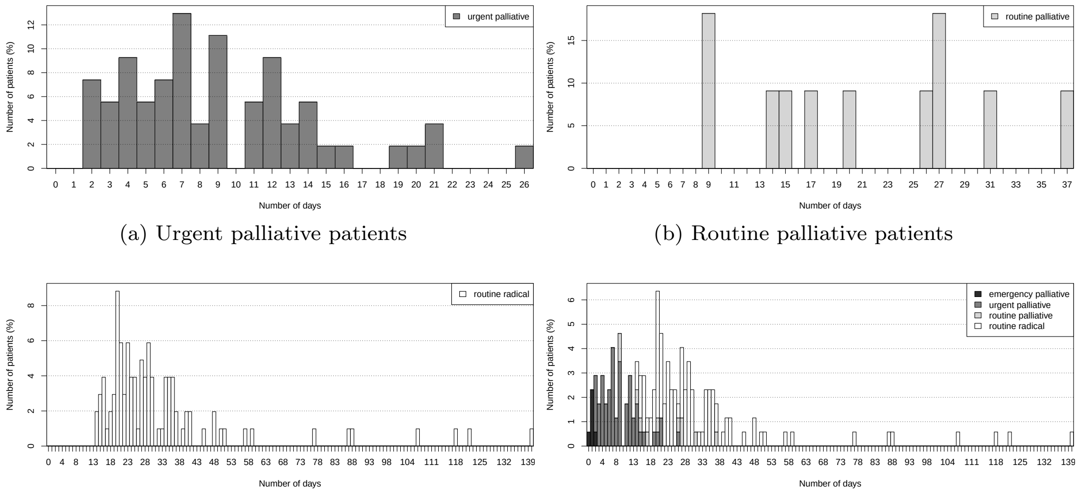
(d) All patients

Figure 3.6: Bar charts of the number of fractions for each patient type

a release date on average 11 days after the decision to treat has been made, routine palliative patients have an average of 18 days and routine radical patients have an average of 33 days. As previously stated, some radical patients have radiotherapy as an adjuvant treatment, which explains the highest values present in Figure 3.7.

It should also be noted that the differences presented in Figure 3.7 make it impossible for some patients to meet some of the due dates. It is impossible for 17% of emergency patients to meet the JCCO good practice waiting time target of 1 day due to their release date being after this due date, but it is possible for all of them to meet the JCCO maximum acceptable waiting time target of 2 days. Around 94% of non-emergency palliative patients cannot meet the JCCO good practice of 2 days, and 23% cannot meet the JCCO maximum acceptable of 14 days. For radical patients, the due dates are even harder to meet, as 98% of radical patients cannot meet the JCCO good practice of 14 days and 45% cannot meet the maximum acceptable of 28 days. In addition, 12% of patients, all of who are routine and radical, cannot meet the breach date of 31 days.

Table 3.2 shows the proportion of patients of each waiting list status with each combination of number of session days per week/sessions per day. There also seems to be a correlation between



(a) Urgent palliative patients

(b) Routine palliative patients

(c) Routine radical patients

(d) All patients

Figure 3.7: Bar charts of the length of the time period between the decision to treat and release date given in days

Waiting list status	Session days per week/sessions per day				
	1/1	2/1	3/1	5/1	7/3
Emergency	3.5%	–	–	–	–
Urgent	19.7%	–	–	11.6%	–
Routine	–	1.2%	0.6%	63.0%	0.6%

Table 3.2: Proportion of patients of each waiting list status with each combination of number of session days per week/sessions per day

these attributes and the waiting list status of each patient. It is possible to see that the majority of patients (around 63%) have 5 sessions per week and 1 session per day, where the exceptions are:

- 6 emergency and 34 urgent patients who have 1 session days per week (who also have 1 session in total),
- 2 routine patients who have 2 sessions per week,
- 1 routine patient who has 3 sessions per week,
- and 1 routine patient classified as CHART (see page 46), who has 3 sessions per day, 7 days per week for 12 days with treatment starting on a Monday.

This data analysis has helped us and the hospital staff in gaining insight into the problem. The results are also used later in the succeeding chapters to generate a synthetic experimental data set, interpret most of the results and to design the more advanced methods presented in this work.

3.3 Combining the Two Data Sets

In order to make better sense of the data and to prepare for generating instances for the experimental part of this work, the author decided to combine the two data sets presented in the previous. In this section, a description of the algorithm developed to combine the data is given.

As established in the previous section, the requirements to be considered for the combined data set are the following:

(a) The two types of seasonality (inside a year and inside a week) found in the first data set and shown in Figures 3.3 and 3.4 should be taken into account.

(b) The frequency of each combination of waiting list status, intent and required radiation type should be similar to the first data set as shown in Figure 3.5a.

(c) The combinations of patient waiting list status, intent and required radiation type which are not present in the second data set in Figure 3.5b should not appear in the combined data set.

(d) The correlations between the remaining attributes of patients present in the second data set presented in Figures 3.6, 3.7 and Table 3.2 must be preserved.

With those requirements in mind, an algorithm is developed to combine the two data sets. The missing attributes of each patient $J1$ of the first data set are filled in by replacing the patient by another patient $J2$ from the second set, such that $J2$ is as similar to $J1$ as possible. This algorithm is detailed below:

Step 1 For each patient $J1$ in the first data set, repeat steps 2 to 9.

Step 2 Take a random patient $J2$ from the second data set with the same waiting list status, intent and required radiation type as patient $J1$.

Step 3 If there is no such patient, take a random patient $J2$ from the second data set with the same waiting list status and intent as patient $J1$.

Step 4 If there is no such patient, take a random patient $J2$ from the second data set with the same waiting list status and required radiation type as patient $J1$.

Step 5 If there is no such patient, take a random patient $J2$ from the second data set with the same waiting list status as patient $J1$.

Step 6 Create a copy of patient $J2$ named JC.

Step 7 Set the decision to treat date of patient JC equal to the decision to treat of patient $J1$;

Step 8 Adjust the remaining dates of patient JC according to the new decision to treat date.

Step 9 Add patient JC to the combined data set.

As the number of patients of each waiting list status on each day in the combined data set is exactly the same as in the first set, both types of seasonality described previously and shown in Figures 3.3 and 3.4 are preserved and requirement (a) is fulfilled. To fulfil requirement (b), for each patient $J1$ from the first set, the algorithm searches the second data set for a patient $J2$ to be added to the combined data set, such that $J2$ as similar as possible to patient $J1$ from the first data set. Finally, since all patients in the combined data set are re-samples from the second data set, requirements (c) and (d) are also fulfilled.

The same data analysis performed on the second data set and presented in Figures 3.5–3.7 and Table 3.2 is performed on the combined data set and presented in Figures 3.8–3.10 and Table 3.3.

From Figure 3.8, it is possible to see that requirements (b) and (c) are respected. The frequency of the combinations for waiting list status, intent and radiation type are similar to the

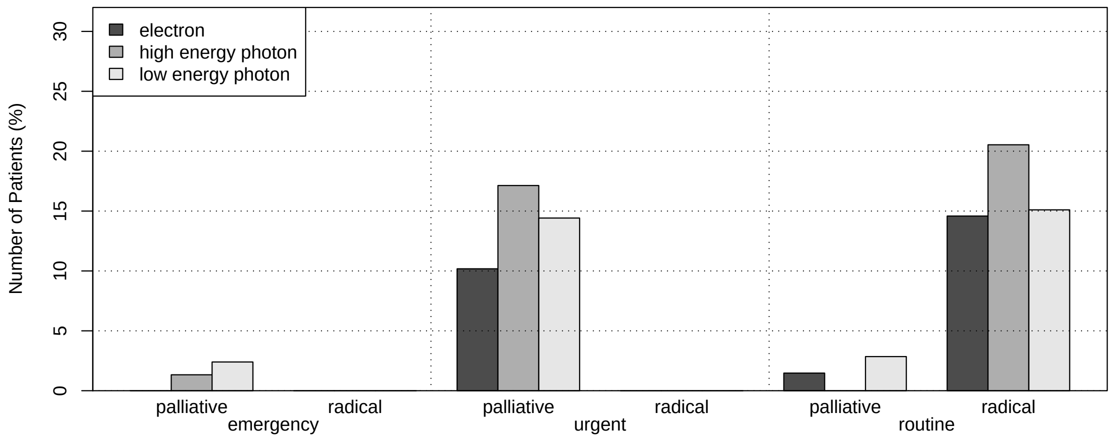


Figure 3.8: Frequency of patient types in the combined data set (same analysis as Figure 3.5)

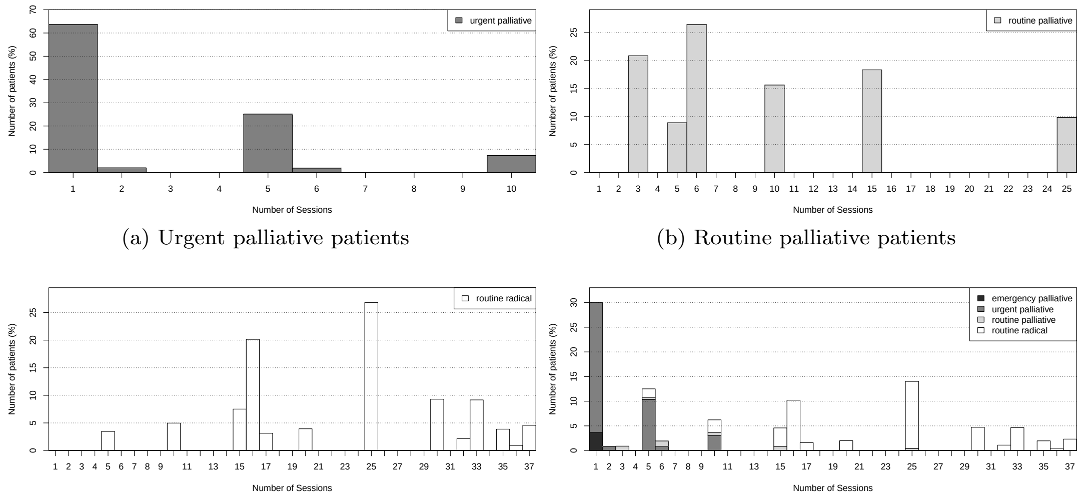
Waiting list status	Session days per week/sessions per day				
	1/1	2/1	3/1	5/1	7/3
Emergency	3.7%	–	–	–	–
Urgent	26.4%	–	–	15.3%	–
Routine	–	0.7%	0.8%	52.8%	0.3%

Table 3.3: Proportion of patients of each waiting list status with each combination of number of session days per week/sessions per day (same analysis as Table 3.2)

ones presented in Figure 3.5a, and the combinations not present in Figure 3.5b do not exist. The frequency of each patient type has changed slightly in account of the differences between the frequency of combinations for waiting list status, intent and radiation type as shown in Figure 3.5 and 3.8. However, the correlations between the remaining attributes in Figures 3.9–3.10 and Table 3.3 remain roughly the same as in Figures 3.6–3.7 and Table 3.2.

3.4 Summary

The description of the radiotherapy treatment scheduling problem investigated in this work is presented in this chapter. It is treated as the problem of creating a treatment schedule for radiotherapy patients who arrived on the current day in a radiotherapy centre. All constraints are explained, such as linac eligibility, possible days of the week to start treatment, and fixed time intervals between sessions. The objectives considered in this problem consist of minimising



(a) Urgent palliative patients

(b) Routine palliative patients

(c) Routine radical patients

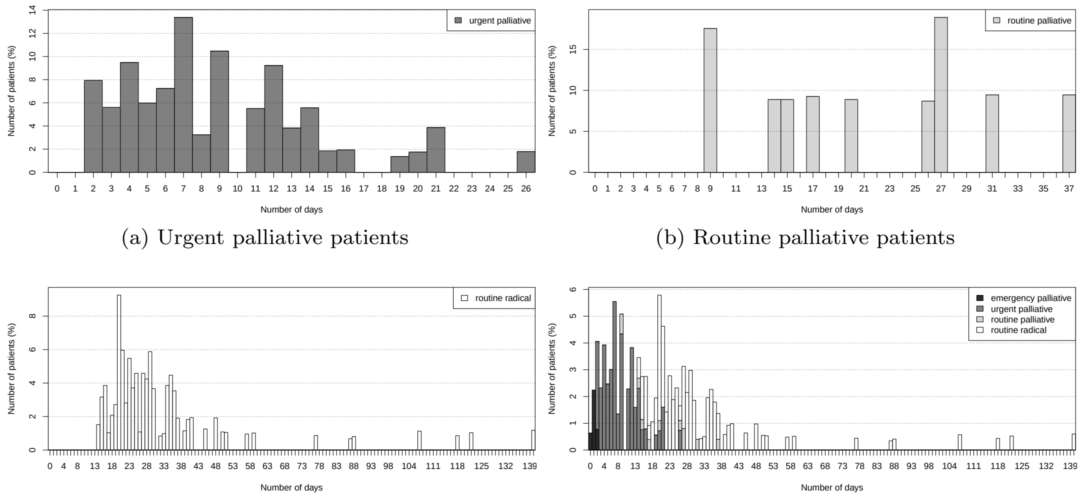
(d) All patients

Figure 3.9: Bar charts of the number of fractions for each patient type in the combined data set (same analysis as Figure 3.6)

the number of patients who miss each of the three considered due dates and minimising the average squared waiting time.

Two real-world data sets are given by the hospital. A data analysis of the most important attributes from each data set is also presented. It reveals that there are two types of seasonality in the arrival of patients, one according to the time of year and one according to the day of the week. Furthermore, all analysed patient attributes seem to be correlated with the waiting list status and the intent.

The first set contains data of a greater number of patients, but is incomplete in the sense that many of the required attributes to create a schedule are missing. The second set contains data for all required attributes, but for a very small number of patients. Neither of the two collected data sets is large enough to be used on its own to make a proper analysis of the data or to run experiments. With that in mind, an algorithm is proposed to combine the two data sets, while following the requirements of reflecting both types of seasonality present in the first data set, and maintaining the correlations between the analysed attributes and the waiting list status. An analysis of the combined data is also presented.



(a) Urgent palliative patients

(b) Routine palliative patients

(c) Routine radical patients

(d) All patients

Figure 3.10: Bar charts of the length of the time period between the decision to treat and release date given in days in the combined data set (same analysis as Figure 3.7)

"When I'm working on a problem, I never think about beauty. I think only how to solve the problem. But when I have finished, if the solution is not beautiful, I know it is wrong."

Richard B. Fuller (1895–1983)

Chapter 4

Constructive Approaches

The first algorithms used to tackle the radiotherapy treatment scheduling problem are four constructive approaches presented in this chapter. The main advantage of constructive heuristics is that they are easy to implement and have very low computational costs. The main goal of this chapter is to investigate the quality of schedules produced by these approaches.

To perform this investigation, an experimental data set is needed. In order to run a large number of experiments and to perform a better statistical analysis of the results, an algorithm is presented to generate an experimental data set based on the combined data set presented in Section 3.3.

Different approaches are developed to achieve schedules that are high quality with regard to the patients currently being scheduled, but which are also robust with regard to patients arriving in the near future. The first constructive approach is concerned with the day the treatment of each patient starts. It tries to postpone the treatment start of some patients in order to leave more space on the machines for future patients. The second approach is concerned with machine utilisation, where the goal is to establish an utilisation threshold for patients of each waiting list status, such that a session is not allowed to be scheduled on given machine and day if it would cause this threshold to be exceeded. The third approach is concerned with the frequency of scheduling. Its main inspiration is that if patients do not have their schedules created immediately and the radiographer waits for the accumulation of patients, the algorithm could generate better schedules since the search space will be larger. Finally, the fourth approach deals with scheduling patients in advance. The release date of routine patients is usually very far away from their decision to treat date. If the creation of these schedules is postponed, schedules for patients of higher priority can be created first. These approaches are detailed in the succeeding sections.

4.1 Experimental Environment

To be able to run a large number of experiments and to perform a statistical analysis of the results, the combined patient data set presented in Section 3.3 is used to generate 33 different input instances. Each instance consists of one and a half year of patient arrivals, where the first six months are used only to fill the booking system and avoid start up effects. At the end of the remaining year of data, the performance of the scheduling algorithm is evaluated. These instances are used in all the experiments in this work.

The experimental data must be generated in a way that complies with the same requirements as the combined data, summarised in Section 3.3. With that in mind, the following algorithm is developed to create each instance:

Step 1 Randomly choose one of the 18 months long time intervals: July/2001 to December/2002, July/2002 to December/2003, July/2003 to December/2004 or July/2004 to December/2005.

Step 2 For each date k in the chosen interval, repeat steps 3 to 9.

Step 3 If date k is a weekend or a bank holiday, set the number of patients arriving on date k to 0 and move to the next date. Otherwise, go to step 4.

Step 4 Select a random year from the combined data set.

Step 5 Choose a date k' from the year selected in the previous step, such that k and k' are the same day of the week, k' is not a bank holiday and is as close as possible to the day of the year of date k. For example, consider that date k is the 3^{rd} of January of 2005 (Monday) and the random year chosen from the combined data set is 2001. Date k' will be a Monday from 2001 which is not a bank holiday and is as close as possible to the 3^{rd} of January, i.e. the 8^{th} of January of 2001.

Step 6 For each waiting list status $t \in$ {emergency, urgent, routine}, repeat steps 7 to 9.

Step 7 Consider N as the number of patients of waiting list status t arriving on day k' on the combined data set.

Step 8 Select N random patients of waiting list status t from the combined data set and copy them to the experimental data set.

Step 9 Change the decision to treat of copied patients to k and adjust the remaining dates accordingly.

It is easy to see that the requirements summarised in Section 3.3 are followed. For each date in the generated instance, the number of patients of each waiting list status is the same as a date in the combined instance of the same day of the week and as close as possible to the same date of the year, fulfilling requirement (a). Finally, as all patients are re-sampled from the combined data, requirements (b), (c) and (d) are also fulfilled.

The linacs environment used in the experiments is the same as the one currently in use in the oncology ward of the hospital. It is consisted of 4 linacs:

- 1 that emits low energy photon radiation (type A),
- 1 that emits electron and low energy photon radiation (type B),
- 2 that emit electron, low and high energy photon (type C).

As a result, each instance contains roughly 3,600 patients who arrive during 18 months long time period.

Linacs are available from 8:45 to 18:00 on Monday to Friday for weekday sessions and from 9:00 to 13:00 on Saturdays and Sundays for weekend sessions.

The current scheduling policies in the hospital are followed. Patients who require low energy photon must be scheduled on linacs of type A, and patients who require electron radiation must be scheduled on linacs of type B. Patients who require high energy photon can only be scheduled on linacs of type C, regardless of any policy.

Each experiment consists of a simulation of the everyday scheduling of a hospital for a period one and a half years. Each day, a number of patients arrive at the radiotherapy department to be scheduled. At the end of the day, the radiographer uses the constructive algorithm to create a schedule for patients who are available to be scheduled that day.

Experiments are run on a PC with an Intel Xeon 3.0GHz CPU and 2GB of RAM under the Scientific Linux operating system.

4.2 Target Date Approach

All constructive approaches follow the same outline and consist of three phases. In the first phase, patients who are available for scheduling are identified. These are, at first, any patients who already had their decision to treat and have not been scheduled yet. In a second phase, patients available for scheduling are ordered lexicographically by their waiting list status, release date and number of required sessions. In the third phase, schedules are created following that order. The approaches differ from each other in the first and third phases, as will be explained further.

The first constructive approach developed for this work is called the *target date* approach and it is a generalisation of two algorithms presented by Petrovic et al. (2006). These algorithms operate in a forward (backward) manner from the release date (due date) of each patient, trying to schedule the required number of sessions subject to the given constraints. If it is not possible to accommodate all the required sessions, the algorithms move the start day forward (backward) and try again. This is repeated until the patient is scheduled or all days between the release and due dates have been tried. Finally, if the patient is still not scheduled, then the start day is moved to the first available day after the due date.

The generalised approach developed in this thesis tries to schedule the patient so that his/her treatment starts as close as possible to a pre-defined date within the [release date, due date] time window. This pre-defined date is referred to as *target date*. The parameter *target index* ($TI : 0 \leq TI \leq 1$) is introduced to calculate the target date in the following way:

$$\text{target date} = \text{release date} + TI\ (\text{due date} - \text{release date}). \tag{4.1}$$

By using Equation 4.1, the algorithm tries to postpone the start of treatment of each patient by a fixed proportion of the time difference between the release and due dates. At the time this method was first developed, the only due date considered was the JCCO maximum acceptable. Therefore, it is also used as due date in Equation 4.1 to maintain consistency.

If it is not possible to schedule the patient on the target date, the algorithm tries to schedule the patient one day later. If that is not possible, it tries one day earlier. If that is not possible, the algorithm keeps trying one more day forward and one more day backward repetitively, until either a schedule is possible or all days between the release and due dates have been tried. If a schedule is still not possible, the patient is scheduled on the earliest available day after the due date.

The motivation for this approach is that if patients of lower priority are scheduled at later dates, it might be possible to achieve a better schedule for patients of higher priority with respect to the due dates. Experiments are run on the sets of generated data with the possible values for TI of 0.00, 0.25, 0.50, 0.75 and 1.00 for urgent and routine patients, while for emergency patients it is set as 0.

When running a number of experiments with randomly generated data, it is possible that one set of experiments achieves a better average result than other sets simply by chance.

In order to determine whether or not the means of the objective function values of two sets of experiments are statistically different, the Mann-Whitney U (MWW) test is used on experiment values re-sampled by bootstrapping (Léger et al., 1992). The MWW test (Mann & Whitney, 1947) is able to determine if there is significant statistical evidence that one *configuration* achieves better results than another, where a configuration corresponds to the set of parameter values used in the experiments. Bootstrap (Efron, 1979) is a computationally intensive technique based on data re-sampling. It enables us to perform valid statistical tests without making unrealistic or unverifiable assumptions about the characteristics of the values of the objective functions, such as their distribution or variance (Yuan & Gallagher, 2007).

The bootstrapping procedure is applied on the results as follows. Given the set $\mathcal{F}$ of 33 values found for a specific objective function and configuration, a bootstrapping procedure creates B sets of bootstrapped values $\mathcal{F}'_a, a = 1, \ldots, B$. Each set $\mathcal{F}'_a$ contains $|\mathcal{F}|$ values (33 in this case, one for each instance) randomly sampled with replacement from the original set $\mathcal{F}$. The next step is to calculate the statistic of interest θ'_a for each set of values $\mathcal{F}'_a$. In this work, this is the average value of the given objective function. The averages θ'_a are then used in the MWW tests to determine with a given confidence which configuration achieves the best results for that objective function.

The MWW tests are run for each pair of configurations with an overall confidence of 90% and the number of bootstrap replications B is set to 1000. In order to achieve the desired overall confidence, the confidence level of each individual test is approximated using the Bonferrani Inequality as

$$\text{individual confidence} = 1 - \frac{1 - \text{overall confidence}}{\frac{n\ (n-1)}{2}}, \tag{4.2}$$

where $\frac{n\ (n-1)}{2}$ is the number of pairwise comparisons of n configurations, i.e. the number of necessary individual tests.

Results for each objective function are shown in Table 4.1, where “Breach” shows the percentage of patients who did not meet their breach date, “JMax” and “JGood” show the weighted percentage of patients who did not meet their JCCO maximum acceptable and good practice due dates respectively, and “Waiting” shows the average weighted squared waiting time

		0.00	0.25	0.50	0.75	1.00
0.00	Breach (%)	**34.95**	**34.95**	**34.94**	**34.97**	**34.96**
	JMax (%)	**56.53**	**56.53**	**56.58**	**56.61**	56.80
	JGood (%)	**93.95**	**93.91**	**93.96**	**93.94**	**93.97**
	Waiting	**1,347**	1,350	1,357	1,368	1,387
0.25	Breach (%)	35.27	35.27	35.26	35.27	35.29
	JMax (%)	**56.68**	**56.68**	**56.71**	56.75	56.93
	JGood (%)	95.28	95.22	95.24	95.24	95.25
	Waiting	1,370	1,373	1,380	1,391	1,411
0.50	Breach (%)	35.57	35.57	35.59	35.62	35.61
	JMax (%)	**56.80**	**56.79**	56.84	56.86	57.10
	JGood (%)	95.30	95.29	95.27	95.27	95.27
	Waiting	1,390	1,393	1,403	1,413	1,432
0.75	Breach (%)	36.26	36.26	36.29	36.30	36.30
	JMax (%)	57.19	57.18	57.26	57.31	57.52
	JGood (%)	95.32	95.32	95.33	95.32	95.33
	Waiting	1,432	1,436	1,444	1,456	1,475
1.00	Breach (%)	37.08	37.08	37.11	37.13	37.14
	JMax (%)	57.32	57.31	57.41	57.54	57.77
	JGood (%)	95.32	95.29	95.30	95.33	95.34
	Waiting	1,485	1,488	1,497	1,508	1,528

Table 4.1: Results obtained varying the target index (TI) values for urgent and routine patients with the constructive algorithm, where each column (row) represents a different TI value for urgent (routine) patients

per patient. The values in bold are the ones where there is no significant statistical evidence that any of the values found by the other configurations are better for that objective and are, therefore, considered the best values found. When there is statistical evidence that two values are different from each other, they are said to be *significantly different*, and if their difference is considered large, they are said to be considerably different. This form of evaluation is also used in the succeeding sections. Experiments are run using all configurations on each of the 33 instances described in Section 4.1.

It can be observed that the majority of the best values for all objective functions are achieved when scheduling routine patients as close as possible to their release date ($TI = 0.00$), while the value of TI for urgent patients did not have a large impact on any of the objective functions which are related to a due date. For the squared waiting time, the best results are achieved by scheduling all patients as close as possible to their release dates. More importantly, this approach did not achieve any statistically significant improvements on any of the objective functions if compared with the simpler obvious approach of scheduling patients on the earliest date possible (TI values of 0 for all patients).

These results are contradictory with the ones found by Petrovic et al. (2006), in which the approach that performs better is the backward algorithm, which tries to schedule patients starting on their due dates (corresponding to using a TI value of 1 for both urgent and routine patients). The author believes that the results differ because of the way the two experiments were conducted. In this work, each experiment includes patient data of 1 year, starting from an empty schedule, and the measures regarding patients from the first 6 months are discarded to avoid start-up effects. In the cited paper, each experiment consists of one month and starts from a previously filled schedule with a pre-defined utilisation level set for linacs. Three levels were introduced, light, normal and heavy, within which the utilisation of the available linac capacity on the first day is 90, 95 and 98%, respectively, on the second day 75, 80 and 85, respectively, and then it decreases by 5 on every following day.

Since the best results are achieved with a value of $TI = 0$ for all patients, these values are used in the experiments in the remainder of this chapter.

4.3 Utilisation Threshold Approach

The second approach introduces limited machine usage, so that when the total utilisation on a machine reaches a proportion specified for a given waiting list status, no more sessions of patients of that status can be scheduled on that machine on that day. This way, machine capacity is reserved for future patients of higher priority.

To implement this approach, the parameter *threshold proportion* ($TP : 0\% \leq TP \leq 100\%$) of machine utilisation is defined for patients of each waiting list status. The values used for TP are 100%, 98%, 96%, 94%, 92% and 90% for urgent and routine patients, whilst it is set to 100% for emergency patients. Since linacs are open from 8:45 to 18:00 (see Section 4.1), a 90% threshold for routine patients can be translated as a reservation of 55 minutes for future emergency and urgent patients.

Some combinations of these values will lead to situations where the available capacity for routine patients is greater than for urgent patients, which is not desirable taking into account their due dates and relative importance. To avoid these situations, only the combinations of values where the threshold for urgent patients is greater or equal to the threshold for routine patients are considered.

To identify the configuration which achieved the best results in these experiments, the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) (Hwang & Yoon, 1981) is used. TOPSIS is a ranking technique which gives higher scores to configurations close to an ideal point and lower scores to configurations close to a negative-ideal, where an ideal point (negative-ideal point) is a configuration where the average value achieved for all objective functions are at their best (worst) values found. It is used in this work when there is no configuration which achieves the best results for all objectives.

The TOPSIS measure is calculated as follows. Consider a decision matrix of Y columns and Z rows, where element v_{nm} corresponds to the average value achieved for objective function m with configuration n. The first step is to construct a weighted normalised decision matrix. The values achieved are divided by the norm of the vector formed by the values achieved for objective function m in all configurations, and multiplied by a weight ϵ_m associated to objective function m. Each element v'_{nm} of the weighted normalised decision matrix can be calculated as

$$v'_{nm} = \epsilon_m \ \frac{v_{nm}}{\sqrt{\displaystyle\sum_{n'=1}^{Z} v_{n'm}^2}}, \tag{4.3}$$

where the weights ϵ_m are decided by the decision maker, such that $\sum_{m=1}^{Y} \epsilon_m = 1$.

The next step is to calculate the distance between each point and the ideal and negative-ideal points measured by the Y-dimensional Euclidean space. The distances of the values achieved for configuration n to the ideal and negative ideal points are calculated by

$$D_n^\star = \sqrt{\sum_{m=1}^{Y}(v'_{nm} - v'^\star_m)^2} \quad \text{and} \tag{4.4}$$

$$D_n^- = \sqrt{\sum_{m=1}^{Y}(v'_{nm} - v'^-_m)^2}, \tag{4.5}$$

respectively, where $v'^\star_m$ and v'^-_m are the best and worst values found for objective function m, respectively.

Finally, the TOPSIS score of configuration n is given by

$$D_n = \frac{D_n^-}{D_n^\star + D_n^-}. \tag{4.6}$$

As can be noticed, the score D_n approaches 1 or 0 as the values achieved by configuration n approach the best or worst values found, respectively. The results of experiments varying the utilisation threshold proportion are presented in Table 4.2.

The best results for the breach objective function are achieved when the utilisation threshold is set to high values for urgent patients (100 and 98) and to slightly lower values for routine patients (94 and 92). These utilisation values favour mostly urgent patients who do not meet their breach date when no utilisation threshold is used for any waiting list status. The treatment of routine patients is slightly postponed, but the number of routine patients who miss their breach date remains nearly constant.

For the JCCO maximum acceptable date, the best results are achieved when TP is set also to high values for urgent patients (100 and 98) and to the lowest values used for routine patients (90). Emergency and urgent patients who do not meet their JCCO maximum acceptable targets with a TP of 100 are able to meet them with these utilisation threshold values. As with the breach date, the number of routine patients who miss their JCCO maximum acceptable dates is roughly the same as in the case with no threshold.

For the JCCO good practice, the results are slightly different. The best results are obtained when using the lowest values for both urgent and routine patients. These values will

		1.00	0.98	0.96	0.94	0.92	0.90
1.00	Breach (%)	34.95	–	–	–	–	–
	JMax (%)	56.53	–	–	–	–	–
	JGood (%)	93.95	–	–	–	–	–
	Waiting	1,347	–	–	–	–	–
	TOPSIS	0.54	–	–	–	–	–
0.98	Breach (%)	34.54	33.60	–	–	–	–
	JMax (%)	56.48	53.11	–	–	–	–
	JGood (%)	93.96	91.39	–	–	–	–
	Waiting	1,300	1,139	–	–	–	–
	TOPSIS	0.58	0.72	–	–	–	–
0.96	Breach (%)	33.19	33.15	34.86	–	–	–
	JMax (%)	55.81	52.72	50.35	–	–	–
	JGood (%)	93.82	91.18	88.86	–	–	–
	Waiting	1,170	1,109	1,258	–	–	–
	TOPSIS	0.68	0.75	0.66	–	–	–
0.94	Breach (%)	**29.24**	31.54	34.61	36.34	–	–
	JMax (%)	51.36	51.47	50.16	50.91	–	–
	JGood (%)	92.70	90.62	88.84	**88.30**	–	–
	Waiting	**977**	1,023	1,227	1,423	–	–
	TOPSIS	0.84	0.82	0.69	0.52	–	–
0.92	Breach (%)	**29.73**	**29.78**	30.94	35.28	38.19	–
	JMax (%)	44.88	45.58	47.89	50.33	52.14	–
	JGood (%)	90.09	89.50	88.80	**88.29**	**88.21**	–
	Waiting	1,038	1,042	1,077	1,307	1,734	–
	TOPSIS	**0.93**	0.92	0.86	0.62	0.25	–
0.90	Breach (%)	30.66	30.68	30.73	32.09	37.96	39.34
	JMax (%)	**43.69**	**43.90**	44.82	48.26	51.92	53.44
	JGood (%)	89.16	88.76	88.45	**88.30**	**88.25**	88.30
	Waiting	1,135	1,137	1,143	1,185	1,686	1,977
	TOPSIS	0.85	0.85	0.84	0.77	0.29	0.09

Table 4.2: Results obtained varying the linac utilisation threshold (TP) values for urgent and routine patients with the constructive algorithm, where each column (row) represents a different TP value for urgent (routine) patients

reserve most of the space available for emergency patients, ensuring that there is enough time for them to comply with their JCCO good practice dates. The number of urgent and routine patients who do not meet this due date stays roughly the same. It should be reminded that it is impossible significantly to increase or decrease the numbers of urgent and routine patients who meet the JCCO good practice due date, since, as stated in Section 3.2, it is impossible for around 96% of them to meet this due date due to long pre-treatment periods. The improvement in this objective function is due to the increase in the number of emergency patients who are able to meet the due date with the lower utilisation threshold values.

By analysing these results, it is possible to see a trend in the value of the threshold and of the due date objective functions, in which the lower utilisation threshold values achieve better results for the tighter due dates. These values enable patients of higher priority to meet their due dates without causing more patients of lower priority to miss theirs. Since it is already impossible for the majority of urgent and routine patients to meet the JCCO good practice due date, their objective functions cannot worsen by much. However, it is possible to improve the weighted number of patients who miss this due date by increasing the number of high priority patients who meet it. For the due dates which are easier to meet, it is enough to impose a slightly higher threshold for routine patients in order to ensure that there is space for emergency and urgent patients to meet the dates.

Since the TP values of 100%/92% for urgent/routine patients achieved the highest rank in TOPSIS, these values are used in the remainder of this chapter.

4.4 Schedule Creation Day (SCD) Approach

In the previous sections, patients have their schedules created on the same day as their decision to treat is made. However, if schedules are not created immediately, an accumulation of the number of patients to be scheduled would increase the search space for a given day, enabling the algorithm to find a better schedule. In this section, an approach is investigated which varies the days of the week when schedules are allowed to be created.

Each patient is assigned a set of weekdays when the creation of a schedule for that patient is allowed. If a patient has a decision to treat on a day when the creation of a schedule is not allowed, the schedule will be created on the first following allowed day. Obviously, the search space becomes larger and it may lead to schedules of higher quality. On the other hand, this approach involves delaying the creation of schedules, which may cause the delay of the start of treatment for some patients and, in turn, lead to schedules of lower quality.

		5	3	2	1
5	Breach (%)	29.73	29.72	29.71	29.71
	JMax (%)	44.88	44.86	44.86	44.77
	JGood (%)	90.09	90.06	90.06	89.90
	Waiting	1,038	1,038	1,037	1,038
	TOPSIS	0.05	0.08	0.12	0.28
3	Breach (%)	29.74	29.74	29.72	29.74
	JMax (%)	44.86	44.79	44.80	44.76
	JGood (%)	90.12	90.10	90.03	**89.87**
	Waiting	**1,037**	**1,037**	**1,036**	**1,036**
	TOPSIS	0.11	0.19	0.22	0.31
2	Breach (%)	29.66	29.66	29.67	29.70
	JMax (%)	44.71	44.66	44.77	44.67
	JGood (%)	90.00	89.96	89.98	**89.85**
	Waiting	**1,035**	**1,035**	**1,035**	**1,036**
	TOPSIS	0.45	0.52	0.38	0.47
1	Breach (%)	**29.59**	**29.58**	**29.56**	**29.57**
	JMax (%)	44.67	**44.63**	44.72	**44.61**
	JGood (%)	90.06	90.04	90.07	89.88
	Waiting	**1,031**	**1,031**	**1,031**	**1,031**
	TOPSIS	0.76	0.82	0.74	**0.92**

Table 4.3: Results obtained varying the schedule creation days for urgent and routine patients with the constructive algorithm, where each column (row) represents a different SCD value for urgent (routine) patients

For each waiting list status, a parameter called Schedule Creation Day (SCD) is specified. The SCD values considered are 5 (every weekday), 3 (on Mondays, Wednesdays and Fridays), 2 (on Tuesdays and Fridays) and 1 (only on Fridays) for urgent and routine patients and fixed as 5 for emergency patients. Results obtained for all combinations of SCD values are given in Table 4.3.

By using different values of the SCD parameter, the constructive algorithm is able to achieve small improvements in all objective function values. As the radiographer waits for an accumulation of patients before creating a schedule, it is possible to find a better solution.

Although the improvements are very small, the patients who benefit are emergency and urgent patients, which is an incentive for using this approach. In addition, creating schedules only once per week using the constructive approach can decrease the workload of the radiographer, since the scheduling of patients would become a weekly procedure.

As the SCD values of 1 for both urgent and routine patients achieved highest rank in TOPSIS, they are used in the experiments in the remainder of this chapter.

4.5 Maximum Number of Days in Advance (MNDA) Approach

The approach described in this section can be seen as a "wait and see" approach. Two important dates in radiotherapy treatment scheduling are the date of the decision to treat and the release date. Since these two dates can be very far apart, one possibility for achieving a better schedule is not to create a schedule for patients immediately when their decision to treat is made (or even on the first Friday after the decision to treat, as in the case when creating schedules once per week), but to wait for the release date to come closer, towards the end of their pre-treatment phase. This might give a better chance of good quality schedules for patients who will arrive in the near future, while still obtaining good quality schedules for current patients.

The fourth constructive approach consists of introducing a parameter called Maximum Number of Days in Advance ($MNDA$) to limit the creation of schedules based on the patient's release date. Given a patient j, the date from which his/her schedule may be created is defined as:

$$sch_j = \max\{b_j, r_j - MNDA\}, \tag{4.7}$$

where b_j corresponds to the date when the decision to treat is made and r_j corresponds to the patient's release date. This date is referred to as *scheduling date*. The time-line can be seen in Figure 4.1.

The values used in the experiments for the $MNDA$ parameter of urgent and routine patients are:

- ∞ (infinity) - the schedule is created as soon as the decision to treat is made,
- 21, 14, 7 - the schedule is created when the release date is within 21, 14 or 7 days respectively,

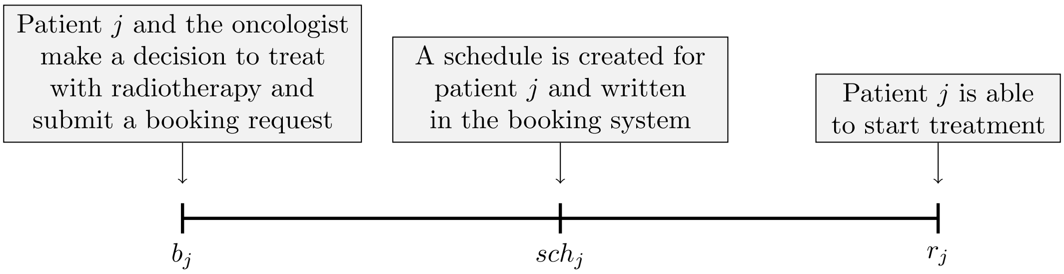


Figure 4.1: Time-line for the scheduling dates of patient j, where b_j is the date when the decision to treat is made and r_j is the release date.

- and 0 - the schedule is created on the release date or afterwards.

For emergency patients, the $MNDA$ value is fixed at ∞.

Experiments are run with every combination of these values for urgent and routine patients. Some combinations might lead to situations where the schedule of routine patients receives a higher priority than the schedule of an urgent patient. To avoid these situations, only the combinations of values where the $MNDA$ value for urgent patients is greater or equal to the value for routine patients are considered. Results are presented in Table 4.4.

The variation of $MNDA$ values achieved considerably better results than the variation of SCD values. For all objective functions, the best results are found when creating schedules for patients only when their release date is within one week away ($MNDA$ value of 7). This way, it is possible to postpone the creation of schedules for patients until their release date is closer in order to leave room for future patients (either high or low priority) who are able to start treatment right away.

It is possible to see a large difference in the objective function values between creating schedules when the release date is within 7 days and on the release date or afterwards ($MNDA$ values 7 and 0), specially when looking at the breach date. This can be explained by the fact that when creating schedules once per week (SCD values of 1 used in these experiments) and on the release date or after it ($MNDA$ value 0), there might be a patient whose schedule is created only after the breach date, while if the schedule is created when the release date is within 7 or more days, it will be created before the breach date thus not violating it. For example, consider

		∞	21	14	7	0
∞	Breach (%)	29.57	–	–	–	–
	JMax (%)	44.61	–	–	–	–
	JGood (%)	89.88	–	–	–	–
	Waiting	1,031	–	–	–	–
21	Breach (%)	28.03	28.03	–	–	–
	JMax (%)	43.49	43.48	–	–	–
	JGood (%)	89.66	89.65	–	–	–
	Waiting	903	903	–	–	–
14	Breach (%)	27.40	27.41	27.30	–	–
	JMax (%)	43.07	43.04	42.65	–	–
	JGood (%)	89.58	89.57	89.56	–	–
	Waiting	845	845	840	–	–
7	Breach (%)	26.79	26.80	26.48	**26.34**	–
	JMax (%)	42.40	42.37	42.06	**41.83**	–
	JGood (%)	**89.39**	**89.39**	89.44	**89.39**	–
	Waiting	799	799	790	**786**	–
0	Breach (%)	33.99	34.00	33.98	33.70	38.21
	JMax (%)	43.65	43.62	43.48	43.37	50.52
	JGood (%)	90.34	90.33	90.59	90.51	89.96
	Waiting	869	869	859	849	943

Table 4.4: Results obtained for different maximum numbers of days in advance for urgent and routine patients with the constructive algorithm, where each column (row) represents a different $MNDA$ value for urgent (routine) patients

the case where schedules for routine patients are created once per week on days 1, and 8, and that a routine patient has release and breach dates on days 2 and 7 respectively. When creating schedules on the release date or afterwards, the schedule of this patient will be created on day 8, thus ensuring a violation of the breach date. If schedules are created within 7 days of the release date, the schedule will be created on day 1 and the breach date may be met. This is, of course, provided there is enough available time on the linacs.

4.6 Comparison of Constructive Approaches

So far, each new approach is tested in combination with the best configuration found for the previous approaches. By using this method, only a fraction of the combinations of the approaches is investigated. This may lead to the final configuration found being sub-optimal. With that in mind, additional configurations are experimented with in this section.

In order to decrease the number of experiments to a manageable amount, only the utilisation threshold, SCD and $MNDA$ approaches, presented in Sections 4.3, 4.4 and 4.5, respectively, are considered. The target approach presented in Section 4.2 does not improve the objective functions and is, therefore, not investigated further.

Like before, the values used in each approach for urgent and routine patients are:

- 100%, 98%, 96%, 94%, 92% and 90% for TP for urgent and routine patients, where the value of TP for urgent patients is greater or equal to the value of TP for routine patients,
- 5 (every weekday), 3 (on Mondays, Wednesdays and Fridays), 2 (on Tuesdays and Fridays) and 1 (only on Fridays) for SCD,
- and ∞ (infinity), 21, 14, 7 and 0 for $MDNA$, where the $MNDA$ value for urgent patients is greater or equal to the value for routine patients.

For emergency patients, the values for TP, SCD and $MNDA$ are set to 100%, 5 and ∞, respectively.

The total number of combinations sums up to 5040 (21 valid configurations for TP, 16 for SCD and 15 for $MNDA$). However, not all combinations are worthy examining. To choose which configurations should be examined, the definition of efficient configurations of parameters is introduced. A configuration 1 is said to be efficient iff there is no configuration 2 such that:

- there is no statistical evidence that the values found with configuration 1 are better than those found with configuration 2 for any objective function, and
- there is statistical evidence that the values found with configuration 2 are better than those found with configuration 1 for at least one objective function.

Forty efficient configurations are identified using the method above and presented in Table 4.5. In addition, results from the configuration with the original values of TP, SCD and $MNDA$ for urgent/routine patients (100%/100%, 5/5 and ∞/∞) are also presented.

TP		SCD		$MNDA$		Breach	JMax	JGood	Waiting	TOPSIS
Urg.	Rou.	Urg.	Rou.	Urg.	Rou.	(%)	(%)	(%)		
1.00	1.00	5	5	∞	∞	34.95	56.53	93.95	1,347	0.47
1.00	0.94	5	5	∞	0	26.00	44.47	90.40	753	0.95
1.00	0.94	3	5	∞	0	25.95	44.43	90.39	751	0.95
1.00	0.94	1	5	∞	0	25.75	44.71	90.43	746	0.95
1.00	0.94	5	5	21	0	26.01	44.44	90.39	753	0.95
1.00	0.94	3	5	21	0	25.96	44.39	90.39	751	**0.95**
1.00	0.94	2	5	21	0	25.89	44.55	90.44	749	0.95
1.00	0.94	1	5	21	0	25.76	44.68	90.43	746	0.95
1.00	0.94	1	5	14	0	25.80	45.14	90.89	741	0.95
1.00	0.92	1	1	7	7	26.34	**41.83**	89.39	786	0.95
1.00	0.94	5	5	0	0	**25.22**	45.46	90.99	**731**	0.95
0.98	0.98	1	5	∞	0	25.98	46.89	90.22	753	0.93
0.98	0.94	3	5	∞	0	26.41	44.69	89.45	762	0.95
0.98	0.9	5	5	∞	∞	30.68	43.90	88.76	1,137	0.68
0.98	0.98	1	5	21	0	25.99	46.86	90.22	753	0.93
0.98	0.96	5	5	21	0	26.32	45.43	89.98	754	0.94
0.98	0.96	3	5	21	0	26.29	45.52	89.96	754	0.94
0.98	0.96	1	5	21	0	26.24	45.83	89.88	755	0.94
0.98	0.94	5	5	21	0	26.44	44.60	89.49	762	0.95
0.98	0.94	3	5	21	0	26.42	44.66	89.45	762	0.95
0.98	0.92	1	1	14	14	27.04	42.63	88.82	825	0.92
0.96	0.96	5	5	∞	0	27.10	44.71	88.58	795	0.93
0.96	0.96	3	5	∞	0	27.17	44.74	88.54	797	0.92
0.96	0.96	2	5	∞	0	27.15	44.81	88.58	799	0.92
0.96	0.94	5	5	∞	0	27.41	44.19	88.48	803	0.92
0.96	0.94	3	5	∞	0	27.41	44.27	88.49	804	0.92
0.96	0.96	5	5	21	0	27.12	44.68	88.58	795	0.93
0.96	0.96	2	5	21	0	27.16	44.78	88.57	799	0.92
0.96	0.94	5	5	21	0	27.42	44.16	88.48	804	0.92
0.96	0.94	3	5	21	0	27.42	44.24	88.49	804	0.92
0.96	0.94	1	5	21	0	27.40	44.45	88.43	804	0.92
0.96	0.94	1	5	7	0	28.06	44.83	88.38	823	0.90
0.94	0.94	5	5	∞	0	28.45	45.09	88.06	862	0.87
0.94	0.94	3	5	∞	0	28.50	45.18	88.09	864	0.87
0.94	0.94	5	5	21	0	28.46	45.06	88.05	862	0.87
0.94	0.94	3	5	21	0	28.52	45.15	88.09	864	0.87
0.94	0.94	2	5	21	0	28.49	45.16	88.06	864	0.87
0.94	0.94	2	5	14	0	28.79	44.97	88.00	875	0.86
0.94	0.94	1	5	7	0	29.47	45.34	87.95	904	0.84
0.94	0.94	5	5	0	0	29.68	45.54	87.89	913	0.83
0.94	0.94	1	5	0	0	34.05	51.18	**87.79**	1,003	0.70

Table 4.5: Results obtained for different maximum numbers of days in advance for urgent and routine patients with the constructive algorithm

Two things are quickly seen by analysing Table 4.5: the majority of efficient configurations include the values 5 and 0 for SCD and $MNDA$, respectively, of routine patients, which indicates that these values are the best for those parameters. Furthermore, there does not seem to be much variation in the values of the objective functions for these configurations, other than what has already been mentioned in the previous sections.

The largest TOPSIS value was found with the configuration with values of 100%/94%, 3/5 and 21/0 for TP, SCD and $MNDA$ for urgent/routine patients. This is considered the best configuration of the constructive algorithm and it will be compared with the other algorithms in the next chapters.

4.7 Summary

Four simple approaches are presented in this chapter to investigate whether it is possible to solve the proposed problem using a constructive algorithm or how good the schedules they generate can be. An experimental data set is also derived from the combined data set. It is generated in a way to comply with the same requirements presented in Section 3.3 and it is to be used in all the experiments in this work.

The experiments suggest that it is possible to improve the quality of a schedule if an utilisation threshold is defined to limit the machine utilisation of routine patients. The threshold that achieves the best results in the experiments is 92% for routine patients and no limit set for other patients.

Waiting before creating schedules for patients, instead of immediately scheduling then when they arrive, also achieves good results in the experiments. Creating schedules for routine and urgent patients only once per week and only when their release date is within one week achieves the best results in the experiments presented in this chapter.

It should be noted that the approaches are described here in the order they were developed. If the experiments are conducted in a different order, the configuration to achieve the best results might be slightly different. However, running all combinations of parameter values is computationally infeasible, since it would require an immense amount of CPU time.

It should be noted that the constructive algorithm presented here does not perform any "clever" optimisation to improve a schedule. In addition, it can supply only one solution to the decision maker. A different algorithm, which performs a more extensive search and is capable of improving the schedule for a given day can prove interesting. However, most of the approaches presented in this chapter can be used within this more sophisticated algorithm. Such algorithm is proposed and investigated in the succeeding chapters.

"Essentially, all models are wrong, but some are useful."

George E. P. Box (1919–)

Chapter 5

Integer Linear Programming

In this chapter, a new Integer Linear Programming (ILP) (Nemhauser & Wolsey, 1989) model is presented for the radiotherapy treatment scheduling problem. Instead of using the constructive algorithm to build a schedule at the end of each day, as done in the previous chapter, an instance of an ILP model is populated with patients available for scheduling, and an integer programming solver is used to find a solution for the instance.

Usually, integer programming solvers employ a Branch-and-bound (B&B) (Land & Doig, 1960; Balas, 1965; Lawler & Wood, 1966) algorithm. A B&B solves a minimisation integer program by relaxing the integrality requirements and solving a sequence of progressively more restricted linear problems, using the best integer solution found as an upper bound to prune nodes of sub-problems with a lower bound larger than the upper bound. The algorithm terminates when there are no more sub-problems with a lower bound smaller than the current upper bound.

Solvers can also make use of cutting plane algorithms (Gomory, 1958; Dantzig, 1959) to decrease the search space. These involve including constraints to the problem to remove fractional solutions without removing the optimal integer solution. The combination of B&B with cutting planes in each node, known as Branch-and-Cut (Padberg & Rinaldi, 1987; Balas et al., 1996a,b), is also commonly used. In this work, all experiments are performed using CPLEX 12.2

developed by IBM (International Business Machines, 2010), as this is one of the most frequently used and well known ILP solvers in the literature. As its default behaviour, it uses a B&B with cutting planes on the root node and it may use cutting planes in the other nodes depending on its effectiveness.

Unlike the constructive approaches investigated in the previous chapter, the scheduled obtained by solving an ILP is guaranteed to be optimal for the patients currently being scheduled. However, there is no guarantee that the schedule is optimal (or even good) for future patients. In order to try to obtain a schedule which is also good for future patients, additional measures must be adopted. The Schedule Creation Day and the Maximum Number of Days in Advance approaches presented in Sections 4.5 and 4.4, respectively, are applied with the ILP model and the results are analysed. Additional experiments are performed with different combinations of values for the parameters of these two approaches. Furthermore, the Utilisation Threshold presented in Section 4.3 is extended to consider a different threshold for each day, and an analysis is made on the correlation of the values achieved for each objective function by different threshold values.

Furthermore, the performance of the model is analysed for different problem sizes. The future increase in the rate of patient arrival foreseen by the hospital staff is considered, as well as the new linacs which should be acquired in the next two years. However, these issues are relevant for any other hospital. These modifications in the problem size are combined with two possible changes in the linac allocation policy. The goal of these changes is to relax the constraints to augment the feasible search space and investigate if better schedules can be found. The changes investigated in this chapter are 1) to consider all types of radiation emitted by each linac and 2) to allow sessions of the same patient to be scheduled in different linacs.

5.1 Radiotherapy Treatment Scheduling Model

The radiotherapy treatment scheduling problem described in Section 3.1, can be formulated as an Integer Linear Programming (ILP) model. The following notation is used:

- M: number of linacs,

- i: index for linacs ($i = 1, \ldots, M$),
- N: number of patients available for scheduling,
- j: index for patients available for scheduling ($j = 1, \ldots, N$),
- $\mathcal{M}_j$: set of machines with the radiation types required for patient j ($\mathcal{M}_j \subseteq \{1, \ldots, M\}$, $\mathcal{M}_j \neq \emptyset$),
- $\mathcal{W}_j$: set containing the days of the week when patient j is allowed to have his/her first session ($\mathcal{W}_j \subseteq \{\text{Monday}, \ldots, \text{Sunday}\}, \mathcal{W}_j \neq \emptyset$),
- w_j: relative importance (weight) assigned to patient j ($w_j > 0$),
- b_j: date when the decision to treat of patient j is made,
- r_j: release date of patient j,
- d_j^1: breach date by which patient j should start the treatment as established by the Department of Health (2005),
- d_j^2: maximum acceptable date by which patient j should start the treatment as established by the Joint Council for Clinical Oncology (1993),
- d_j^3: good practice date by which patient j should start the treatment as established by the Joint Council for Clinical Oncology (1993),
- T: number of days in the planning horizon,
- k: index for days in the planning horizon ($k = 1, \ldots, T$),
- q_k: day of the week of day k ($q_k \in \{\text{Monday}, \ldots, \text{Sunday}\}$),
- C_{ik}: total capacity of linac i on day k given in minutes,
- U_{ik}: used capacity of linac i on day k by patients previously scheduled given in minutes,
- S_j: number of sessions required for patient j,
- l: index for sessions of patient j ($l = 1, \ldots, S_j$),

- p_{jl}: duration of session l of patient j given in minutes,
- u_{jkl}: number of days patient j must wait between sessions l and $l+1$ if session l is scheduled on day k ($u_{jkl} \geq 0$).

Decision variables $\mathbf{x}$ are defined as follows:

$$x_{ijkl} = \begin{cases} 1 & \text{if session } l \text{ of patient } j \text{ is scheduled on day } k \text{ on linac } i, \\ 0 & \text{otherwise.} \end{cases}$$

The first constraints are presented to ensure that sessions are not scheduled on any invalid machine or day. Constraint (5.1) imposes that sessions of patient j are not scheduled on machines that do not emit the types of radiation required for patient j, while constraints (5.2-5.4) ensure that patients are not scheduled on invalid days. Constraint (5.2) imposes that any session cannot be scheduled before the patient's release date, constraint (5.3) guarantees that the first session of each patient is not on an invalid day of the week, and constraint (5.4) ensures that no session of any patient other than the first one can take place on the first day of the planning horizon.

$$x_{ijkl} = 0 \qquad i = 1, \ldots, M, i \notin \mathcal{M}_j, j = 1, \ldots, N, k = 1, \ldots, T, l = 1, \ldots, S_j \tag{5.1}$$

$$x_{ijkl} = 0 \qquad i = 1, \ldots, M, j = 1, \ldots, N, k = 1, \ldots, T, k < r_j, l = 1, \ldots, S_j \tag{5.2}$$

$$x_{ijk1} = 0 \qquad i = 1, \ldots, M, j = 1, \ldots, N, k = 1, \ldots, T, q_k \notin \mathcal{W}_j \tag{5.3}$$

$$x_{ij1l} = 0 \qquad i = 1, \ldots, M, j = 1, \ldots, N, l = 1, \ldots, S_j, l > 1 \tag{5.4}$$

Each pair of consecutive sessions of the same patient must be scheduled u_{jkl} days apart, depending on the day k when session l is scheduled. To ensure that session $l+1$ is scheduled

u_{jkl} days after session l, constraint (5.5) is included.

$$x_{ijk'l'} = x_{ijkl} \qquad k' = k + u_{jkl}, k' \leq T, l' = l + 1,$$
$$i = 1, \dots, M, j = 1, \dots, N, k = 1, \dots, T, l = 1, \dots, S_j - 1, \quad (5.5)$$

It is necessary to guarantee that all sessions are scheduled, and that each session is scheduled on exactly one day and one linac. Constraint (5.6) imposes this restriction.

$$\sum_{i=1}^{M} \sum_{k=1}^{T} x_{ijkl} = 1 \qquad j = 1, \dots, N, l = 1, \dots, S_j \quad (5.6)$$

Finally, the available capacity on linacs must be respected. Constraint (5.7) ensures that the total time used by sessions on day k on linac i does not exceed the linac capacity for that day.

$$\sum_{j=1}^{N} \sum_{l=1}^{S_j} p_{jl}\ x_{ijkl} \leq C_{ik} - U_{ik} \qquad i = 1, \dots, M, k = 1, \dots, T \quad (5.7)$$

The objective functions to be minimised in this work are presented in their order of importance and defined below. This order has been decided following hospital staff preference.

- the number of patients who miss the breach date

$$f_1(\mathbf{x}) = \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=d_j^1+1}^{T} x_{ijk1}, \quad (5.8)$$

- the weighted number of patients who miss the JCCO maximum acceptable target

$$f_2(\mathbf{x}) = \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=d_j^2+1}^{T} w_j\ x_{ijk1}, \quad (5.9)$$

- the weighted number of patients who miss the JCCO good practice target

$$f_3(\mathbf{x}) = \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=d_j^3+1}^{T} w_j\ x_{ijk1}, \quad (5.10)$$

- the sum of the weighted squared waiting times

$$f_4(\mathbf{x}) = \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=b_j+1}^{T} (k - b_j)^2 \; w_j \; x_{ijk1}. \tag{5.11}$$

  It should be noted that, even though the squared waiting time is being calculated, no decision variables are in fact squared and the model remains linear.

In order to handle multiple objectives optimisation, a lexicographical ordering (Steuer, 1986; Yu, 1989) is used. The set of Y objectives is indexed so that objective m is more important than objective $m+1$. A lexicographical ordering preference is defined as follows: solution $\mathbf{x}^1$ is preferred to solution $\mathbf{x}^2$ iff $f_1(\mathbf{x}^1) < f_1(\mathbf{x}^2)$ or there is some $m \in \{2, \ldots, Y\}$ so that $f_m(\mathbf{x}^1) < f_m(\mathbf{x}^2)$ and $f_{m'}(\mathbf{x}^1) = f_{m'}(\mathbf{x}^2)$ for $m' = 1, \ldots, m-1$.

5.2 Deriving the Planning Horizon

The model presented is dependent on the duration of the planning horizon T, which must be supplied as input. To calculate the value of T, a schedule is first created using the constructive algorithm presented in Chapter 4. Patients are ordered lexicographically by their release date, waiting list status and number of required sessions. Following this order, a schedule is created for each patient starting on the earliest day possible. The value of T is determined by the latest session in this schedule.

However, if this horizon is too short, the optimal schedule from the search space might be removed. In order to augment the search space, the value of T is increased by a pre-determined number of days, referred to as *slack*. To decide on a appropriate value for this slack, experiments are run to investigate the effectiveness of different values. The values experimented with are 0, 7, 14, 21 and 28 days.

As before, each experiment consists of a simulation of the everyday scheduling of a hospital for one and a half years. At the end of the day, the radiographer creates a schedule by populating an instance of the model with the patients who are available for scheduling and then using the lexicographic algorithm in conjunction with an ILP solver to find a solution. On each day the algorithm is given a time limit of 10 minutes to create a schedule. This limit was suggested by the hospital staff, as execution times longer than that would be considered a problem to them. If the algorithm does not terminate within this limit, the best solution found so far is adopted.

The solution created using the constructive approach is used as a starting point by the chosen solver. Three improvements are made to the algorithm in order to speed up the process of finding a schedule each day. The first involves identifying when the solution found by the constructive algorithm is ideal, where an ideal solution is the one where all objective functions are at their individual optimal values. When the constructive algorithm finds an ideal solution, there is no need to search any further. A fast algorithm capable of identifying some of the ideal solutions is used. It verifies if all patients start their treatment on the earliest day possible considering the current machine availability. Since the only possible way to improve an objective is if at least one patient is able to start their treatment on an earlier date, it is easy to see that such schedule cannot be improved in any way. However, it should be noted that not all ideal solutions can be identified by this algorithm.

The second improvement consists of dividing the problem into sub-problems. As described in Section 4.1, each patient can be scheduled only on a specific set of linacs, such that the schedule of patients who require one type of radiation has no influence on the schedule of patients who require other types. Therefore, it is safe to split the problem into three sub-problems, each consisting only of patients who require one specific type of radiation (low energy photon, high energy photon or electron) and the linacs where these patients must be scheduled. Each sub-problem is solved individually and the schedules found are combined to form a complete schedule. The time limit of 10 minutes is equally divided amongst the three sub-problems.

A third improvement consists of removing an objective function from the lexicographical ordering when all solutions in the feasible search space yield the same value for it. To identify such situations, the minimum (f_m^{min}) and maximum (f_m^{max}) values of each objective function m

are calculated. If these two values are equal, there is no need to consider objective function m on that day.

The calculation of these two values is quite simple. Since all objectives are non-decreasing functions of the start date of each patient, f_m^{min} is calculated by considering that all patients are scheduled on the earliest feasible date considering current machine availability, and f_m^{max} is calculated by considering that all patients are scheduled on the latest feasible date such that all sessions are scheduled inside the chosen planning horizon T. It is not unusual for f_m^{min} to have the same value as f_m^{max} when f_m is one of the objective functions related to a due date. Amongst all possibilities, two situations are fairly common:

- When the earliest feasible start date is after the JCCO good practice target d_j^3 for all patients and, therefore, no patient is able to meet this target. This can often happen when all patients to be scheduled are urgent and routine with long pre-treatments.
- When the last feasible start date of all patients is before the breach date d_j^1 for all patients and, therefore, no patients can miss this target. This can often happen when all patients to be scheduled are emergency and urgent with only one session and short pre-treatments.

It should be noted that the fourth objective function (weighted squared waiting time) is never removed from the lexicographical ordering, as this objective is a strictly increasing function of the start date.

Each value for the slack is run on the 33 instances described in Section 4.1. Table 5.1 shows the average values found for each objective function. As before, MWW is used with bootstrapping to make a statistical analysis of the results, and the values considered the best for each objective function are presented in bold. The average required CPU time per day is always below 1 second.

It is possible to see that there is no significant difference among the values found for each objective function. In order to make a deeper analysis of the performance of the algorithm, Table 5.2 shows the average number of days in the planning horizon, the percentage of runs when

Slack	Breach (%)	JMax (%)	JGood (%)	Waiting
0	**34.42**	**56.34**	**93.91**	**1,312**
7	**34.38**	**56.29**	**93.90**	**1,309**
14	**34.38**	**56.32**	**93.91**	**1,308**
21	**34.36**	**56.32**	**93.91**	**1,308**
28	**34.36**	**56.33**	**93.91**	**1,308**

Table 5.1: Results obtained for different values of slack with the ILP model

Slack	Planning Horizon	Ideal (%)	Variables per Run	Constr. per Run
0	76.29	80	214	57
7	83.01	79	240	62
14	90.04	79	272	68
21	97.05	79	303	73
28	104.06	79	333	79

Table 5.2: Performance details for different values of slack with the ILP model

the ideal solution was found by the constructive approach and identified, and the number of variables and constraints per run.

The ideal solution is found by the constructive approach in around 80% of the cases for any value of slack. As the problem is split into three sub-problems (one for each radiation type), the average number of patients considered by each sub-problem is usually very small, making it easy enough for the constructive algorithm to find and identify the ideal solution.

As expected, the average size of the planning horizon grows with the size of slack. This causes a slight increase in the number of variables and constraints per run. However, the required CPU time is not noticeably affected by this increase.

In order to augment the search space in the succeeding experiments, a slack of 14 days is used.

5.3 Schedule Creation Day (SCD) Approach

As stated in the beginning of this chapter, some of the approaches used with the constructive algorithm can also be used in combination with the ILP model. This section investigates how

		5	3	2	1
5	Breach (%)	34.38	34.41	34.50	34.67
	JMax (%)	56.32	56.23	56.25	56.19
	JGood (%)	93.91	93.89	93.89	**93.84**
	Waiting	1,308	1,313	1,313	1,327
	Time (ms)	17	18	17	17
3	Breach (%)	34.06	33.99	34.16	34.27
	JMax (%)	56.17	56.07	56.16	56.06
	JGood (%)	93.92	93.89	93.90	**93.86**
	Waiting	1,288	1,284	1,296	1,307
	Time (ms)	27	37	25	34
2	Breach (%)	33.87	33.88	33.85	33.94
	JMax (%)	56.15	56.13	56.00	55.86
	JGood (%)	93.90	93.88	93.88	**93.85**
	Waiting	1,276	1,277	1,276	1,280
	Time (ms)	22	22	44	58
1	Breach (%)	**33.06**	**32.96**	**32.91**	**32.89**
	JMax (%)	55.76	55.67	55.63	**55.49**
	JGood (%)	**93.86**	**93.84**	**93.84**	**93.83**
	Waiting	**1,224**	**1,222**	**1,221**	**1,217**
	Time (ms)	52	53	107	173

Table 5.3: Results obtained varying the schedule creation days for urgent and routine patients with the ILP model. Each column (row) represents a different SCD value for urgent (routine) patients.

the schedule creation day (SCD) approach presented in Section 4.4 performs. As before, the SCD values considered are 5 (every weekday), 3 (on Mondays, Wednesdays and Fridays), 2 (on Tuesdays and Fridays) and 1 (only on Fridays) for urgent and routine patients and fixed as 5 for emergency patients. Results are presented in Table 5.3, in addition to "Time", which is the average required CPU time per run in milliseconds when the ILP model is used.

Slight improvement is achieved for all objective functions using this approach, similarly to the constructive algorithm. In general, good results for the breach date are found when schedules are created for routine patients once a week, regardless of how often urgent patients have their schedules created. As the breach date is the least restrictive target (it is the latest target date), it is possible to achieve better results by slightly delaying the creation of schedules for patients, so that schedules are created only once a week to increase the search space. The improvement achieved is statistically significant. However, it is not very noteworthy.

Similarly for the breach, the best results for the JCCO maximum acceptable target are found when creating a schedule for urgent and routine patients once a week. The JCCO maximum acceptable target is slightly more restrictive than the breach date, and it is also possible to achieve better results for this target by creating schedules less frequently. However, the improvement achieved is even less noteworthy than that for the breach date.

Most configurations do not achieve results significantly different from each other for the JCCO good practice target. Since being able to meet this date is so rare amongst patients, as can be seen in Figure 3.7, all the configurations achieve very similar results.

For the squared waiting time, the best results are found when routine patients are scheduled once a week, similarly to the breach date. The author believes that the configurations that achieve good results for the breach date also achieve the good results for the waiting time squared because both objective functions give a much greater penalty to patients who have a large waiting time than patients for whom the waiting time is not so large.

Table 5.4 shows the average number of patients scheduled per run, the percentage of runs where the constructive algorithm found and identified the ideal solution, and the average numbers of variables and constraints per run. As expected, the number of patients per run increases with the reduction of the frequency of scheduling. It should be noted that the number of patients per run is not five times larger when creating schedules once per week than when creating schedules every day, since there are days when there are no booking requests.

The number of times when the solution found by the constructive approach is identified as the ideal solution increases as the number of patients considered drops or when schedules are created for urgent and routine patients on different days. If there are only urgent patients being scheduled, it is more likely that the schedule created for one of them will not clash with schedules for the others, since the majority of urgent patients have only one session. This situation happens often when creating schedules for urgent patients every day and for routine patients with a lower frequency (SCD values of 5 for urgent and low values for routine). When schedules are created for urgent and routine patients only once per week and the search space is at its largest, the number of ideal solutions found and identified by the constructive approach is smaller.

		5	3	2	1
5	Patients per Run	4.54	4.56	4.61	4.66
	Ideal (%)	79	79	79	78
	Variables per Run	272	341	308	351
	Constr. per Run	68	70	64	60
3	Patients per Run	4.89	6.45	5.38	6.63
	Ideal (%)	81	74	77	73
	Variables per Run	376	582	420	701
	Constr. per Run	72	85	71	81
2	Patients per Run	4.86	5.39	7.91	7.99
	Ideal (%)	80	77	70	71
	Variables per Run	238	270	715	1,047
	Constr. per Run	61	61	90	96
1	Patients per Run	5.22	6.90	8.37	12.06
	Ideal (%)	84	78	77	70
	Variables per Run	336	498	1,069	1,815
	Constr. per Run	63	73	100	124

Table 5.4: Performance details varying the schedule creation days for urgent and routine patients with the ILP model. Each column (row) represents a different SCD value for urgent (routine) patients.

As the SCD values are decremented and the search space is augmented, the numbers of variables and constraints increase. This is also expected, as there is an increase in the number of patients scheduled per run.

Similarly to the results found by the constructive algorithm in Section 4.4, the values for the SCD parameter and 1/1 for urgent/routine achieve the best results for all 4 objective functions and are used in the next sections.

5.4 Maximum Number of Days in Advance (MNDA) Approach

The next approach to be used in combination with the ILP model is the Maximum Number of Days in Advance ($MNDA$), first presented in Section 4.5. As before, the $MNDA$ values are fixed at ∞ for emergency patients, and are varied as ∞, 21, 14, 7 and 0 for urgent and routine patients. Only combinations where the value for urgent patients is greater or equal to the value for routine patients are considered. Results are presented in Table 5.5.

		∞	21	14	7	0
∞	Breach (%)	32.89	–	–	–	–
	JMax (%)	55.49	–	–	–	–
	JGood (%)	93.83	–	–	–	–
	Waiting	1,217	–	–	–	–
	Time (ms)	173	–	–	–	–
21	Breach (%)	29.31	29.33	–	–	–
	JMax (%)	54.14	54.11	–	–	–
	JGood (%)	93.62	93.61	–	–	–
	Waiting	1,028	1,028	–	–	–
	Time (ms)	134	136	–	–	–
14	Breach (%)	26.21	26.22	26.52	–	–
	JMax (%)	52.52	52.50	52.26	–	–
	JGood (%)	93.49	93.47	93.44	–	–
	Waiting	872	873	883	–	–
	Time (ms)	89	89	93	–	–
7	Breach (%)	**23.69**	**23.70**	23.91	24.44	–
	JMax (%)	49.92	**49.89**	**49.78**	50.30	–
	JGood (%)	**93.06**	**93.05**	**93.03**	93.10	–
	Waiting	**766**	**766**	774	790	–
	Time (ms)	135	136	136	102	–
0	Breach (%)	32.42	32.43	32.69	33.01	38.11
	JMax (%)	50.71	50.68	50.54	51.26	57.66
	JGood (%)	94.12	94.11	94.10	94.14	94.00
	Waiting	827	827	832	847	946
	Time (ms)	314	241	110	108	140

Table 5.5: Results obtained for different maximum numbers of days in advance for urgent and routine patients with the ILP model. Each column (row) represents a different $MNDA$ value for urgent (routine) patients.

The variation of $MNDA$ values is able to considerably improve the quality of the schedule with respect to most of the objective functions. For the breach date, the best results are obtained when creating schedules for routine patients when their release date is within 7 days, while for urgent patients when their release date is within 21 or more days. As with the constructive algorithm, there is a large difference between the values achieved for a $MNDA$ value of 7 and 0.

For both JCCO target dates, which are more restrictive than the breach date, the best results are obtained when creating schedules for routine patients when their release date is within 7 days, while for urgent patients when their release date is within 14 or more days. With these values, it is possible to give a higher priority to urgent patients, without compromising the schedules of emergency patients.

		∞	21	14	7	0
∞	Planning Horizon	107.06	–	–	–	–
	Ideal (%)	70	–	–	–	–
	Variables per Run	1,815	–	–	–	–
	Constr. per Run	124	–	–	–	–
21	Planning Horizon	88.32	88.29	–	–	–
	Ideal (%)	71	71	–	–	–
	Variables per Run	784	786	–	–	–
	Constr. per Run	82	82	–	–	–
14	Planning Horizon	79.81	79.77	80.03	–	–
	Ideal (%)	72	72	72	–	–
	Variables per Run	708	712	735	–	–
	Constr. per Run	78	78	79	–	–
7	Planning Horizon	73.61	73.61	73.62	73.78	–
	Ideal (%)	73	73	73	73	–
	Variables per Run	665	671	685	730	–
	Constr. per Run	76	76	76	77	–
0	Planning Horizon	71.24	71.23	70.94	71.03	71.91
	Ideal (%)	71	71	71	71	74
	Variables per Run	709	716	728	767	670
	Constr. per Run	78	78	78	78	73

Table 5.6: Performance details for different maximum numbers of days in advance for urgent and routine patients with the ILP model. Each column (row) represents a different $MNDA$ value for urgent (routine) patients.

Similarly to the experiments with the variation of SCD, the best results for the weighted squared waiting time are found using the same configurations which find the best results for the breach date objective function: creating schedules for urgent patients when their release date is within 21 or more days and for routine patients when it is within 7 days. These values lead to schedules where only few patients have very large waiting times.

Table 5.6 shows details about the performance of the experiments. It is possible to see how the average planning horizon decreases with the decrement of $MNDA$ values. Smaller $MNDA$ values mean that release dates of patients being considered are closer to the current day. Thus, no patients must have their treatment scheduled on a date that is very far, and the necessary planning horizon is shorter. A shorter planning horizon also implies a smaller numbers of variables and constraints on each solver run.

In general, the most interesting schedules in terms of objective function values are achieved when creating schedules for urgent patients when their release date is within 21 days

SCD		$MNDA$		Breach	JMax	JGood	Waiting	Time	TOPSIS
Urg.	Rout.	Urg.	Rout.	(%)	(%)	(%)		(ms)	
5	5	∞	∞	34.38	56.32	93.91	1,308	17	0.15
5	5	∞	0	24.29	**48.43**	92.62	**747**	18	**0.97**
5	5	21	0	24.30	**48.40**	92.61	**747**	18	0.97
3	5	∞	0	24.32	48.47	92.62	**746**	18	0.97
3	5	21	0	24.33	48.44	92.61	**746**	18	0.97
3	5	14	0	24.52	**48.26**	92.55	752	19	0.96
1	5	0	0	29.93	55.59	**92.48**	869	17	0.66
1	1	∞	7	**23.69**	49.92	93.06	766	135	0.95
1	1	21	7	**23.70**	49.89	93.05	766	136	0.95

Table 5.7: Results obtained by efficient combinations of SCD and MNDA values for urgent and routine patients with the ILP model

and for routine patients when their release date is within 7 days.

5.5 Combining the SCD and MNDA Approaches

So far, the best results are found when creating schedules for emergency patients any day ($SCD = 5$) as soon as their booking request arrives ($MNDA = \infty$), and for urgent and routine patients once per week ($SCD = 1$) when their release dates are within 21 and 7 days for urgent and routine patients, respectively ($MNDA = 21$ and 7). However, it is possible that a different combination of SCD and $MNDA$ values will achieve better results. In this section, experiments are run combining all SCD values from Section 5.3 (1, 2, 3 and 5) with all $MNDA$ values from Section 5.4 (∞, 21, 14, 7 and 0).

The total number of combinations sums up to 240. However, not all combinations are worthy examining. To choose which configurations should be examined, the definition of efficient configuration presented in Section 4.6 is used.

Eight efficient configurations are identified and presented in Table 5.7. In addition, results from the configuration with the original values of SCD and $MNDA$ for urgent/routine patients (5/5 and ∞/∞) are also presented for comparison purposes.

By comparing the results from the efficient configuration with the results of the original configuration, it is possible to see that all objective functions can be significantly improved, and all but the JCCO good practice are considerably improved.

There are mainly two types of efficient configurations: with high frequency of scheduling (high SCD values) and with low frequency (low SCD values). In both cases, efficient results are achieved by creating schedules for urgent patients with a large number of days in advance from the release date ($MNDA = \infty$ and 21). However, the best $MNDA$ value for routine patients is different for each situation. When using a high frequency of scheduling, efficient results are found when creating schedules for routine patients when their release date arrives ($MNDA = 0$) and, when creating schedules with a low frequency, efficient results are found when creating schedules for routine patient when their release date is within 7 days. When creating schedules every day, the increase in value of an objective function from a $MNDA$ value of 7 to a value of 0 described in Section 4.5 does not occur, and it is safe to wait for the release date of routine patients to arrive before creating their schedule.

The configurations of SCD–$MNDA$ values 5/5–∞/0 for urgent/routine patients achieve the highest TOPSIS rank, and are used in the next experiments.

5.6 Variable Utilisation Threshold Approach

As previously stated, a possible way to improve the quality of the schedules is to introduce an utilisation threshold to limit the linac time available to patients of certain waiting list status and reserve capacity for future patients. In this section, a variation of the utilisation threshold presented in Section 4.3 is investigated. Previously, the same proportion of capacity was used as threshold for all days. However, when creating schedules for patients, there is no advantage in reserving capacity on the following day (tomorrow) for future patients, since future patients will only be able to start one day after that at the earliest. Therefore, an utilisation threshold which varies with time is proposed. The threshold is set to 100% on the following day and is slowly decremented in the succeeding days.

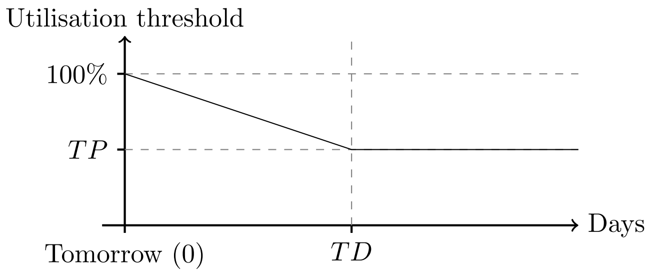


Figure 5.1: Utilisation threshold over days.

The threshold is implemented by introducing two parameters for each patient waiting list status: threshold proportion TP, representing a proportion of the total linac capacity, and threshold days TD, given in number of days. For tomorrow, the full linac capacity is made available for all patients. For the succeeding days, the available capacity linearly decreases for TD days, when it reaches TP, remaining that value for the remaining days. The change of the utilisation threshold over time is given in Figure 5.1, where tomorrow is considered as day 0.

In order to implement this threshold, the model must be modified. The additional notation is adopted:

- s_j: waiting list status of patient j (1 = routine, 2 = urgent, 3 = emergency),
- $\mathcal{N}_t^{vt}$: set of patients of waiting list status t and lower ($\mathcal{N}_t^{vt} = \{j : s_j \leq t, j = 1, \ldots, N\}, t = 1, 2, 3$),
- c_{ikt}: capacity threshold for patients of waiting list status t and lower on linac i and day k, defined as

$$c_{ikt} = \begin{cases} C_{ik} \left(\frac{TP_t - 1}{TD_t} k + 1 \right) & \text{if } k < TD_t, \\ C_{ik}\ TP_t & \text{otherwise} \end{cases}$$

$$t = 1, 2, 3, i = 1, \ldots, M, k = 1, \ldots, T. \quad (5.12)$$

The following constraint is added to enforce the utilisation threshold:

$$\sum_{j}^{\mathcal{N}_t^{vt}} \sum_{l=1}^{S_j} p_{jl}\ x_{ijkl} \leq \max\{c_{ikt} - U_{ik}, 0\} \quad t = 1, 2, 3, i = 1, \ldots, M, k = 1, \ldots, T. \tag{5.13}$$

In the experiments, urgent and routine patients are assigned values 1.00, 0.95 and 0.90 for threshold ratio TP and values 0, 7 and 14 for threshold days TD. For emergency patients, the values used are 1.00 and 0.95 for TP, and 0 and 14 for TD. Similarly to before, some combinations of these values will lead to situations when the available capacity for routine patients is greater than for urgent or emergency patients, which is not desirable taking into account their waiting time targets and relative importance. To avoid these situations, only the combinations of values where the available capacity for emergency (urgent) patients is greater or equal to the available capacity for urgent (routine) patients on all days are considered.

Experiments are run with all sensible combinations of the proposed values of utilisation threshold (TP and TD) as described above, summing up to 45 different configurations. However, not all are worth examining. As in the previous section, only the efficient configurations are shown in Table 5.8. For the sake of comparison, the configuration where no reservations are made is also shown.

The experiments reveal that it is possible to improve all objective functions by using this approach, although the improvement is not very noteworthy for the Breach and Waiting objectives. The improvement in the breach date objective function is very small and is achieved only by the configuration with the smallest restriction on the utilisation. All the other configurations, which impose greater restrictions on the utilisation, achieve a worse value for the breach date objective function than the configuration where no threshold is set.

For the JCCO maximum acceptable target, the improvement is the most considerable, as it drops from 48% in the case with no reservations to 40%. Similarly to the utilisation threshold used with the constructive algorithm, the configurations which found the best values have in common high or no thresholds for emergency and urgent patients and low thresholds for routine patients.

Emergency		Urgent		Routine		Breach	JMax	JGood	Waiting	TOPSIS
TP	TD	TP	TD	TP	TD	(%)	(%)	(%)		
-	-	-	-	-	-	24.29	48.43	92.62	747	0.71
-	-	-	-	0.95	14	**23.96**	44.60	91.28	**718**	0.83
-	-	-	-	0.95	7	24.60	44.59	90.83	730	0.83
-	-	-	-	0.95	0	25.41	45.05	90.74	741	0.79
-	-	-	-	0.9	14	24.55	**40.26**	88.73	785	0.86
-	-	0.95	14	0.95	14	24.86	42.83	89.20	755	**0.86**
-	-	0.95	14	0.95	7	25.62	42.67	88.64	770	0.82
-	-	0.95	14	0.9	7	25.92	40.89	88.07	821	0.75
-	-	0.95	7	0.9	14	24.62	41.01	88.23	793	0.84
-	-	0.95	7	0.9	7	25.97	41.40	87.90	825	0.74
-	-	0.9	14	0.9	14	25.62	41.48	88.18	861	0.69
-	-	0.9	14	0.9	7	27.61	41.91	87.80	919	0.53
-	-	0.9	7	0.9	7	28.61	44.20	**87.76**	992	0.36
0.95	14	0.95	14	0.9	14	24.66	40.54	88.30	791	0.85
0.95	14	0.95	14	0.9	7	25.91	40.91	88.05	821	0.75
0.95	14	0.95	7	0.9	7	25.98	41.45	87.93	825	0.74
0.95	14	0.9	14	0.9	7	27.60	41.92	87.80	918	0.53

Table 5.8: Results obtained by efficient combinations of TP and TD values

The value for JCCO good practice objective function is improved from 93% to around 88% on most of the presented cases. The most restrictive thresholds (values of 0.9/7 for TP/TD) achieve the best values for this target date. However, it was not considerably different from other not so restrictive configurations. In addition, all efficient configurations achieved better values for both JCCO targets than the case with no limitation.

The improvement is also not so considerable for the waiting time squared, as it drops from 747 to 718. Furthermore, the value achieved is worse than the configuration with no limitation in most cases. This objective function behaves similarly to the breach date, where a little improvement can be achieved with a small restriction on the utilisation, but larger restrictions yield worse results.

It should also be noted that the Utilisation Threshold method presented in Section 4.3 is equivalent to the method presented in this section with a TD value of 0. However, only one configuration with a TD value of 0 is considered efficient amongst the ones experimented. Using a threshold which varies with the days seems to achieve significantly better results.

The configuration with values 1/0–1/0–0.9/14 for the parameters TP/TD of emer-

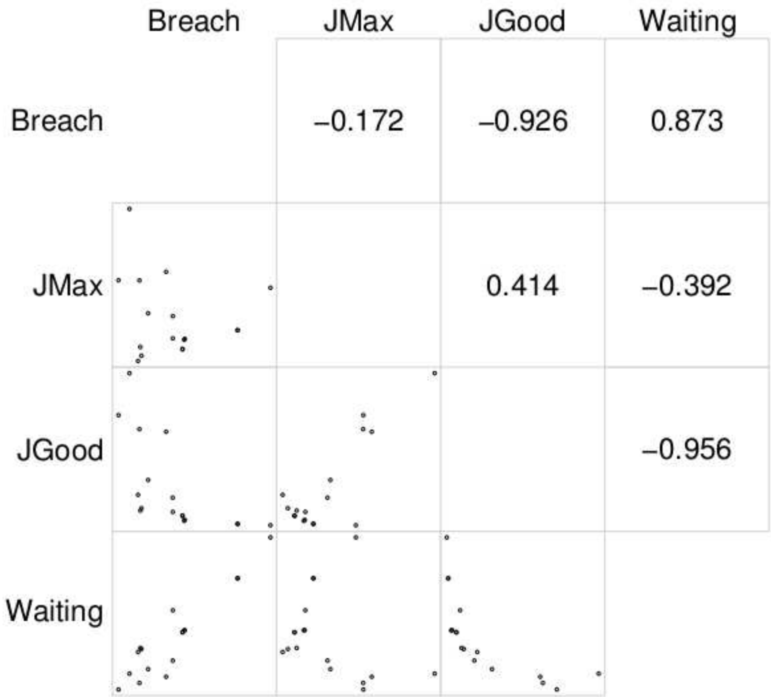


Figure 5.2: Correlation of the values found for each objective function

gency–urgent–routine patients achieved the highest TOPSIS rank, and is adopted in the next experiments.

5.7 Criteria Correlation

In order to better understand the correlation amongst the objectives and help the hospital staff in choosing the best utilisation threshold values, the correlation between the values found in the previous section for each pair of objective functions is calculated and presented in Figure 5.2. The values above the diagonal show Spearman's correlation coefficient ρ, which is close to 1 if the pair of objective functions are positively correlated, to 0 if they are uncorrelated and to -1 if they are negatively correlated. The plots below the diagonal show the scatter plots of each pair of objective functions.

The figure shows a strong negative correlation between the pairs Breach–JGood and JGood–Waiting, meaning that, in the schedules, one of the objective functions can usually be improved by allowing the other to become worse. However, the pair Breach–Waiting is strongly correlated, meaning that configurations that achieve good results for one likely achieve good results for the other. The remaining pairs (Breach–JMax, JMax–JGood and JMax–Waiting) seem to be uncorrelated or weakly correlated.

This indicates that if the decision maker chooses a configuration which yields schedules with a small number of patients who miss the breach date, it is likely that patients will also have small average squared waiting time in this schedule. However, configurations which achieve schedules with low violation of the JCCO good practice target will likely have high violation of the breach date and high squared waiting times.

5.8 Change in the Problem Size and Linacs Allocation

The performance of the algorithm is further analysed when the size of the problem is changed. The size of the problem is determined by the number of linacs and the number of patients to be scheduled. In addition to the current set of linacs currently used by the hospital (1 which emits low energy photon radiation, 1 which emits electron and low energy photon radiations and 2 which emit electron, low and high energy photon radiations), a scenario with 2 additional linacs which emit electron, low and high energy photon radiations is considered. This scenario is chosen due to the hospital's intent of acquiring 2 linacs of this type in the next two years.

In addition, an increase of around 10% in the number of patients per day in the next two years has been estimated. This future increase in the number of patients is also considered in the experiments.

Changes are also made to the linac allocation policy to investigate their effect. The first change consists of removing the linac constraint described in Section 4.1, by considering that patients can be scheduled on any linac that emits their required radiation type. This is easily considered in the model by updating the sets $\mathcal{M}_j$, which contain the linacs that patient j can be assigned to. However, it is no longer possible to split the daily problem into one sub-problems for each radiation type. This change is referred to as "Relaxation 1".

The second change consists of relaxing the linacs constraints to allow patients to be scheduled on a different linac for each session. Constraint (5.5) is replaced by constraint (5.14) to consider all appropriate linacs:

$$\sum_{i=1}^{M} x_{ijk'l'} = \sum_{i=1}^{M} x_{ijkl} \quad k' = k + u_{jkl}, l' = l + 1,$$
$$j = 1, \ldots, N, k = 1, \ldots, T - u_{jkl}, l = 1, \ldots, S_j - 1 \quad (5.14)$$

As in constraint (5.5), sessions l and $l+1$ of patient j are scheduled u_{jkl} days apart. This change is referred to as "Relaxation 2".

The results obtained by varying these parameters are shown in Table 5.9, where each column (row) specifies if relaxation 1 (2) is used. The values in bold are the ones for which there is no other value for the same objective function in the same sets of linacs and patients which is significantly better.

When using relaxation 2 (allowing patients to have each session on a different linac), the space of feasible schedules is increased and, in theory, it should be possible to find better solutions. However, by comparing each configuration where this relaxation is used to its counterpart, it is possible to see that there is no significant improvement gained in most cases, and the objective function values found are even significantly worse on some cases. This relaxation is also accompanied by a large increase in the required CPU time, even if this increase is still well within acceptable limits.

In addition, there are other points to be taken into consideration. Currently, the first session of each patient is slightly longer than the other sessions in order to allow time for the treatment plan to be loaded into the machine and for the patient to become familiar with the machine, the positioning, any cast that might be necessary for the treatment, as well as becoming familiar with the staff itself. If different sessions are scheduled on different linacs, the required time might increase since the treatment plan will need to be loaded into different machines, and the patient will need to become familiar with different machines. For this reason, this relaxation is frowned upon by the hospital staff.

		No	Yes
No	Breach (%)	24.86	**11.19**
	JMax (%)	42.83	**33.71**
	JGood (%)	89.20	**87.75**
	Waiting	755	**552**
	Time (ms)	3	5
Yes	Breach (%)	24.84	**11.19**
	JMax (%)	42.82	**33.71**
	JGood (%)	89.25	**87.77**
	Waiting	755	**552**
	Time (ms)	15	109

(a) Current sets of linacs and patients

		No	Yes
No	Breach (%)	30.41	**11.75**
	JMax (%)	45.30	**33.64**
	JGood (%)	89.30	**88.00**
	Waiting	1,148	**557**
	Time (ms)	5	14
Yes	Breach (%)	30.32	11.81
	JMax (%)	45.27	33.70
	JGood (%)	89.25	88.08
	Waiting	1,148	558
	Time (ms)	33	127

(b) Current set of linac and future set of patients

		No	Yes
No	Breach (%)	24.78	**11.10**
	JMax (%)	42.82	**33.69**
	JGood (%)	89.15	**87.70**
	Waiting	754	**550**
	Time (ms)	2	0
Yes	Breach (%)	24.77	**11.10**
	JMax (%)	42.81	**33.69**
	JGood (%)	89.20	**87.70**
	Waiting	753	**550**
	Time (ms)	4	0

(c) Future set of linac and current set of patients

		No	Yes
No	Breach (%)	29.78	**11.09**
	JMax (%)	45.21	**33.53**
	JGood (%)	88.90	**87.60**
	Waiting	1,139	**548**
	Time (ms)	3	0
Yes	Breach (%)	29.75	**11.09**
	JMax (%)	45.15	**33.53**
	JGood (%)	88.84	**87.60**
	Waiting	1,139	**548**
	Time (ms)	6	0

(d) Future sets of linacs and patients

Table 5.9: Results obtained varying the size of the problem and relaxation of constraints, where each column (row) specifies if relaxation 1 (2) is used

By using relaxation 1 (considering all radiation types from each linac), the algorithm is able to achieve the best results. The utilisation levels achieved for all machines are roughly the same, and no specific machine is overloaded. As previously stated, some patients may have their release date after their due dates, making it impossible to meet these targets. By implementing this change in the policy, the algorithm was able to find objective function values very close to the minimum values of the objectives functions. When relaxations 1 and 2 are used, the schedule found often overloads the linacs which emit all three types of radiation, leaving no room for future patients who require high energy photon radiation. Using only relaxation 2 results in schedules where patients are more evenly distributed amongst the machines.

Increasing the number of patients causes an increase in the values of the objective functions in most cases, as expected. This is specially true when not using relaxation 1 (considering only one type of radiation from each linac). In these cases, the low energy linacs are the busiest, and a large number of the patients who do not meet the due dates and have large waiting times are patients who require low energy radiation. When using relaxation 1, the increase in patients does not affect the values of the objective functions too much. Although the absolute value of the objective functions increases, the corresponding values per patient remain roughly the same. The extra patients who arrive during high linac utilisation periods of the year are "responsible" for a large part of the increase of the objective functions. However, the extra patients who arrive in periods of low utilisation are able to receive good schedules and help decrease the average values of the objective functions.

The increase in the number of linacs improves the objective function values in most cases, but not by a considerable amount. When not using relaxation 1, the linacs which emit only low energy photon radiation are the busiest, and the patients using them have the largest waiting times. Adding two more linacs which emit all radiation types benefits only patients who require high energy radiation, and therefore, does not greatly improve the schedule. When using relaxation 1, the values of the objective functions achieved when the the original number of linacs is used are already very close to the minimum values, as suggested above, and there is not much room for improvement.

In addition to the objective function values presented, Table 5.10 shows the average number of patients considered per run, the percentage of runs when the ideal solution was found by the constructive approach and identified, the average CPU time required per day, the average number of solver runs per day, and the average numbers of variables and constraints per run.

Using relaxation 2 (allowing patients to have each session on a different linac) does not make the daily problem more complex for the constructive approach. It can still find the ideal solution for roughly the same number of runs as before. It does, however, cause an increase in the number of variables and constraints. This causes an increase in the required CPU time, which is still well within acceptable limits.

		No	Yes
No	Patients per Run	4.21	8.81
	Ideal (%)	82	96
	Time per day (ms)	6	5
	Runs per day	2.06	0.98
	Variables per Run	159	932
	Constr. per Run	136	490
Yes	Patients per Run	4.21	8.81
	Ideal (%)	82	96
	Time per day (ms)	30	107
	Runs per day	2.06	0.98
	Variables per Run	1,716	12,725
	Constr. per Run	300	959

(a) Current sets of linacs and patients

		No	Yes
No	Patients per Run	4.54	9.69
	Ideal (%)	74	87
	Time per day (ms)	11	14
	Runs per day	2.10	0.98
	Variables per Run	230	901
	Constr. per Run	174	481
Yes	Patients per Run	4.54	9.69
	Ideal (%)	75	86
	Time per day (ms)	70	125
	Runs per day	2.10	0.98
	Variables per Run	2,255	10,949
	Constr. per Run	354	902

(b) Current set of linac and future set of patients

		No	Yes
No	Patients per Run	4.21	8.81
	Ideal (%)	84	100
	Time per day (ms)	4	0
	Runs per day	2.06	0.98
	Variables per Run	123	0
	Constr. per Run	118	0
Yes	Patients per Run	4.21	8.81
	Ideal (%)	84	100
	Time per day (ms)	9	0
	Runs per day	2.06	0.98
	Variables per Run	1,258	0
	Constr. per Run	268	0

(c) Future set of linac and current set of patients

		No	Yes
No	Patients per Run	4.54	9.69
	Ideal (%)	80	100
	Time per day (ms)	5	0
	Runs per day	2.10	0.98
	Variables per Run	148	0
	Constr. per Run	136	0
Yes	Patients per Run	4.54	9.69
	Ideal (%)	80	100
	Time per day (ms)	12	0
	Runs per day	2.10	0.98
	Variables per Run	1,297	0
	Constr. per Run	291	0

(d) Future sets of linacs and patients

Table 5.10: Performance details varying the size of the problem and relaxation of constraints

By using relaxation 1, the daily problem becomes easier for the constructive heuristic, which is able to find the ideal solution for a higher number of days. When the number of linacs is also increased, the constructive approach is able to find the ideal solutions in 100% of the days, due to the large available linac capacity. The change in the average required CPU time for this modification depends on the other parameter values. Although using relaxation 1 causes an increase in the number of variables and constraints at each solver run, and therefore an increase in the required CPU time of each individual run, the total number of solver runs necessary per day is decreased, since there is at most one solver run per day instead of three (one for each type of radiation).

As expected, the future increase in the number of patients makes the problem slightly more complex for the constructive approach, which can find the ideal solution for a smaller number of days. The number of variables and constraints per run are also slightly increased, which causes an increase in the average required CPU time per day. However, this time is still well within acceptable limits.

The increase in the number of linacs makes the problem easier to solve. With more space available on the linacs, it is more likely that the constructive approach is able to find an ideal schedule. As highlighted before, with the increase in the number of linacs and considering all radiation types available, the constructive approach is always able to find the ideal solution in our experiments.

When the increase in the number of linacs is considered, but only one radiation type from each linac, the average number of variables and constraints necessary per run decreases. This result seems counter intuitive at first, but it can be easily explained. With the original linac setup, the sub-problems which consider only high energy photon patients have a higher number of variables and constraints than the other sub-problems, since they are the only ones with more than one linac. A larger number of linacs which emit high energy photon radiation does cause an increase in the number of variables and constraints for the sub-problems which consider linacs of this type. However, the number of times when the constructive approach finds an ideal solution also increases. Since the model is not generated when the constructive approach finds an ideal solution, the increase in the number of linacs results in a lower average number of variables and constraints.

An important observation must be made. Although the constructive algorithm is able to find an ideal solution for 100% of the runs when considering the future number of linacs and using relaxation 1, such solutions might not be implemented in reality. The hospital oncologists currently prescribe treatments with small number of sessions and sessions of small duration due to the limited capacity of linacs. If a larger number of linacs is made available, longer treatments and sessions might be prescribed. In addition, the new linacs might not be used exclusively for radiotherapy treatment, and might be also used for some of the pre-treatment phases. This is yet to be decided by the centre's administrative staff, and could decrease the linac capacity made available for treatment.

5.9 Summary

This chapter describes the ILP model developed for the radiotherapy problem investigated in this thesis. The model is consisted mainly of four types of constraints and one set of binary decision variables. The four objective functions considered previously (minimisation of the number of patients who do not meet the breach date, the JCCO maximum acceptable and good practice waiting time targets, and the minimisation of the average squared waiting time) are also modelled in the ILP functions. In order to consider multiple objectives, a lexicographic approach is used.

The model developed is dependent on the duration of the planning horizon T, which must be supplied as input by the decision maker. Experiments are conducted by taking the duration of the planning horizon from a schedule created by the constructive algorithm and increasing it by a fixed amount of days, called *slack*. Several values of slack are experimented with to investigate how they affect the performance of the algorithm. The results indicate that a slack value of 14 is appropriate for the purposes of this work.

Section 5.3 investigates the quality of the schedule generated by using the ILP model with the SCD approach introduced in Section 4.4. Experiments reveal that it is possible to achieve a small improvement for each of the evaluated objective functions by decreasing the frequency with which schedules are created. When decreasing this frequency, unscheduled patients accumulate, increasing the search space for each run and enabling the algorithm to find new more interesting solutions. Furthermore, while demanding a smaller number of solver runs, the decrease in frequency of scheduling yields a small increase in the total required CPU time.

The $MNDA$ approach presented in Section 4.5 is also combined with the ILP model to investigate if the objective functions can be improved by waiting for the release date of patients to become closer to the current date before creating their schedule. Experiments reveal that considerable improvements can be found when using this approach. The best results are achieved when waiting for the release date of urgent and routine patients to be within 21 and 7 days respectively, as these margins result in more space available for emergency and urgent patients on days closer to the current date.

In order to analyse if the schedule can be improved by using a different combination of SCD and $MNDA$ values, a full factorial experimental design of the suggested values for these parameters is presented in Section 5.5, and the efficient configurations are presented. The experiments suggest that different combinations of $SCD/MNDA$ values will favour different objective functions at different costs of required CPU time per run. Creating schedules for urgent and routine patients once per week and when the release date of routine patients is within 7 days results in the lowest number of patients missing the breach date. It also has a higher CPU cost. However, this cost is well within acceptable limits when divided per day. However, creating schedules with a high frequency for all patients, but only on the release date for routine patients, achieves the best values for the compliance with the JCCO maximum acceptable and the weighted squared waiting time.

Furthermore, the Utilisation Threshold approach presented in Section 4.3 is extended to a threshold which varies in time and it is presented in Section 5.6. Since it is highly unlikely that a patient who arrives on the current date is able to start treatment in the next few days, the threshold is set to large values for days close to the current date and slowly decreases for the succeeding days. The experiments suggest that a variable threshold as presented is a better alternative then a fixed threshold. In addition, when high thresholds values are used, more patients are able to meet the least strict target dates (breach and JCCO maximum acceptable). In contrast, when low threshold values are used, more patients are able to meet the most strict target dates (JCCO good practice).

An analysis of the criteria correlation is also presented. It concludes that the breach and squared waiting time objective functions are highly correlated. It is speculated that this is due to both objectives highly penalising very long delays, while giving no or relatively little penalty to small delays. They are also considered to be negatively correlated to the JCCO good practice objective function, which penalises small and large delays equally.

The algorithm scales well for the foreseen increase in patients and linacs according to the experiments. Allowing patients to be scheduled on any linac that emits their required type of radiation can greatly improve the quality of schedules. However, allowing patients to be scheduled on a different linac for each session does not achieve significant improvements in any of the investigated scenarios. On the contrary, it may achieve significantly worse results due to overloading linacs which can emit all radiations, besides being frowned upon by the hospital staff.

So far, all approaches presented involve imposing constraints on list of patients available for scheduling either according to the day of the week or to the patient's release dates, or constraints on machine utilisation. Subject to these constraints, a schedule is built trying to optimise the objectives considering the patients currently being scheduled. However, no attempt is made at finding more than one solution to be suggested to the decision maker, or at predicting which patients will arrive in the near future. These are investigated in the next chapter.

"As far as the laws of mathematics refer to reality, they are not certain; and as far as they are certain, they do not refer to reality."
Albert Einstein (1879–1955)

Chapter 6

Robust Approaches

In a real-world application dealing with multiple objectives, it is important to provide more than one solution to the decision maker. There is often no single optimal solution, but a set of optimal solutions where the only way of improving one objective, if such improvement is possible, is at the cost of another objective. Such a set is commonly referred to as the *Pareto frontier* (Deb, 2005; Pareto, 1906). In order to present the possible trade-offs to the decision maker, the Pareto frontier (or an approximation of it) should be supplied.

This chapter presents a Pareto-type multi-objective algorithm as an alternative to the lexicographical ordering used in the previous chapter. The algorithm is an adaptation of the algorithm previously suggested by Sylva & Crema (2004) for minimisation problems and is able to find a set of solutions which represent the whole Pareto frontier for integer linear programming problems. This algorithm is used in combination with the approaches presented in this chapter.

As suggested in Chapter 1, the greatest challenge in the problem tackled in this thesis is building a schedule of high quality not only for patients who already arrived, but which is also robust in terms of patients who will arrive in the future. Two robust approaches to radiotherapy scheduling are also investigated in this chapter. The first approach is referred to as *pre-scheduling*. The idea is to consider patients whose scheduling date has not yet arrived when creating a schedule, such that their schedule is created by the algorithm, but not written in the booking system. Patients whose release date is later than a specific number of days away remain unscheduled. This enables the algorithm to reserve the exact amount of space needed for these patients, and to re-arrange the reserved space as needed on the next days.

The second robust approach is called *rolling horizon*. It is consisted of trying to predict which patients will arrive in the near future and creating the schedule considering these predicted future patients.

To achieve a more robust schedule, multiple future scenarios are generated and the schedule which achieves the best objective functions considering all scenarios is adopted.

Finally, a rescheduling approach is presented. Although this is contradictory with the concept of "robust" (no patients would need to be rescheduled in a perfectly robust schedule), the author believes that it is important to investigate what improvement is achievable through rescheduling and to compare it with the results from the other approaches.

6.1 Multi-objective Integer Linear Programming

One of the goals in this work is to develop an algorithm capable of giving a set of possible schedules for the decision maker to choose from. Ideally, these suggested schedules should represent the Pareto frontier. In order to find such a set, an algorithm based on the work of Sylva & Crema (2004) is used. The algorithm is able to find a set of solutions which represent the whole Pareto frontier for problems where all objective functions yield integer values. It is a variation of the work of Klein & Hannan (1982), and consists of a number of steps where increasingly more constrained integer linear programming models are used to generate a new solution at each step. The algorithm can be outlined as:

Step 1. Find a solution which minimises a weighted sum of the objectives.

Step 2. Store this solution in the set of efficient solutions found.

Step 3. Remove from the feasible search space the solution found and all solutions dominated by it.

Step 4. Repeat steps 1 to 3 until there are no more solutions in the feasible search space.

Step 5. Return the set of efficient solutions found.

The outline of the algorithm is quite intuitive and simple. However, its implementation for Integer Linear Programming is not trivial, particularly the implementation of step 3. This implementation is discussed below.

The Sylva–Crema algorithm has been originally defined for maximisation problems. In this work, an adaptation is proposed for minimisation problems. The implementation of the algorithm is given as follows. A general Multi-objective Integer Linear Programming (MOILP) problem $\mathbf{P}$ can be stated as:

$$\mathbf{P} : \{\text{minimise } \mathbf{f}(\mathbf{x}) : \mathbf{g}(\mathbf{x}) \geq 0\}, \tag{6.1}$$

where $\mathbf{f}(\mathbf{x})$ is a vector of Y linear objective functions, $\mathbf{g}(\mathbf{x}) \geq 0$ is a vector of linear constraints and $\mathbf{x}$ is a vector of integer variables which represent a solution.

A feasible solution $\mathbf{x}^\star$ to problem $\mathbf{P}$ is said to be an *efficient solution* iff $\nexists \mathbf{x} : f_m(\mathbf{x}) \leq f_m(\mathbf{x}^\star)\ \forall m, \exists m' : f_{m'}(\mathbf{x}) < f_{m'}(\mathbf{x}^\star)$, i.e. there is no solution $\mathbf{x}$ such that $f_m(\mathbf{x})$ is at least as good as $f_m(\mathbf{x}^\star)$ for all objectives m and $f_{m'}(\mathbf{x})$ is strictly better than $f_{m'}(\mathbf{x}^\star)$ for at least one objective m'.

As it is widely known, if $\mathbf{x}^\star$ is an optimal solution to the single objective problem

$$\mathbf{P}^0 : \{\text{minimise } \boldsymbol{\lambda}^\intercal\ \mathbf{f}(\mathbf{x}) : \mathbf{g}(\mathbf{x}) \geq 0\} \tag{6.2}$$

for a given vector of weights $\boldsymbol{\lambda} > 0$, $\mathbf{x}^\star$ is also an efficient solution to the original problem $\mathbf{P}$ (Steuer, 1986). Efficient solutions for problem $\mathbf{P}$ which are also optimal for problem $\mathbf{P}^0$ are called *supported solutions*. Some efficient solutions (known as *non-supported*) will not be optimal for

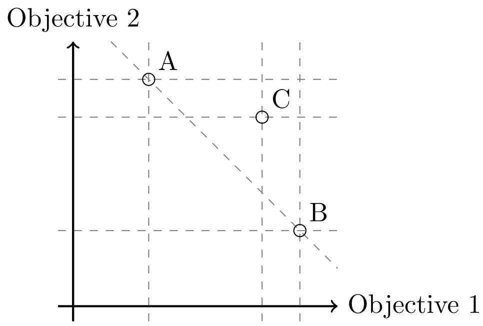


Figure 6.1: Example of a non-supported solution, where objectives 1 and 2 are to be minimised, solutions A and B are supported solutions and solution C is non-supported

problem $\mathbf{P}^0$ for any vector of weights $\boldsymbol{\lambda} > 0$ (Bitran, 1977). However, non-supported efficient solutions can be found if known efficient solutions and the solutions dominated by these are removed from the search space.

An example is shown in Figure 6.1, where A, B and C are the only efficient solutions of the search space. There is no $\boldsymbol{\lambda} > 0$ which will enable us to find solution C. For any $\boldsymbol{\lambda}$, either solution A or B will be found. However, if either solution A or B is removed from the feasible search space, there will be at least one vector $\boldsymbol{\lambda}$ which will enable us to find solution C.

The Sylva–Crema algorithm starts by finding an optimal solution to problem $\mathbf{P}^0$. If there is no such solution, problem $\mathbf{P}$ cannot be satisfied and the algorithm terminates. Otherwise, solution $\mathbf{x}^0$ is found. At each succeeding step $n+1$, a new problem $\mathbf{P}^{n+1}$ is created by adding conditions to problem $\mathbf{P}^n$ which remove the point $\mathbf{f}(\mathbf{x}^n)$ from the objective search space as well as points dominated by it. This is implemented by adding the following constraints and variables:

$$f_m(\mathbf{x}) + \delta_m \leq f_m(\mathbf{x}^n) + G\ (1 - y_{mn}) \quad m = 1, \ldots, Y, \tag{6.3}$$

$$\sum_{m=1}^{Y} y_{mn} \geq 1, \tag{6.4}$$

$$y_{mn} \in \{0, 1\} \quad m = 1, \ldots, Y.$$

where G is a very large integer, δ_m is a precision factor chosen by the decision maker which determines the minimum difference between $f_m(\mathbf{x})$ and $f_m(\mathbf{x}^n)$ so that $f_m(\mathbf{x})$ is considered a noteworthy improvement over $f_m(\mathbf{x}^n)$, and y_{mn} is a binary variable which takes the value 1 if $f_m(\mathbf{x})$ is less than $f_m(\mathbf{x}^n)$ and their difference is of at least δ_m, 0 otherwise. Since the values of all objective functions are integer in the problem presented in this work, δ_m is set to 1 for all objectives in order to find a set of solutions representing the whole Pareto frontier.

The value of G chosen in this work (and deducible from Sylva & Crema (2004)) is

$$G = f_m^{max} - f_m(\mathbf{x}^n) + \delta_m, \tag{6.5}$$

where f_m^{max} is the maximum possible value for objective function m. For the model presented in Section 5.1, finding the value of f_m^{max} is trivial. Since all objectives considered are non-decreasing functions of the start date, f_m^{max} is given by considering each patient as being scheduled on the latest feasible day taking into account the value of the chosen planning horizon T and the current linac utilisation. This is the same calculation method used in Section 5.2.

After creating problem $\mathbf{P}^{n+1}$, the new problem is solved. If no feasible solution is found, the algorithm terminates. Otherwise, solution $\mathbf{x}^{n+1}$ is found and the algorithm proceeds to step $n+2$. For examples and a formal proof of why this algorithm can find a set of solutions which represent the whole Pareto frontier, including non-supported solutions, please refer to Sylva & Crema (2004).

There are several other multi-objective algorithms (MOAs) which could be used in this work. From classical methods, such as goal programming and ϵ-constraint methods (Miettinen, 1999), to more modern ones, such as the Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) (Deb et al., 2002; Deb, 2005), a Genetic Algorithm specially tailored at finding solutions which balance a number of objectives. However, these other approaches either require a much larger number of steps where a ILP problem is solved at each step (e.g. ϵ-constraint methods), or give no guarantee that all efficient solutions will be found in a finite number of steps (e.g. NSGA-II and the majority of evolution-based MOAs). The Sylva–Crema algorithm is chosen for not having those limitations. Furthermore, the problem tackled in this thesis has a small number of efficient solutions per day, making this algorithm ideal for it, as will be seen in the next sections.

6.2 Pre-scheduling Approach

The first robust approach investigated in this work consists of patient pre-scheduling. It extends the $MNDA$ approach, which is investigated in Section 4.5. In that approach, the maximum number of days in advance with which a patient could be scheduled is controlled by a numeric parameter supplied by the decision maker referred to as ($MNDA$). The date on which the schedule for a patient j is created is referred to as sch_j, as defined in Equation 4.7.

In the pre-scheduling approach presented in this section, a patient j, whose scheduling date sch_j is within a specific number of days, is considered when creating the schedule but the schedule is not implemented. In other words, patient j is included in the problem instance for that day and the solver finds a schedule for patient j, but this schedule is not written in the booking system. Patient j remains unscheduled until date sch_j actually arrives. This can be seen as equivalent to implementing the schedule, but not informing the patient, so the schedule could be modified without cost. This way, it is possible to build schedules which take patients who already arrived into consideration, but without the issue of creating schedules for them too soon.

Patients are considered in the schedule when their scheduling date sch_j is within $Hpre$ days, where $Hpre$ is referred to as the *pre-scheduling horizon*. The date from which patient j is considered when creating the daily schedule can be defined as:

$$pre_j = \max\{b_j, sch_j - Hpre\}. \tag{6.6}$$

A time-line of the new scheduling dates for a given patient j is shown in Figure 6.2.

The hospital staff has a preference for large values of $MNDA$, as they indicate that patients are informed of their schedule sooner. This increases the quality of service, since patients are able to plan far ahead the days on which they would need to come to the hospital to receive their fractions. With that in mind, experiments are conducted in order to investigate which levels of improvement are achievable when using different lengths of the pre-scheduling horizon combined with each level of $MNDA$.

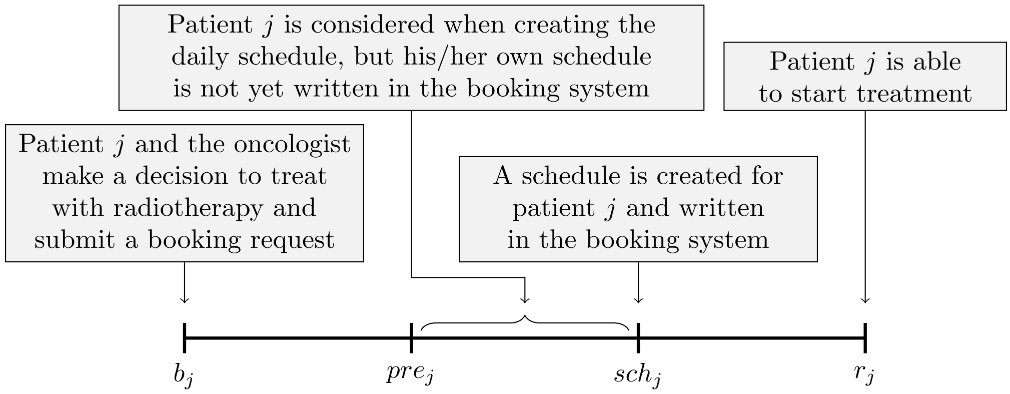


Figure 6.2: Time-line for the scheduling dates of patient j

The Sylva–Crema algorithm is used on each day to find a set of Pareto schedules. In the real world, a decision maker could choose any of the schedules generated to be used. In order to try to mimic that behaviour, a random schedule is chosen from the set of Pareto schedules found to be implemented.

In order to evaluate the effect of different $MNDA$ values in combination with different lengths for the pre-scheduling horizon, experiments are conducted with $MNDA$ values of 0, 7, 14 and 21, and pre-scheduling horizon $Hpre$ values of 0, 7, 14 and ∞, both only for routine patients. Results are shown in Table 6.1, where a value of 0 for $Hpre$ indicates that no pre-scheduling is made and a value of ∞ indicates that all patients are pre-scheduled as soon as their booking request is received. For urgent and emergency patients, the values used for $MNDA$ and $Hpre$ are ∞ and 0, respectively. As before, MWW is used with bootstrapping to make a statistical analysis of the results, and the values considered the best for each objective function are presented in bold.

One can notice that all objective functions are improved when the pre-scheduling horizon increases, particularly Breach, JMax and Waiting. However, this improvement is also accompanied by a considerable increase in the CPU time.

The maximum number of days in advance with which patients are scheduled ($MNDA$ value) seems to have a large influence on the results. Large values of $MNDA$ cause the scheduling date sch_j to be close to the decision to treat b_j. Since it is only possible to pre-schedule a patient

		∞	21	14	7	0
0	Breach (%)	34.40	31.44	28.88	26.11	24.29
	JMax (%)	56.32	55.20	53.95	51.62	48.39
	JGood (%)	93.89	93.73	93.57	93.27	92.64
	Waiting	1,309	1,110	969	827	746
	Time (ms)	5	7	24	24	20
7	Breach (%)	–	30.15	26.31	23.64	21.88
	JMax (%)	–	54.96	52.82	48.56	45.70
	JGood (%)	–	93.74	93.55	93.04	92.08
	Waiting	–	1,069	918	790	723
	Time (ms)	–	498	14,213	18,727	18,368
14	Breach (%)	–	29.75	24.95	21.53	20.00
	JMax (%)	–	54.92	52.44	47.00	43.01
	JGood (%)	–	93.75	93.56	92.99	91.52
	Waiting	–	1,047	881	754	703
	Time (ms)	–	1,069	30,101	53,590	54,168
∞	Breach (%)	–	29.25	24.06	20.30	**17.83**
	JMax (%)	–	54.89	52.36	46.25	**41.20**
	JGood (%)	–	93.74	93.56	93.01	**91.21**
	Waiting	–	996	851	730	**674**
	Time (ms)	–	1,403	38,034	67,883	75,070

Table 6.1: Results obtained for different maximum numbers of days in advance of routine patients and sizes of the pre-scheduling horizon, where each column (row) represents a different $MNDA$ ($Hpre$) value for routine patients

when the schedule is being created between these two dates, fewer patients can be pre-scheduled when large values of $MNDA$ are used. When small values of $MNDA$ are used, more patients are pre-scheduled and a larger improvement of the objective functions can be found.

In order to better understand the performance of the algorithm, Table 6.2 presents the average increase in the number of patients considered per day, the percentage of runs when the ideal solution is found and identified by the constructive algorithm, the average number of efficient solutions obtained per day, the percentage of times when the Sylva–Crema algorithm is used and exceeds the time limit, and the average number of variables and constraints per run.

The number of patients considered daily becomes larger with the increase of the pre-scheduling horizon. This is the expected behaviour, since the time interval $[pre_j, sch_j]$ is longer in these cases. The decrement of the $MNDA$ values of patients for the same value of the pre-scheduling horizon also causes an increase in the number of patients considered. Since not many patients have their decision to treat date b_j many days before their release date r_j, not

		∞	21	14	7	0
0	Patient Incr. (%)	0	0	0	0	0
	Ideal (%)	80	80	81	82	82
	Efficient Solutions	1.01	1.02	1.03	1.03	1.04
	Exceeded Time (%)	0.00	0.00	0.03	0.04	0.00
	Variables per Run	182	124	130	134	138
	Constr. per Run	57	50	50	52	54
7	Patient Incr. (%)	–	145	208	246	244
	Ideal (%)	–	63	61	61	59
	Efficient Solutions	–	1.31	2.38	2.64	2.66
	Exceeded Time (%)	–	0.16	4.25	5.12	4.97
	Variables per Run	–	232	383	416	439
	Constr. per Run	–	66	89	95	97
14	Patient Incr. (%)	–	237	353	455	491
	Ideal (%)	–	62	60	59	57
	Efficient Solutions	–	1.37	2.94	3.08	3.07
	Exceeded Time (%)	–	0.27	7.92	16.25	16.75
	Variables per Run	–	276	542	725	780
	Constr. per Run	–	70	102	109	112
∞	Patient Incr. (%)	–	588	799	1,053	1,306
	Ideal (%)	–	62	60	59	56
	Efficient Solutions	–	1.39	3.00	2.80	2.63
	Exceeded Time (%)	–	0.42	10.48	26.46	34.93
	Variables per Run	–	442	768	1,399	2,559
	Constr. per Run	–	78	111	126	153

Table 6.2: Performance details for different maximum numbers of days in advance of routine patients and sizes of the pre-scheduling horizon, where each column (row) represents a different $MNDA$ ($Hpre$) value for routine patients

many patients will be available for pre-scheduling if large values of $MNDA$ are used. As the number of patients considered grows, the problem becomes larger and harder to solve. The number of variables and constraints increases, and probability of the constructive algorithm finding an ideal solution drops.

As expected, the number of efficient solutions found also increases with the number of patients considered. However, when the increase in the number of patients considered is considerably large, e.g. above 10 times larger, the number of runs that exceed the maximum CPU time is also large. Each time the algorithm is terminated before finishing, it is possible that additional efficient solutions exist, but were not found owing to the early termination of the algorithm. This may lead to a smaller total number of efficient solutions found in the end of the experiment.

The configuration where routine patients are considered as soon as their booking request is received, but their schedule is only written to the booking system on their release date ($MNDA = 0, Hpre = \infty$), achieved the best results and is used in the next sections.

6.3 Rolling Horizon Approach

The next approach to be investigated is referred to as *rolling horizon*. It consists of trying to predict which patients will arrive in the next number of days, and consider these predicted patients when creating the schedule for the real patients.

This approach is also used by Sadki et al. (2010b), cited in Section 2.5.2. In order to extend their approach, multiple scenarios of future patients are generated instead of just one. A number of possible future scenarios are generated and included in the model instance. The schedules which achieve the best performance considering all future scenarios are considered the most robust schedules found.

To implement this approach, the model presented in Section 5.1 is modified. The following additional input is considered:

- H: number of days in the rolling horizon,
- E: number of predicted scenarios,
- e: index for predicted scenarios ($e = 1, \ldots, E$),
- $\mathcal{N}^{rhz}$: set of real patients,
- $\mathcal{N}_e^{rhz}$: set of patients from predicted scenario e.

To ensure that the daily linac capacity is correctly respected separately in each predicted scenario, constraint (5.7) is substituted by constraint (6.7):

$$\sum_{j}^{\mathcal{N}^{rhz}} \sum_{l=1}^{S_j} p_{jl}\ x_{ijkl} + \sum_{j}^{\mathcal{N}_e^{rhz}} \sum_{l=1}^{S_j} p_{jl}\ x_{ijkl} \leq C_{ik} - U_{ik}$$

$$i = 1, \ldots, M, k = 1, \ldots, T, e = 1, \ldots, E. \quad (6.7)$$

This ensures that the sessions scheduled for linac i on day k of real patients and patients of predicted scenario e do not exceed the available machine capacity for linac i on day k.

An algorithm is developed to create the scenarios of predicted patients based on historical data. For that effect, we choose an algorithm similar to the one used to generate the experimental data sets and presented in Section 4.1. The algorithm generates patients for E scenarios, each H days long, by randomly re-sampling from the combined data set. It can be outlined as:

Step 1 For each scenario $e = 1, \ldots, E$, repeat steps 2 to 9.

Step 2 For each date $k = k^0 + 1, \ldots, k^0 + H$, where k^0 is the current day, repeat steps 3 to 9.

Step 3 If date k is a weekend or a bank holiday, set the number of patients arriving on date k to 0 and move to the next date. Otherwise, go to step 4.

Step 4 Select a random year from the combined data set.

Step 5 Choose a date k' from the year selected in the previous step, such that k and k' are the same day of the week, k' is not a bank holiday and is as close as possible to the day of the year of date k.

Step 6 For each waiting list status $t \in \{$emergency, urgent, routine$\}$, repeat steps 7 to 9.

Step 7 Consider N as the number of patients of waiting list status t arriving on day k' on the combined data set.

Step 8 Select N random patients of waiting list status t from the combined data set and copy them to the experimental data set.

Step 9 Change the date of the decision to treat of copied patients to k and adjust the remaining dates accordingly.

It should be noted that, although this is in practice the same algorithm used to generate the experimental data sets, the random seeds used for each are different and it is very unlikely that this algorithm will predict the exact same patients as the ones in the experimental data set.

Furthermore, this method can be easily applied to the data of other hospitals in order to build a tailored patient generator.

By increasing the number of scenarios, the probability that at least one of them will have many emergency and/or urgent patients increases. This specific scenario can cause the schedule of real patients to be overly conservative, having an excess of unused reserved time, thus decreasing its quality. In order to prevent these situations, a greater weight is given to real patients in the objective functions. To implement this new set of weights, a weight ω_j is introduced for each patient j and defined as

$$\omega_j = \begin{cases} E & \text{if patient } j \text{ is a real patient, i.e. } j \in \mathcal{N}^{rhz}, \\ 1 & \text{otherwise.} \end{cases} \tag{6.8}$$

These weights are included in the objective functions $\mathbf{f}(\mathbf{x})$, which are previously defined in equations (5.8–5.11). They are redefined as:

$$f_1(\mathbf{x}) = \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=d_j^1+1}^{T} \omega_j \; x_{ijk1}, \tag{6.9}$$

$$f_2(\mathbf{x}) = \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=d_j^2+1}^{T} \omega_j \; w_j \; x_{ijk1}, \tag{6.10}$$

$$f_3(\mathbf{x}) = \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=d_j^3+1}^{T} \omega_j \; w_j \; x_{ijk1}, \tag{6.11}$$

$$f_4(\mathbf{x}) = \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=b_j+1}^{T} \omega_j \; (k-b_j)^2 \; w_j \; x_{ijk1}. \tag{6.12}$$

By using only integer values for ω_j, the values of all objective functions $\mathbf{f}(\mathbf{x})$ remain integer. Therefore, it is possible to continue using δ_m with a value of 1 for all objectives m in the Sylva–Crema algorithm in order to find a set of solutions representing the whole Pareto frontier.

The rolling horizon is combined with pre-scheduling to investigate if it is possible to further improve pre-scheduling. The values experimented with for the number of days in the

		0	1	2	4	7
0	Breach (%)	**17.83**	–	–	–	–
	JMax (%)	41.20	–	–	–	–
	JGood (%)	91.21	–	–	–	–
	Waiting	**674**	–	–	–	–
	Time (ms)	75,070	–	–	–	–
	TOPSIS	0.15	–	–	–	–
1	Breach (%)	–	**17.89**	**17.91**	17.97	18.01
	JMax (%)	–	40.76	40.50	40.20	39.69
	JGood (%)	–	91.00	90.76	90.56	90.34
	Waiting	–	**674**	**673**	**673**	**673**
	Time (ms)	–	74,807	76,310	76,621	78,150
	TOPSIS	–	0.23	0.35	0.47	0.68
2	Breach (%)	–	17.93	17.94	18.01	18.01
	JMax (%)	–	40.71	40.27	39.89	39.38
	JGood (%)	–	90.97	90.73	90.51	90.36
	Waiting	–	675	**674**	**673**	**674**
	Time (ms)	–	76,439	77,514	79,984	81,799
	TOPSIS	–	0.24	0.43	0.59	0.79
3	Breach (%)	–	**17.84**	17.94	17.96	17.98
	JMax (%)	–	40.56	39.97	39.72	**39.08**
	JGood (%)	–	90.91	90.55	90.41	**90.16**
	Waiting	–	**672**	**674**	**674**	**673**
	Time (ms)	–	77,568	78,778	82,518	86,853
	TOPSIS	–	0.34	0.58	0.68	**0.87**

Table 6.3: Results obtained for different lengths of rolling horizon and numbers of scenarios, where each column (row) represents a different H (O) value

rolling horizon H are 1, 2 and 3, and for the number of scenarios E are 1, 2 and 3. Values higher than those would sometimes cause the software to run out of memory. The results obtained are shown in Table 6.3. For the sake of comparison, results without using rolling horizon are also presented.

There is not a large variation in the value of the breach or the squared waiting time objective functions for different numbers of days or scenarios in the rolling horizon. The other objective functions seem to improve when either the number of days or the number of scenarios are increased. However, this improvement is accompanied by an increase in the CPU time.

Table 6.4 shows additional performance measures when using rolling horizon. The number of efficient solutions found on each day does not increase when the number of days in the rolling

		0	1	2	4	7
0	Patient Incr. (%)	1,306	–	–	–	–
	Ideal (%)	56	–	–	–	–
	Efficient Solutions	2.63	–	–	–	–
	Exceeded Time (%)	34.93	–	–	–	–
	Variables per Run	2,559	–	–	–	–
	Constr. per Run	153	–	–	–	–
1	Patient Incr. (%)	–	1,334	1,358	1,395	1,486
	Ideal (%)	–	55	55	55	53
	Efficient Solutions	–	2.49	2.47	2.43	2.40
	Exceeded Time (%)	–	37.02	40.73	40.86	43.10
	Variables per Run	–	2,787	2,982	3,185	3,761
	Constr. per Run	–	156	159	163	173
2	Patient Incr. (%)	–	1,364	1,409	1,486	1,667
	Ideal (%)	–	55	54	54	53
	Efficient Solutions	–	2.36	2.30	2.19	2.11
	Exceeded Time (%)	–	40.37	43.70	47.73	52.99
	Variables per Run	–	3,079	3,310	3,771	4,746
	Constr. per Run	–	197	215	247	274
3	Patient Incr. (%)	–	1,391	1,459	1,576	1,849
	Ideal (%)	–	54	54	53	52
	Efficient Solutions	–	2.35	2.19	2.06	2.34
	Exceeded Time (%)	–	42.10	47.07	53.62	57.44
	Variables per Run	–	3,141	3,601	4,349	5,662
	Constr. per Run	–	240	277	341	393

Table 6.4: Performance details for different lengths of rolling horizon and numbers of scenarios, where each column (row) represents a different H (O) value

horizon is incremented, as could be expected. Similarly to the case with pre-scheduling, this is likely due to the time limit being frequently exceeded before all solutions are found.

As the number of patients considered increases, the probability of the constructive algorithm finding the ideal solution decreases slightly. With a larger number of patients being considered by the scheduling algorithm, the number of possible schedules is also larger. This makes it more unlikely that an ideal solution will be found by the constructive algorithm.

As expected, the number of variables and constraints also increase with the increase in the number of patients considered. This causes the problem to become increasingly difficult for the solver, requiring more CPU time. In addition, the number of times when the time limit is exceeded also grows. With the largest number of days and scenarios, around 59% of the runs exceeded the time limit. In addition, the more runs exceed the time limit, the fewer efficient solutions are found due to the early termination of the algorithm.

6.4 Rescheduling Approach

The possibility of changing previously scheduled patients to accommodate new ones is also investigated. When creating a schedule for new patients, it might be possible to achieve a good schedule if patients that have already been scheduled are allowed to have their treatment delayed slightly further. Although this is contradictory with the concept of a robust schedule, it might be interesting to see if and how much the solution could be improved.

Changing previously scheduled patients is usually not desired by the hospital staff or by patients. It implies additional work to the hospital staff, that of calling patients and informing that their scheduled has been changed. There might be clashes between plans that a patient has already made and the new schedule, causing dissatisfaction from both ends. However, it might achieve very good results. Depending on what is the percentage of patients that actually are rescheduled and the improvements on the overall schedule, rescheduling could prove to be an interesting approach.

The rescheduling approach chosen for this work aims at minimising patient inconvenience. When creating a schedule for a group of new patients, previously scheduled patients are considered for rescheduling only if their treatment should start at least $Hres$ days from the current day, where $Hres$ is referred to as the rescheduling horizon. If we consider that patients who start their treatment in the next few days would be the most inconvenienced if rescheduled, it is possible to minimise this inconvenience by using a large rescheduling horizon.

To avoid situations where a patient is rescheduled several times, a limit is imposed on the number of times that a patients can be rescheduled. This limit value is referred to as R. Patients who have already been rescheduled R times will not be considered for rescheduling on the remaining days, regardless of how far from current day their treatment should start. In addition, only routine patients are allowed to be rescheduled.

For this method, the objective of minimising the number of rescheduled patients is included. The following input data is introduced:

- $\mathcal{N}^{res}$: set of patients who are available to be rescheduled,
- $\hat{i}_j$: linac where patient j was previously scheduled,
- $\hat{k}_j$: day when the first session of patient j was previously scheduled.

For the sake of simplification, treatments are only allowed to be postponed when being rescheduled, i.e. they cannot start earlier. Constraint 6.13 guarantees this.

$$x_{ijk1} = 0 \quad i = 1, \dots, M, j \in \mathcal{N}^{res}, k = 1, \dots, \hat{k}_j - 1 \tag{6.13}$$

When allowing rescheduling, minimising the weighted number of rescheduled patients is included as an objective. The total number of objectives Y is adjusted to 5 and the new objective is defined as:

$$f_5(\mathbf{x}) = \sum_{j}^{\mathcal{N}^{res}} w_j \ (1 - x_{\hat{i}_j j \hat{k}_j 1}). \tag{6.14}$$

Using rescheduling in combination with pre-scheduling and rolling horizon was not possible for this work. These methods use a large amount of memory and the software would often result in a memory fault. Experiments are run using the best variable threshold values achieved in Section 5.6 instead: an utilisation threshold for urgent and routine patients which is 100% on the next day, linearly decreases for 14 days to 95%, and remains at 95% for the subsequent days. It should be noted that the results are slightly different from the previous experiments with variable threshold presented in Table 5.8 because the lexicographical approach is used in those experiments, while the Sylva–Crema algorithm is used here.

The values experimented with for the rescheduling horizon $Hres$ are ∞, 21, 14, 7 and 1, where a value of ∞ for the rescheduling horizon means that no rescheduling is allowed and a value of 1 means that any patient who has not started treatment yet can be rescheduled. The rescheduling limit is set to 1, 2 and 3. Results are shown in Table 6.5.

		∞	21	14	7	1
0	Breach (%)	24.91	–	–	–	–
	JMax (%)	42.94	–	–	–	–
	JGood (%)	89.33	–	–	–	–
	Waiting	755	–	–	–	–
	Rescheduled (%)	0.00	–	–	–	–
	Time (ms)	18	–	–	–	–
1	Breach (%)	–	23.97	23.53	21.65	20.72
	JMax (%)	–	42.97	42.07	42.04	40.18
	JGood (%)	–	89.33	89.66	90.65	89.79
	Waiting	–	727	711	698	696
	Rescheduled (%)	–	1.92	2.70	3.97	4.94
	Time (ms)	–	2,076	3,780	7,840	10,719
2	Breach (%)	–	23.94	23.40	20.66	18.77
	JMax (%)	–	43.02	41.91	41.79	38.97
	JGood (%)	–	89.31	89.92	91.04	89.92
	Waiting	–	720	699	686	680
	Rescheduled (%)	–	2.50	3.54	5.52	7.16
	Time (ms)	–	4,592	7,324	15,194	21,046
3	Breach (%)	–	23.92	23.36	20.20	**17.49**
	JMax (%)	–	43.00	41.80	42.01	**38.11**
	JGood (%)	–	**89.26**	89.96	91.25	89.88
	Waiting	–	715	693	680	**672**
	Rescheduled (%)	–	2.80	3.98	6.26	8.36
	Time (ms)	–	5,425	8,562	17,789	24,178

Table 6.5: Results obtained for different lengths of rescheduling horizon and rescheduling limits, where each column (row) represents a different $Hres$ (R) value

All objectives improve significantly when allowing patients to be rescheduled. Apart from the JCCO good practice objective function (JGood), all objective functions improve when the number of patients allowed to be rescheduled increases by changing either $Hres$ or R. As the rescheduling horizons is decremented to 14 days or less, the value found for JGood is worse. This likely happens since it is possible for the algorithm to find a greater number of efficient schedules which have better values for the other objectives. With more schedules with smaller values of the other objective functions at the cost of JGood, it is more likely that such a schedule is chosen and implemented for each day.

Table 6.6 shows the increase in the number of patients considered daily, the percentage of cases where the constructive algorithm finds the ideal solution, the average number of efficient solutions found per run, the percentage of cases where the solver exceeds the time limit, and the

		∞	21	14	7	1
0	Patient Incr. (%)	0	–	–	–	–
	Ideal (%)	82	–	–	–	–
	Efficient Solutions	1.04	–	–	–	–
	Exceeded Time (%)	0.00	–	–	–	–
	Variables per Run	146	–	–	–	–
	Constr. per Run	130	–	–	–	–
1	Patient Incr. (%)	–	45	59	67	74
	Ideal (%)	–	75	72	70	67
	Efficient Solutions	–	1.44	1.65	2.25	2.91
	Exceeded Time (%)	–	0.74	1.29	2.38	2.23
	Variables per Run	–	318	366	383	396
	Constr. per Run	–	155	165	179	190
2	Patient Incr. (%)	–	59	79	98	107
	Ideal (%)	–	75	72	69	67
	Efficient Solutions	–	1.50	1.77	2.60	3.66
	Exceeded Time (%)	–	1.90	2.83	4.20	3.79
	Variables per Run	–	384	451	492	506
	Constr. per Run	–	159	172	188	205
3	Patient Incr. (%)	–	63	85	112	124
	Ideal (%)	–	75	72	69	67
	Efficient Solutions	–	1.51	1.76	2.66	3.91
	Exceeded Time (%)	–	2.32	3.38	4.91	4.17
	Variables per Run	–	391	462	533	548
	Constr. per Run	–	160	172	191	210

Table 6.6: Performance details for different lengths of rescheduling horizon and rescheduling limits, where each column (row) represents a different $Hres$ (R) value

numbers of variables and constraints per run.

Since patients are only considered for rescheduling when their start date is within $Hres$ or more days and they were rescheduled less than R times, the shorter the rescheduling horizon $Hres$ and the larger is the rescheduling limit R, the larger is the number of patients considered for rescheduling per run. Similarly to previous cases, this increase causes the problem to becomes slightly more complex for the constructive algorithm, which finds the ideal solution in a smaller number of runs. The number of efficient solutions found per day also grows with the number of patients considered. As five objectives are considered here, in contrast with four in previous sections, the algorithm is able to find a number of efficient solutions per run which is larger than previous cases (at most 3.91 for rescheduling, in comparison with 3.08 for pre-scheduling and 2.63 for rolling horizon). In addition, a larger number of runs meet the time limit and are able to

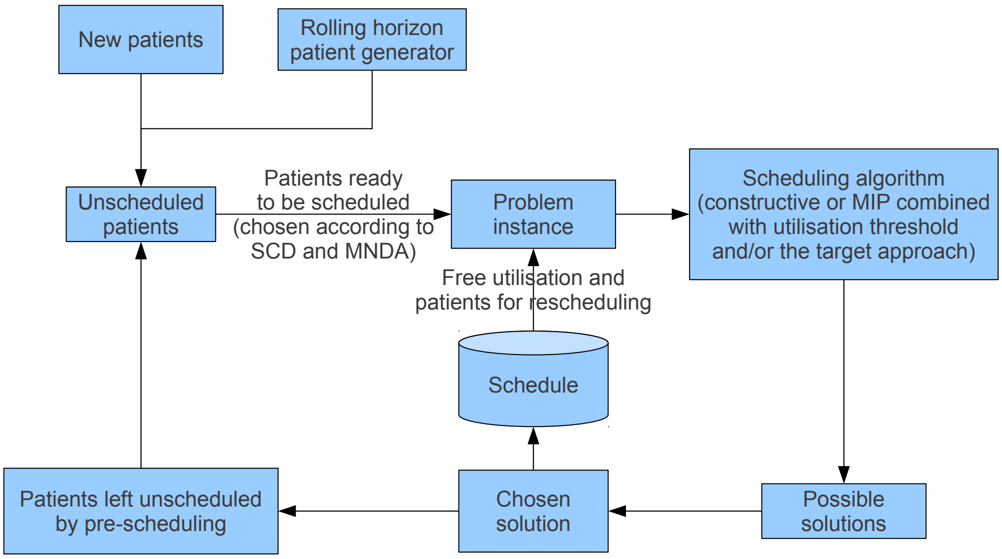


Figure 6.3: Diagram of showing how the rolling horizon approach fits in the general algorithm

finish, finding all schedules necessary to represent the whole Pareto frontier.

The increase in patients considered per run also causes an increase in the number of variables and constraints. However, not as many variables and constraints are created as in the robust approaches, since the increase in patients considered is not so large.

6.5 Comparison of the Main Approaches

Throughout this thesis, many different approaches have been presented to tackle the radiotherapy treatment scheduling problem. In this section, a brief comparison of the results obtained is given. To better understand the full algorithm developed and where each algorithm or approach fit in the general picture, a diagram is presented in Figure 6.3.

Tables 6.7 and 6.8 show the best results obtained in Chapter 4, Chapter 5 and the approaches presented in this chapter. Column “Method” specifies the approach used, where the approaches compared are:

Method	Breach (%)	JMax (%)	JGood (%)	Waiting	Rescheduled (%)	Time (ms)
Constructive	25.96	44.39	90.39	751	0.00	0
Var. Threshold	24.86	42.83	**89.20**	755	0.00	3
Robust	17.98	39.08	90.16	**673**	0.00	86,853
Rescheduling	**17.49**	**38.11**	89.88	**672**	8.36	24,178

Table 6.7: Results obtained for the main approaches

Constructive: constructive algorithm, with a linac utilisation threshold of 94% for routine patients, while creating schedules for emergency and routine patients every day and for urgent patients three times per week, when the release date of urgent and routine patients is within 21 and 0 days of the current date, respectively (best results achieved in Chapter 4 and presented in Table 4.5).

Var. Threshold: ILP model with lexicographical ordering of objectives, while creating schedules for all patients every day, for routine patients only when their release date arrives, with a linac utilisation threshold for urgent and routine patients being 100% on the next day, while linearly decreasing for 14 days to 95%, and remaining at 95% for the subsequent days (best results achieved in Chapter 5 and presented in Table 5.8).

Robust: ILP model with the Sylva–Crema algorithm, pre-scheduling all patients who have arrived combined with 3 sets of patients, each containing a number of predicted patients which may arrive in the next 7 days (best results achieved in Section 6.3 presented in Table 6.3).

Rescheduling: ILP model with the Sylva–Crema algorithm, considering for rescheduling any routine patient who has not started treatment and has not been rescheduled or has been rescheduled less than 3 times, with a linac utilisation threshold for urgent and routine patients being 100% on the next day, while linearly decreasing for 14 days to 95%, and remaining at 95% for the subsequent days (best results achieved in Section 6.4 presented in Table 6.5).

All approaches have their own advantages and disadvantages. The constructive approach is easy to implement and its computational complexity is very small, making the required CPU

Method	Patient Incr. (%)	Efficient Solutions	Exceeded Time (%)	Variables per Run	Constr. per Run
Constructive	0	1.00	0.00	0	0
Var. Threshold	0	1.00	0.00	159	136
Robust	1,849	2.34	57.44	5,662	393
Rescheduling	124	3.91	4.17	548	210

Table 6.8: Performance details for the main approaches

time nearly zero. In addition, it is the only approach which does not require the acquisition of an integer programming solver. Although such solvers can usually be obtained at no cost by the academic community, their commercial licenses can be very expensive. There are open source alternatives, such as CBC, the branch-and-cut software distributed by COIN-OR (Computational Infrastructure for Operations Research, 2011), but their performance is not always as good. However, this approach does not achieve very attractive results when comparing to the other approaches. Particularly, the values found for the breach and the squared waiting time objective functions are considerably larger than those found by the other approaches.

The variable threshold approach achieves better results than the constructive approach with very little additional CPU cost. The values achieved for the breach and squared waiting time objective functions are considerably lower. For the JCCO good practice objective function, the values are significantly better, but the difference is not so considerable. However, for the JCCO maximum acceptable objective function, the values are slightly worse. The values found for the breach, JCCO maximum acceptable and squared waiting time objective functions are also worse than those found for the robust and rescheduling approaches.

The robust method achieves very good values for the breach, JCCO maximum acceptable and squared waiting time objective functions. This method is also able to provide a number of solutions to the decision maker, giving him/her a better insight into the possible trade-offs that can be achieved amongst the objectives. It should be noted that, although the average number of solutions per run is 2.3, an ideal solution is found by the constructive algorithm in 52% of the runs. If only runs when the Sylva–Crema algorithm is used are considered, the average of efficient solutions found per run is 4.9.

However, the robust method has a very high required CPU time, that of 1m27s per run, and 57% of the runs exceed the time limit of 10 minutes. This extra CPU time is caused by the large number of variables and constraints resulting from the increase in the number of patients considered per run.

The rescheduling method achieves significantly better results for 3 of the 4 main objectives considered compared to the other methods. The exception is the JCCO good practice objective function, for which there is not much variation amongst all methods. The required CPU time is higher than the constructive and variable threshold approaches, but it is much lower than for the robust approach, with an average of 24s per run and 4.17% of the runs exceeding the time limit. The main drawback of this approach is that it requires patients to be rescheduled, which is not desirable.

Taking all the above into account, the author recommends the robust approach to be implemented at the radiotherapy centre. It achieves very good results, comparing with the other approaches presented in this work. Although the rescheduling approach achieves slightly better results, allowing any routine patient who has not started treatment to be rescheduled up to three times is likely to be too high a cost. All other rescheduling limits investigated in Table 6.5 achieve worst results than the robust approach.

6.6 Summary

In this chapter, an adaptation of the Sylva–Crema algorithm for minimisation problems is presented. This algorithm is an exact Pareto-type multi-objective approach capable of finding a set of solutions representing the whole Pareto frontier for integer linear programing problems.

Two robust approaches are presented. The first approach is referred to as pre-scheduling, and consists of considering patients whose booking request already arrived but whose scheduling date is still a specific number of days away. By pre-scheduling all routine patients and writing their schedule only when their release date arrives, it was possible to achieve considerably better results than previous approaches, specially for the breach objective function.

The second robust approach is called rolling horizon, and it consists of trying to predict which patients will arrive in the near future and consider them when creating a schedule, leaving empty time slots on linacs to accommodate the predicted patients. In addition, more than one scenario of future patients can be generated and the schedule which achieves the best result considering all future scenarios is implemented. The approach is combined with pre-scheduling to investigate if the previous approach can be further improved. Experiments suggest that the objective functions of both JCCO due dates can be improved by increasing the numbers of days and scenarios in the rolling horizon. However, there is no large variation in the values found for the breach and squared waiting time objective functions. Due to high memory requirements of this method, a largest rolling horizon investigated is of 7 days and 3 scenarios.

In addition to the robust approaches, a rescheduling approach is presented. The goal is to allow patients whose treatment start more than a specific number of days away to be rescheduled. This number of days is referred to as rescheduling horizon. In addition, a limit is imposed on the number of times a patient can be rescheduled. Experiments indicate that the smaller is the rescheduling horizon and the larger is the rescheduling limit, the better are the values found for the breach, JCCO maximum acceptable and squared waiting time objective functions. The values found for the JCCO good practice objective function do not improve in the majority of experiments, and are, in fact, significantly worse for a few values of rescheduling horizon and rescheduling limit.

In the overall comparison of the main methods presented in this thesis, the author finds that the robust methods presented in this chapter can achieve the most interesting results. The combination of pre-scheduling and rolling horizon achieves values for all objective functions which are almost as low as those achieved by rescheduling, but without the necessity of changing the schedule of any patients.

"Now this is not the end. It is not even the beginning of the end. But it is, perhaps, the end of the beginning."

Sir Winston Churchill (1874–1965)

Chapter 7

Conclusions

This thesis focuses on the radiotherapy treatment scheduling problem, which is faced by many radiotherapy centres. The dynamic arrival of patients with a different rate for each period of the year and of the week, and the large diversity of treatment types with linac eligibility constraints and different number of sessions per week and in total required by patients make this a very challenging problem.

The work was conducted in collaboration with the radiotherapy centre of the City Hospital Campus, Nottingham University Hospitals NHS Trust, UK. Data were collected about the daily income of patients, and analysed in Chapter 3. This analysis proved essential for a proper understanding of the problem and for the designing and planning of the majority of methods and experiments presented in this thesis. In addition, it has helped the hospital staff in gaining better insight into the problem of radiotherapy patient scheduling.

Four constructive approaches are proposed for this problem and investigated in Chapter 4. Their main advantage is their simplicity and small computational requirements. The first tries to postpone the treatment start of patients of lower priority in order to leave more space on the linacs machines for future patients of higher priority. The second establishes an utilisation threshold on linacs machines for patients of each waiting list status, such that a session is not allowed to be scheduled on given machine and day if it would cause this threshold to be exceeded. The third approach, referred to as SCD approach, limits the creating of schedules for patients of each waiting list status to specific days of the week in order to allow the accumulation of patients to increase the search space and find a better schedule. The fourth approach, referred to as $MNDA$ approach, limits the maximum number of days in advance with which schedules may be created for patients of each waiting list status. The best combination of approaches found by the experiments consists of:

- establishing a threshold of 92% of the machine utilisation for routine patients, such that these patients can no longer be scheduled on a specific machine and day when the utilisation for that day reaches this threshold,
- creating schedules every day for emergency patients, but only once per week for urgent and routine patients,
- and creating schedules for emergency patients on the day their booking request arrives, and for urgent and routine patients only when their release date is within 7 days.

An Integer Linear Programming (ILP) model for the radiotherapy treatment scheduling problem is presented in Chapter 5. By populating an instance of the model and solving it with an integer programming solver, it is possible to obtain the optimal schedule for the patients currently being scheduled on each day. This approach is used on a daily basis instead of the constructive algorithm presented in the previous chapter. In order to deal with multiple objectives, a lexicographical ordering is used.

The model is combined with the SCD and $MNDA$ approaches in order to investigate their potential benefits. Furthermore, all parameter values for these two approaches are experimented with to find the most effective combinations. A generalisation of the utilisation threshold approach previously presented for the constructive algorithm is developed. It is based on the idea that it is not necessary to impose any utilisation restrictions on the day following the current day (i.e. tomorrow), since the next time a schedule is created, no patients will be able to start treatment on that day. The best combination of approaches found by the experiments consists of:

- creating schedules every day for all patients,
- creating schedules for emergency and urgent patients on the day their booking request arrives, and for routine patients only when their release date arrives, and
- establishing an utilisation threshold for urgent and routine patients which is set to 100% on the next day, linearly decreases for 14 days to 95%, and remains at 95% for the subsequent days.

In addition, the model is tested against different sizes of the problem and different linac allocation policies. The expected increases of 10% in the rate of patient arrival and 2 more linacs which emit all radiation types are considered. This was of special interest to the hospital. Two changes in linac allocation policy are considered. The first change would be to consider all linacs that can deliver the required radiation type. The second change would be to allow for different sessions of the same patient to be scheduled on different linacs, increasing the number of feasible schedules. The results suggest that the model still works well for this expected increase, finding schedules optimal for patients currently being scheduled in a reasonable amount of time. The implementation of the first change can greatly increase the quality of schedules by improving all objective functions. However, the implementation of the second change does not improve the quality of the created schedules and greatly increases the complexity and computational requirements for the problem.

Chapter 6 presents a Pareto-type multi-objective algorithm, robust and rescheduling approaches. The multi-objective algorithm is an adaptation for minimisation problems of the algorithm presented by Sylva & Crema (2004). It is capable of finding a set of solutions representing the whole Pareto frontier for integer programming problems. The first robust approach presented is referred to as pre-scheduling, and it consists of creating schedules for patients who are not yet available to be scheduled, but without writing the schedule chosen for these patients on the booking system. This is later combined with the second robust approach, referred to as rolling horizon, which consists of trying to predict which patients will arrive in the near future and consider them when creating the current schedule. To achieve a more robust schedule, several possible scenarios of future patients are generated and the schedule which achieves the best results considering all scenarios is implemented.

A rescheduling approach is developed to be compared with the robust approach. To minimise the inconvenience caused to patients, limits are imposed on the minimum number of days in advance and on the number of times each patient can be rescheduled. Experiments are run with different values for these limits to investigate the trade-off in the number of rescheduled patients and the improvement in the other objective functions.

A comparison is made including the best combinations of approaches found using the constructive algorithm, the ILP model, the robust and the rescheduling approaches. It concludes that each of the evaluated methods has their advantages and disadvantages over the others. However, the robust approach is considered the best approach overall, since it achieves very good results without need of patient rescheduling.

At this point, it is possible to refer back to the three hypotheses presented in Section 1.2:

Hypothesis 1: *An Integer Linear Programming approach can improve the schedule created by constructive approaches.* The method which creates an instance of the ILP model developed for the radiotherapy treatment scheduling problem and solves it starting from the schedule provided by the constructive approach can improve the starting schedule in many cases. Even though this method does not include the future patients, the overall schedule is also improved in the experiments.

Hypothesis 2: *Robust advanced approaches can improve the quality of the schedule found by simpler myopic approaches for the radiotherapy treatment scheduling problem.* The pre-scheduling and rolling horizon approaches confirm this hypothesis, as they are able to achieve schedules of higher quality than either the constructive algorithm or the mathematical model when used on their own.

Hypothesis 3: *Rescheduling approaches can find schedules of higher quality than robust approaches at an acceptable number of rescheduled patients for the radiotherapy treatment scheduling problem.* The rescheduling approach is able to improve the quality of the schedule generated by the robust approach for the majority of the objectives considered. However, the improvement achieved is not considerable and the number of patients rescheduled is too high. Therefore, this hypothesis may have to be rejected.

It is important to note that not all combinations of parameters have been tested for the approaches, since that would require an immense and unrealistic amount of computational time. Instead, a subset of the combinations has been tested.

7.1 Reflection

The main limitation of this work is that the experimental data has been generated using a data-driven approach instead of developing an explicit data model. Although it was the simplest approach at the time, it has some drawbacks. Changing the distribution of an input parameter for an experiment is not as easy, if at all possible. Developing an explicit data model for this problem would make this analysis possible.

Another limitation is due to integer programming. It is an adequate approach to this problem up to Section 6.2, when pre-scheduling is introduced and the problem starts to become too large. As previously said, it was not possible to run experiments combining pre-scheduling, rolling horizon and re-scheduling for this reason. A meta-heuristic tailored for this problem might succeed where integer programming did not.

7.2 Impact

This work is part of the EPSRC funded project Novel Approaches to Radiotherapy Planning and Scheduling in the NHS (EP/C549511/1), which is currently being conducted with the collaboration of the radiotherapy centre of the Nottingham University Hospitals UK, NHS Trust. More details about the project can be found on the website: `http://www.asap.cs.nott.ac.uk/`

`projects/narps/`. All data used in the experiments is based on real data collected from the hospital, and all experiments are conducted considering the hospital's policies and preferences.

In addition to what is presented in this work, the full project also encompasses the development of software for scheduling the pre-treatment stage of radiotherapy, as well as a software tool to help oncologists in planning the radiotherapy treatment. At the time of writing thesis, the development of the algorithm for scheduling of the pre-treatment is under way. When it is ready, it will be made part of the scheduling software, so the hospital staff can create a schedule for both phases in one go.

A prototype with a graphical user interface has been developed as a result of this work and is detailed in Appendix B. It is currently being evaluated by the hospital staff. A letter from the hospital acknowledging their satisfaction with the project to this point is given in Appendix C. Once the pre-treatment scheduling module of the software is complete, the hospital staff intents to evaluate the complete scheduling tool, before fully adopting it. The pre-treatment scheduling module is part of another PhD thesis and should be completed in mid 2011.

7.3 Future work

A few extensions of this work could prove interesting. Additional objectives could be considered. For example, in the variable threshold approach, instead of using the utilisation thresholds as constraints, they could be used as an additional objective function where their violation is minimised. In combination with the Sylva–Crema algorithm, it could provide the decision maker with a greater number of more diverse schedules, which can be interesting on days when an exceptionally large number of patients are being scheduled.

Another possibility would be to use the Sylva–Crema algorithm with a larger δ_m, in order to find a set of efficient solutions where the objective function values are more spaced apart in the cases where the algorithm exceeds the time limit. If the algorithm reaches an end where no more schedules are found, but the time limit has not been reached yet, it is possible to decrement the value of δ_m to find additional schedules. Furthermore, Sylva & Crema (2007) presents an adaptation of the algorithm which aims at finding a solution at each step with maximum distance from the set dominated by the solutions which have already been found. An implementation of this approach could also be of interest to find diverse solutions in the cases where the algorithm exceeds its time limit.

As shown in Chapter 5, the integer programming model presented can be used to solve real-world problems on a daily basis requiring very little CPU time. However, when combined with the robust methods presented in Chapter 6, the resulting model instances can be exceptionally large and take a long time to solve. This process could be made faster by using more advance mathematical programming techniques, such as Lagrangian relaxation (Fisher, 1981) and column generation (Lübbecke & Desrosiers, 2005). Another option would be to use a multi-objective heuristic in these cases, such as the Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) (Deb et al., 2002). The tutorial by Konak et al. (2006) could provide a good start in that direction.

Using rolling horizon with different methods of predicting which patients will arrive in the near future could also prove to be interesting future work. Instead of considering a large number of scenarios, it might be possible to use an approximation of the number and type of future patients. The resulting approach could result in a smaller number of variables and constraints, which would be easier to solve in acceptable time. Unlike the pre-scheduling approach, rolling horizon can be applied even if schedules for patients are created as soon as they arrive ($MNDA = \infty$). Using rolling horizon on its own with a different and more efficient forecasting method could also achieve good results.

Another possibility is to use Approximate Dynamic Programing (Powell, 2009, 2010), a new technique that has been achieving very good results for other dynamic problems which are similar to the one tackled in this thesis (Cant, 2006; Yu, 2007; Novoa & Storer, 2009; Schütz & Kolisch, 2010). Perhaps a brand new method would be able to achieve better results.

Future work can also be done in the direction of a deeper study of the distributions of the input data and a further analysis of changes in these distributions. With a better undersanding of the data, it might be possible to implement a better patient generator for the rolling horizon approach, or might even give us a completely different idea from what has been tried in this thesis.

Other factors can be included in the problem to make it more realistic, such as including patient preferences for being treated in the mornings or afternoons. Another possibility is to include transportation constraints, since some patients require hospital transportation to the treatment centre and then home.

Furthermore, the methods presented here can be used in combination with the methods developed for the scheduling of pre-treatment in order to present a full solution to the decision maker.

The algorithms presented in this thesis can also be adapted to other similar health-care problems, such as chemotherapy scheduling (Agur et al., 2006). This may include any other problem where patients (or jobs) have different priorities, arrival times, release and due dates.

Bibliography

Agur, Z., Hassin, R., & Levy, S. (2006). Optimizing chemotherapy scheduling using local search heuristics. *Operations Research*, *54*(5), 829–846.

Alexopoulos, C., Goldsmana, D., Fontanesib, J., Kopaldb, D., & Wilsonc, J. R. (2008). Modeling patient arrivals in community clinics. *Omega*, *36*(1), 33–43.

Ash, D., Barretty, A., Hinksz, A., & Squirex, C. (2004). Re-audit of radiotherapy waiting times 2003. *Clinical Oncology*, *16*(6), 387–394.

Bagchi, U., Sullivan, R. S., & Chang, Y.-L. (1987). Minimizing mean squared deviation of completion times about a common due date. *Management Science*, *33*(7), 894–906.
URL `http://www.jstor.org/stable/2632140`

Bailey, N. T. J. (1952). A study of queues and appointment systems in hospital out-patient departments, with special reference to waiting-times. *Journal of the Royal Statistical Society. Series B (Methodological)*, *14*(2), 185–199.
URL `http://www.jstor.org/stable/2983867`

Balas, E. (1965). An additive algorithm for solving linear programs with zero-one variables. *Operations research*, *13*(4), 517–546.

Balas, E., Ceria, S., & Cornuéjols, G. (1996a). Mixed 0-1 programming by lift-and-project in a branch-and-cut framework. *Management Science*, *42*(9), 1229–1246.
URL `http://www.jstor.org/stable/2634434`

Balas, E., Ceria, S., Cornuéjols, G., & Natraj, N. (1996b). Gomory cuts revisited. *Operations Research Letters*, *19*, 1–9.

Barendsen, G. W. (1982). Dose fractionation, dose rate and iso-effect relationships for normal tissue responses. *International Journal of Radiation Oncology*Biology*Physics*, *8*(11), 1981–1997.

Bitran, G. R. (1977). Linear multiple objective programs with zero-one variables. *Mathematical Programming*, *13*(1), 121–139.

Board of the Faculty of Clinical Oncology (1998). A national audit of waiting times for radiotherapy. Tech. rep., Royal College of Radiologists, The Royal College of Radiologists, London.
URL `http://www.rcr.ac.uk/index.asp?PageID=149&PublicationID=50`

Cant, L. (2006). *LIFE-SAVING DECISIONS: A Model for Optimal Blood Inventory Management*. Undergraduate senior thesis, Department of Operations Research and Financial Engineering, Princeton University.

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

Cayirli, T., & Veral, E. (2003). Outpatient scheduling in health care: A review of literature. *Production and Operations Management*, *12*(4), 519–549.

Cayirli, T., Veral, E., & Rosen, H. (2006). Designing appointment scheduling systems for ambulatory care services. *Health Care Management Science*, *9*(1), 47–58.

Computational Infrastructure for Operations Research (2011). COIN-OR: Cbc. Website. Accessed on the 11th of January of 2011.
URL `http://www.coin-or.org/projects/Cbc.xml`

Conforti, D., Guerriero, F., & Guido, R. (2008). Optimization models for radiotherapy patient scheduling. *4OR: A Quarterly Journal of Operations Research*, *6*(3), 263–278.

Conforti, D., Guerriero, F., & Guido, R. (2010a). Non-block scheduling with priority for radiotherapy treatments. *European Journal of Operational Research*, *201*(1), 289–296.

Conforti, D., Guerriero, F., Guido, R., Cerinic, M. M., & Conforti, M. L. (2010b). An optimal decision making model for supporting week hospital management. *Health Care Management Science*, *14*(1), 74–88.

Conforti, D., Guerriero, F., Guido, R., & Veltri, M. (2011). An optimal decision-making approach for the management of radiotherapy patients. *OR Spectrum*, *33*(1), 123–148.

Dantzig, G. B. (1959). Note on solving linear programs in integers. *Naval Research Logistics Quarterly*, *6*(1), 75–76.

Deb, K. (2005). *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, chap. Multi-objective Optimization, (pp. 273–316). Springer.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197.

Department of Health (2000). The NHS cancer plan: a plan for investment, a plan for reform. Tech. rep., NHS (National Health Services), UK, United Kingdom.
URL `http://www.dh.gov.uk/en/Publicationsandstatistics/Publications/PublicationsPolicyAndGuidance/DH_4009609`

Department of Health (2004). The NHS cancer plan: Providing a patient centred service. Tech. rep., NHS (National Health Services), UK, United Kingdom.
URL `http://www.dh.gov.uk/en/Publicationsandstatistics/Publications/PublicationsPolicyAndGuidance/DH_4092531`

Department of Health (2005). One page guide to 2005 cancer waiting times targets. Tech. rep., National Cancer Waits Project, UK, United Kingdom.
URL `http://www.docstoc.com/docs/24374159/cancer-waiting-times`

Department of Health (2007). Radiotherapy: developing a world class service for England. Tech. rep., NHS (National Health Services), UK, United Kingdom.
URL `http://www.dh.gov.uk/en/Publicationsandstatistics/Publications/PublicationsPolicyAndGuidance/DH_074575`

Dodwell, D., & Crelin, A. (2006). Waiting for radiotherapy. *British Medical Journal (BMJ)*, *332*, 107–109.

Dominic, P. D. D., Kaliyamoorthy, S., & Saravana Kumar, M. (2004). Efficient dispatching rules for dynamic job shop scheduling. *International Journal of Advanced Manufacturing Technology*, *24*(1–2), 70–75.

Drinkwater, K. J., & Williams, M. V. (2008). Re-audit of radiotherapy waiting times in the united kingdom, 2007. Tech. rep., The Royal College of Radiologists, 38 Portland Place, London W1B 1JQ, UK.
URL `http://rcrad.org.uk/docs/general/pdf/ReportUKFINAL100408.pdf`

Dueck, G., & Scheuer, T. (1990). Threshold accepting: A general purpose optimization algorithm appearing superior to simulated annealing. *Journal of Computational Physics*, *90*(1), 161–175.

Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, *7*(1), 1–26.
URL `http://www.jstor.org/stable/2958830`

Erdelyi, A., & Topaloglu, H. (2009). Computing protection level policies for dynamic capacity allocation problems by using stochastic approximation methods. *IIE Transactions*, *41*(6), 498–510.

Fetter, R. B., & Thompson, J. D. (1966). Patients' waiting time and doctors' idle time in the outpatient setting. *Health Services Research*, *1*(1), 66–90.

Fisher, M. L. (1981). The lagrangian relaxation method for solving integer programming problems. *Management Science*, *27*(1), 1–18.

Gerchak, Y., Gupta, D., & Henig, M. (1996). Reservation planning for elective surgery under uncertain demand for emergency surgery. *Management Science*, *42*(3), 321–334.
URL `http://www.jstor.org/stable/2634346`

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, *13*(5), 533–549.

Glover, F. (1989). Tabu Search–Part I. *ORSA Journal on Computing*, *1*(3), 190–206.

Glover, F. (1990). Tabu Search–Part II. *ORSA Journal on Computing*, *2*(1), 4–32.

Gomory, R. E. (1958). Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, *64*(5), 275–278.
URL `http://projecteuclid.org/euclid.bams/1183522679`

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

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

Hall, N. G., & Sriskandarajah, C. (1996). A survey of machine scheduling problems with blocking and no-wait in progress. *Operations Research*, *44*(3), 510–525.

Hwang, C.-L., & Yoon, K. (1981). *Multiple Attribute Decision Making: Methods and Applications*. Lecture Notes in Economics and Mathematical Systems. Berlin, Heidelberg, New York: Springer-Verlag.

International Business Machines (2010). IBM ILOG CPLEX Optimizer. Website. Accessed on the 19th of October of 2010.
URL `http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/`

Johnson, N. L. (1949). Systems of frequency curves generated by methods of translation. *Biometrika*, *1*(36), 149–176.

Joint Council for Clinical Oncology (1993). Reducing delays in cancer treatment: some targets. Tech. rep., Royal College of Radiologists, Royal College of Physicians of London.
URL `http://www.rcr.ac.uk/publications.aspx?PageID=149&PublicationID=136`

Kaandorp, G. C., & Koole, G. (2007). Optimal outpatient appointment scheduling. *Health Care Manage Science*, *10*(3), 217–229.

Kapamara, T., & Petrovic, D. (2009). A heuristics and steepest hill climbing method to scheduling radiotherapy patients. In *International Conference on Operational Research Applied to Health Services (ORAHS), 12–17 July*. Leuven, Belgium.

Kapamara, T., Sheibani, K., Haas, O. C. L., Petrovic, D., & Reeves, C. R. (2006). A review of scheduling problems in radiotherapy. In *Proceedings of the 18th International Control Systems Engineering Conference (ICSE 2006), 5–7 September*, (pp. 207–211). Coventry, UK: Coventry University Publishing.

Kapamara, T., Sheibani, K., Petrovic, D., Hass, O., & Reeves, C. R. (2007). A simulation of a radiotherapy treatment system: A case study of a local cancer centre. In *Proceedings of the 2007 OR Peripatetic Post-Graduate Programme Conference (ORP3), 12–15 September*, (pp. 29–35). Guimarães, Portugal.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*(4598), 671–680.

Klassen, K. J., & Rohleder, T. R. (1996). Scheduling outpatient appointments in a dynamic environment. *Journal of Operations Management*, *14*(2), 83–101.

Klein, D., & Hannan, E. (1982). An algorithm for the multiple objective integer linear programming problem. *European Journal of Operational Research*, *9*(4), 378–385.

Konak, A., Coit, D. W., & Smith, A. E. (2006). Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety*, *91*(9), 992–1007.

Kutanoglu, E., & Sabuncuoglu, I. (1999). An analysis of heuristics in a dynamic job shop with weighted tardiness objectives. *International Journal of Production*, *37*(1), 165–187.

Land, A. H., & Doig, A. G. (1960). An automatic method of solving discrete programming problems. *Econometrica*, *28*(3), 497–520.
URL `http://www.jstor.org/stable/1910129`

Lawler, E. L., & Wood, D. E. (1966). Branch-and-bound methods: A survey. *Operations Research*, *14*(4), 699–719.
URL `http://www.jstor.org/stable/168733`

Léger, C., Politis, D. N., & Romano, J. P. (1992). Bootstrap technology and applications. *Technometrics*, *34*(4), 378–398.
URL http://www.jstor.org/stable/1268938

Lev, B., & Caltagirone, R. J. (1974). Evaluation of various scheduling disciplines by computer systems. In *Proceedings of the 1974 Winter Simulation Conference (WSC'74), 14–16 January*, vol. 1, (pp. 365–370). New York, NY, USA: ACM.

Lev, B., Revesz, G., Shea, F., & Caltagirone, R. (1976). Patient flow analysis and the delivery of radiology service. *Socio-Economic Planning Sciences*, *10*(4), 159–166.

Liu, S., & Ong, H. L. (2002). A comparative study of algorithms for the flow-shop scheduling problem. *Asia-Pacific Journal of Operational Research*, *19*(2), 205–222.

Liu, S., & Ong, H. L. (2004). Meta-heuristics for the mixed-shop scheduling problem. *Asia-Pacific Journal of Operational Research*, *21*(4), 97–115.

Liu, S., Ong, H. L., & Ng, K. M. (2005). Meta-heuristics for minimising the makespan of the dynamic shop scheduling problem. *Advances in Engineering Software*, *36*(3), 199–205.

Lübbecke, M. E., & Desrosiers, J. (2005). Selected topics in column generation. *Operations Research*, *53*(6), 1007–1023.

Mackillop, W. J. (2007). Killing time: The consequences of delays in radiotherapy. *Radiotherapy and Oncology*, *84*, 1–4.

Magerlein, J. M., & Martin, J. B. (1978). Surgical demand scheduling: a review. *Health Services Research*, *13*(4), 418–433.

Mann, H. B., & Whitney, D. R. (1947). On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, *18*(1), 50–60.

Miettinen, K. M. (1999). *Nonlinear multiobjective optimization*. Dordrecht: Kluwer academic.

Moore, C. G., Wilson-Witherspoon, P., & Probst, J. C. (2001). Time and money: effects of no-shows at a family practice residency clinic. *Family Medicine*, *33*(7), 522–527.
URL http://www.ncbi.nlm.nih.gov/pubmed/11456244

Nemhauser, G. L., & Wolsey, L. A. (1989). *Optimization*, vol. 1 of *Handbooks In Operations Research And Management Science*, chap. Integer Programming, (pp. 447–527). New York, NY, USA: Elsevier North-Holland, Inc.

Nottingham University Hospitals, NHS Trust (2011). Nottingham University Hospitals – NHS Trust. Website. Accessed on the 12th of January of 2011.
URL http://www.nuh.nhs.uk/

Novoa, C., & Storer, R. (2009). An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. *European Journal of Operational Research*, *196*(2), 509–515.

Padberg, M., & Rinaldi, G. (1987). Optimization of a 532-city symmetric traveling salesman problem by branch and cut. *Operations Research Letters*, *6*(1), 1–7.

Pareto, V. (1906). *Manuale di economia politica*. Societa Editrice.

Patrick, J., Puterman, M. L., & Queyranne, M. (2008). Dynamic multi-priority patient scheduling for a diagnostic resource. *Operations Research*, *56*(6), 1507–1525.

Petrovic, D., Morshed, M., & Petrovic, S. (2009). Genetic algorithm based scheduling of radiotherapy treatments for cancer patients. In C. Combi, Y. Shahar, & A. Abu-Hanna (Eds.) *Artificial Intelligence in Medicine*, vol. 5651 of *Lecture Notes in Computer Science*, (pp. 101–105). Berlin / Heidelberg: Springer.

Petrovic, D., Morshed, M., & Petrovic, S. (2011). Multi-objective genetic algorithms for scheduling of radiotherapy treatments for categorised cancer patients. *Expert Systems with Applications*, *38*(6), 6994–7002.

Petrovic, S., Leung, W., Song, X., & Sundar, S. (2006). Algorithms for radiotherapy treatment booking. In R. Qu (Ed.) *Proceedings of the 25th Workshop of the UK Planning and Scheduling Special Interest Group (PlanSIG2006), 14–15 December*, (pp. 105–112). Nottingham, UK.

Pinedo, M. L. (2008). *Scheduling: Theory, Algorithms, and Systems*. New York: Springer, third ed.

Podgorelec, V., & Kokol, P. (1997). Genetic algorithm based system for patient scheduling in highly constrained situations. *Journal of Medical Systems*, *21*(6), 417–427.

Powell, W. B. (2009). What you should know about approximate dynamic programming. *Naval Research Logistics*, *56*(3), 239–249.

Powell, W. B. (2010). Merging AI and OR to solve high-dimensional stochastic optimization problems using approximate dynamic programming. *INFORMS Journal on Computing*, *22*(1), 2–17.

Proctor, S., Lehaney, B., Reeves, C., & Khan, Z. (2007). Modelling patient flow in a radiotherapy department. *OR Insight*, *20*(3), 6–14.

Ragaz, J., Olivotto, I. A., Spinelli, J. J., Phillips, N., Jackson, S. M., Wilson, K. S., Knowling, M. A., Coppin, C. M. L., Weir, L., Gelmon, K., Le, N., Durand, R., Coldman, A. J., & Manji, M. (2005). Locoregional radiation therapy in patients with high-risk breast cancer receiving adjuvant chemotherapy: 20-year results of the british columbia randomized trial. *Journal of the National Cancer Institute (JNCI)*, *97*(2), 116–126.

Rajendran, C., & Holthaus, O. (1999). A comparative study of dispatching rules in dynamic flowshops and jobshops. *European Journal of Operational Research*, *116*(1), 156–170.

Reeves, C. R., & Rowe, J. E. (2002). *Genetic Algorithms–Principles and Perspectives, A Guide to GA Theory*. Norwell, Massachusetts, USA: Kluwer Academic Publishers.

Roy, B., & Sussman, B. (1964). Les problemes d'ordonnancement avec contraintes disjonctives. Tech. Rep. 9, SEMA, Note D.S.

Sadki, A., Xie, X., & Chauvin, F. (2010a). Bedload balancing for an oncology ambulatory care unit. In *8th International Conference of Modeling and Simulation (MOSIM'10), 10–12 May*. Hammamet, Tunisia.

Sadki, A., Xie, X., & Chauvin, F. (2010b). Patients assignment for an oncology outpatient unit. In *Proceedings of the 2010 IEEE Conference on Automation Science and Engineering (CASE2010), 21–24 August*, (pp. 891–896). Toronto, Canada.

Schütz, H.-J., & Kolisch, R. (2010). Approximate dynamic programming for capacity allocation in the service industry. Available at SSRN.
URL `http://ssrn.com/abstract=1618315`

Spurgeon, P., Barwell, F., & Kerr, D. (2000). Waiting times for cancer patients in england after general practitioners' referrals: retrospective national survey. *British Medical Journal (BMJ)*, *320*, 838–839.

Steuer, R. E. (1986). *Multiple criteria optimization: theory, computation, and application*. Wiley Series in Probability and Mathematical Statistics–Applied. Singapore: John Wiley & Sons, Inc.

Summers, E., & Williams, M. (2006). Re-audit of radiotherapy waiting times 2005. Tech. rep., The Royal College of Radiologists, London.
URL `http://www.library.nhs.uk/healthmanagement/ViewResource.aspx?resID=78954`

Suresh, V., & Chaudhuri, D. (1993). Dynamic scheduling–a survey of research. *International Journal of Production Economics*, *32*, 53–63.

Sylva, J., & Crema, A. (2004). A method for finding the set of non-dominated vectors for multiple objective integer linear programs. *European Journal of Operational Research*, *158*(1), 46–55.

Sylva, J., & Crema, A. (2007). A method for finding well-dispersed subsets of non-dominated vectors for multiple objective mixed integer linear programs. *European Journal of Operational Research*, *180*(3), 1011–1027.

Thames, H. D., Jr., Withers, H. R., Peters, L. J., & Fletcher, G. H. (1982). Changes in early and late radiation responses with altered dose fractionation: Implications for dose-survival relationships. *International Journal of Radiation Oncology*Biology*Physics*, *8*(2), 219–226.

Thomas, S. J. (2003). Capacity and demand models for radiotherapy treatment machines. *Clinical Oncology*, *15*(6), 353–358.

Thomsen, M. S., & Nørrevang, O. (2009). A model for managing patient booking in a radiotherapy department with differentiated waiting times. *Acta Oncologica*, *48*(2), 251–258.

Turner, K. J., & Bing, Q. (2002). Protocol techniques for testing radiotherapy accelerators. In D. Peled, & M. Vardi (Eds.) *Formal Techniques for Networked and Distributed Sytems – FORTE 2002*, Lecture Notes in Computer Science, (pp. 81–96). Springer Berlin / Heidelberg.

Vermeulen, I. B., Bohte, S. M., Elkhuizen, S. G., Lameris, H., Bakker, P. J., & Poutré, H. L. (2009). Adaptive resource allocation for efcient patient scheduling. *Articial Intelligence in Medicine*, *46*(1), 67–80.

Vermeulen, I. B., Bohte, S. M., Elkhuizen, S. G., Lameris, J. S., Bakker, P. J. M., & Poutré, J. A. L. (2007). Adaptive optimization of hospital resource calendars. In R. Bellazzi, A. Abu-Hanna, & J. Hunter (Eds.) *Artificial Intelligence in Medicine*, vol. 4594 of *Lecture Notes in Computer Science*, (pp. 305–315). Springer Berlin / Heidelberg.

Williams, M. V. (2008). The cancer reform strategy. *Clinical Oncology*, *20*(4), 271–274.

Williams, M. V., Drinkwater, K. J., Jones, A., O'Sullivan, B., & Tait, D. (2008). Waiting times for systemic cancer therapy in the United Kingdom in 2006. *British Journal of Cancer*, *99*, 695–703.

Yang, K. K., Lau, M. L., & Quek, S. A. (1998). A new appointment rule for a single-server, multiple-customer service system. *Naval Research Logistics (NRL)*, *45*(3), 313–326.

Yu, P. (1989). *Optimization*, vol. 1 of *Handbooks In Operations Research And Management Science*, chap. Multiple Criteria Decision Making: Five Basic Concepts, (pp. 663–699). New York, NY, USA: Elsevier North-Holland, Inc.

Yu, V. (2007). *Approximate Dynamic Programming for Blood Inventory Management*. Undergraduate senior thesis, Department of Operations Research and Financial Engineering, Princeton University.

Yuan, B., & Gallagher, M. (2007). *Parameter Setting in Evolutionary Algorithms*, vol. 54 of *Studies in Computational Intelligence*, chap. Combining Meta-EAs and Racing for Difficult EA Parameter Tuning Tasks, (pp. 121–142). Berlin / Heidelberg: Springer.

Appendix A

Data Collection Form

This appendix presents the form prepared to collect data from the hospital, shown in Figure A.1.

The name and radiotherapy number of the patient were included in the form so that the hospital staff could easily identify each patient as their form was completed. These fields were manually removed from each form before it was given to us in order to protect the patients' identity.

In addition to the attributes necessary for treatment scheduling, several attributes needed for pre-treatment are also present in the form. Although the work presented in this thesis focuses only in the treatment stage, the project which sponsored this work also encompasses the pre-treatment.

In a meeting with the hospital staff, we distributed the collection forms and explained what was expected in each field and the purpose of the data collection. An explanation of each field was also printed on the back of each form. This is presented in Figure A.2.

Radiotherapy Patient – Data Collection Form

Patient's Name

Radiotherapy Number

✂ ✂

Target patient?
❑ Yes ❑ No

Target date
/ /

Site

First oncology consultation
/ /

Initials of assigned doctor

Patient pre-treatment bookings dates (dd/mm/yy)

Mould room
/ /

CT scanner
/ /

Doctor (planning)
/ /

Physics (Isodose plan)
/ /

Simulator (Treatment verification)
/ /

Doctor (Approve verification and prescribe dose)
/ /

Physics (Plan check)
/ /

Oncentra (Enter + calculation)
/ /

Oncentra (Check)
/ /

Required radiation types
❑ High energy photon (10MV)
❑ Low energy photon (6MV)
❑ Electron

Decision to treat (dd/mm/yy)
/ /

Booking request (dd/mm/yy)
/ /

Earliest start date (dd/mm/yy)
/ /

Good practice start (dd/mm/yy)
/ /

Intent
❑ Radical
❑ Palliative

Number of fractions

Treated on weekends
❑ Yes ❑ No

Priority
❑ A (24h)
❑ B
❑ C

Duration of sessions

Sessions before the weekend

Schedule created on (dd/mm/yy)
/ /

First fraction date (dd/mm/yy)
/ /

Figure A.1: The form used in the data collection process

Definitions

First oncology consultation date: date on which the patients have their first examination after they have been admitted.

Target patient: If the patient must comply with the 31/62 days target.

Target date: 31/62 days target date.

Patient pre-treatment booking dates: dates on which the pre-treatment bookings take place. Boxes show all possible examinations. For sites with fewer operations, please fill in only the appropriate boxes.

Booking request date: when a clinical oncologist and a patient have agreed to a plan of treatment including a course of radiotherapy, the clinical oncologist should complete a booking request; this date is the date a booking form request is completed or an electronic request is logged.

Decision to treat: the date of the consultation in which the patient and clinician agree the treatment plan for first treatment. It may be the same as the date of the booking request, but only if this was completed on the same day.

Earliest start date: this date should only differ from “Booking request date” if there was an elective delay.

Good practice start date: limit date by which is good practice for the patient to start the treatment.

Duration of sessions: for how long each treatment session keeps the linac busy.

Sessions before the weekend: how many fractions the patient must receive before a weekend.

Schedule created on: date when the treatment is scheduled on Oncentra.

List of consultants and their initials.

Consultant		SPR	
1	SAM	A	NR
2	SS	B	EL
3	KF	C	IS
4	MG	D	DP
5	EMB	E	AC
6	JAC	F	PD
7	DALM	G	LM
8	PL		
9	SC		
10	MS		

Figure A.2: The explanation of the fields in the form for data collection

Appendix B

User Manual for the Scheduling Software

This chapter describes the prototype software that has been developed for the hospital. It also serves as instruction manual for anyone interested in using the software. Although the screen-shots presented are taken from the Linux version of the software, the software is also available for Windows (the operating system used by the hospital).

The prototype system starts with an empty schedule, shown in Figure B.1. First, it is necessary to populate the current schedule by loading it from a file. This is done by selecting *File* on the menu, and then clicking on *Populate* as presented in Figure B.2.

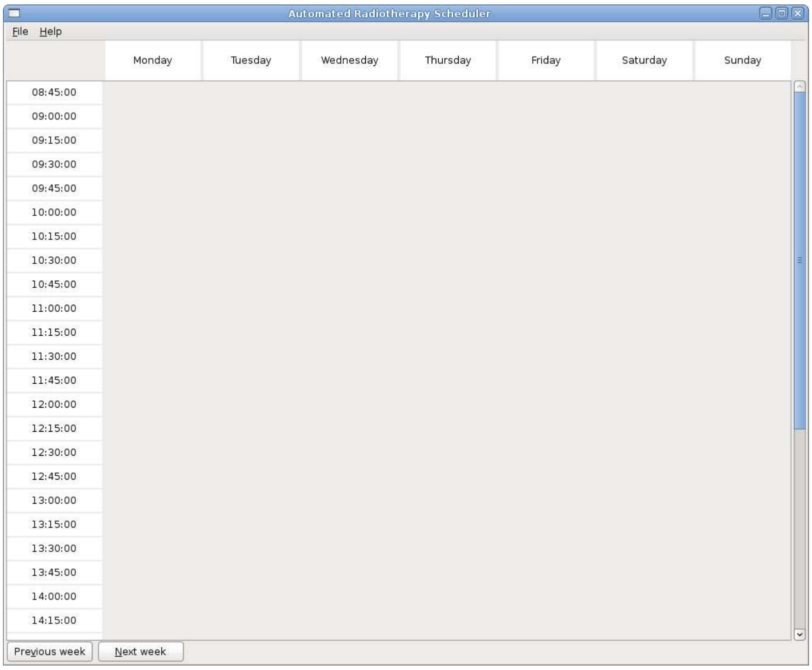


Figure B.1: Initial system state

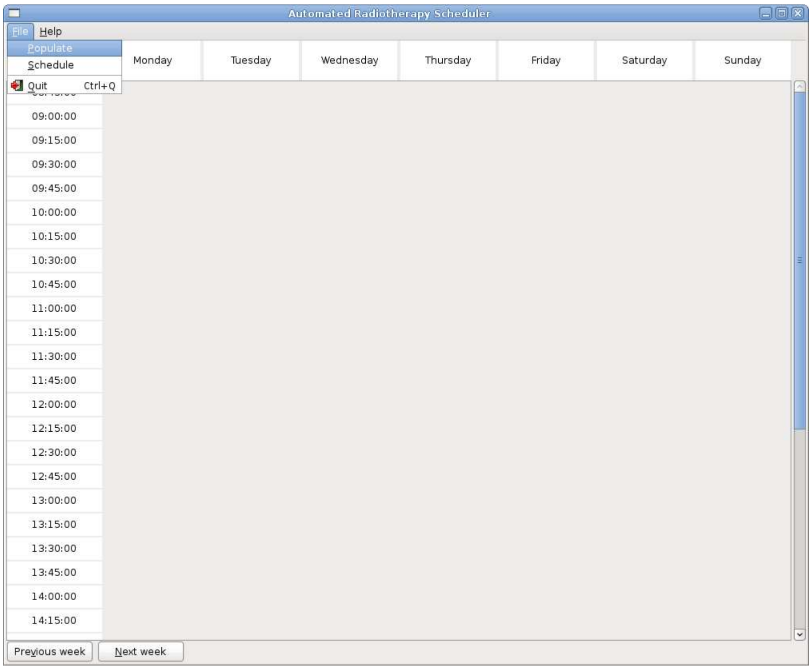


Figure B.2: Menu to populate the schedule

This brings the *Please choose a file* dialogue, as presented in Figure B.3, which enables the user to choose the file containing a schedule. The file should be a simple text file with the following format:

- One line with the number of linacs in the system, e.g. 4.
- One line for each linac with the attributes of each linac separated by white spaces:
  - a linac ID (positive integer),
  - opening and closing times on weekdays (hh:mm:ss),
  - opening and closing times on weekends (hh:mm:ss),
  - types of radiation emitted by the linac separated by commas and no spaces (high, low and/or electron),
  - a name for the linac with no spaces,
    e.g. `2 08:45:00 18:00:00 09:00:00 13:00:00 high,low,electron High1`.
- One line with the number of patients currently scheduled, e.g. 2267.
- One line for each patient with the patient's attributes separated by white spaces:
  - waiting list status,
  - treatment intent,
  - types of radiation required separated by commas and no spaces (high, low and/or electron),
  - relative weight of the patient (for the objective functions),
  - number of session days per week,
  - number of sessions per day,
  - possible days of the week when the patient can start treatment separated by commas and no spaces (Monday, . . . , Sunday),
  - 1 if the patient is treated on weekends, 0 otherwise,

- number of times the patient has been rescheduled,
- date when the decision to treat was made (yyyy-mm-dd),
- release date of the patient (yyyy-mm-dd),
- good practice target date according to the JCCO (yyyy-mm-dd),
- maximum acceptable target date according to the JCCO (yyyy-mm-dd),
- breach date (yyyy-mm-dd),
- number of sessions,
- one line for each session with the following attributes:
  * the duration of the session in minutes,
  * day the session is scheduled to (yyyy-mm-dd),
  * time the session is scheduled to (hh:mm:ss),
  * the ID of the linac the session is scheduled on.

e.g. `routine palliative high,low 1 5 1 Monday,Wednesday 0 0`

```
2003-01-02 2003-01-03 2003-01-03 2003-01-04 2003-02-02 2
15 2003-01-08 08:45:00 2
15 2003-01-09 08:45:00 2.
```

The result is the populated schedule shown in Figure B.4.

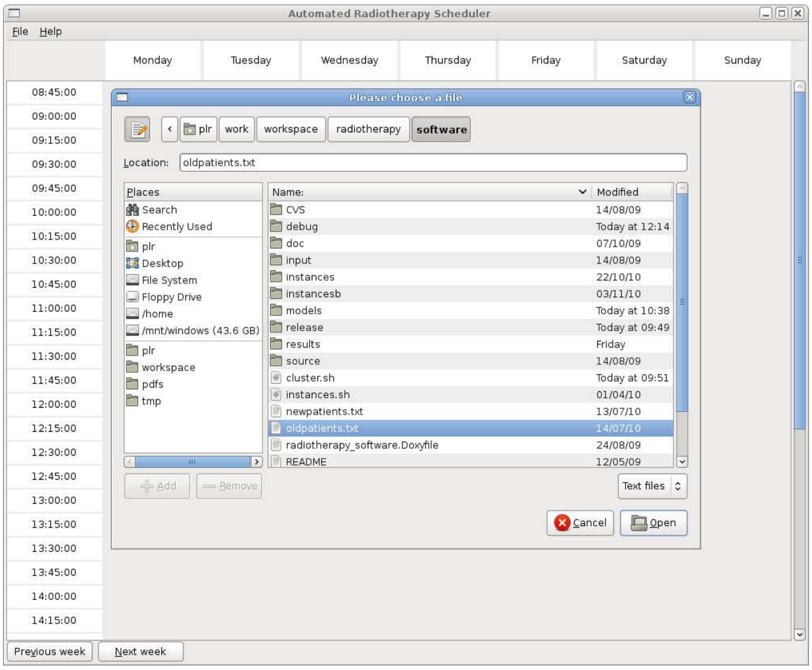


Figure B.3: Choosing a file which will be used to populate the schedule

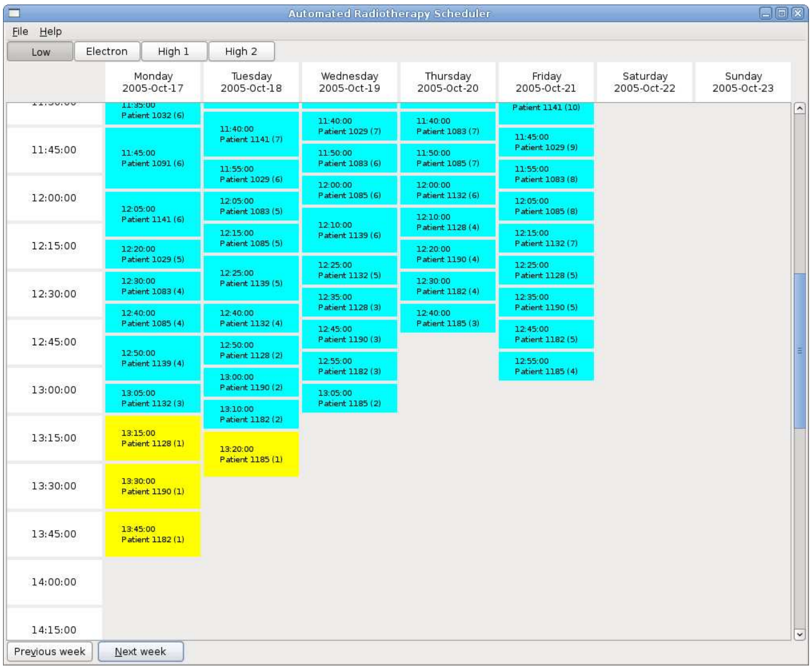


Figure B.4: Populated schedule

A schedule is shown for a specific machine and week at a time. To visualise a different week, the user should click on the *Previous week* or *Next week* buttons on the bottom left of the main screen. It is also possible to show a different machine by clicking on its respective button of the top left of the main screen.

Each slot shows the start time of the session for that slot, the name of the scheduled patient, and the order number of the session in brackets. In the examples, the name of each patient is simply *Patient N*, where N is an unique number assigned to each patient. The session slots are also colour coded in the following way:

Red: sessions of emergency patients.

Yellow: first session of each patient.

Cyan: remaining sessions.

If desired, this colour scheme can be easily changed and new colours and definitions can be added.

The process of finding a schedule for new patients is similar. First, the user should click on *File*, and then on *Schedule* as presented in Figure B.5. This brings another *Please choose a file* window shown in Figure B.6, where the user should enter the file containing the new patients who should be scheduled. The format for this file is nearly same as the file used to populate the schedule. The difference is that it consists only of patient information (no linacs) and the dates of unscheduled sessions should be “0000-00-00”.

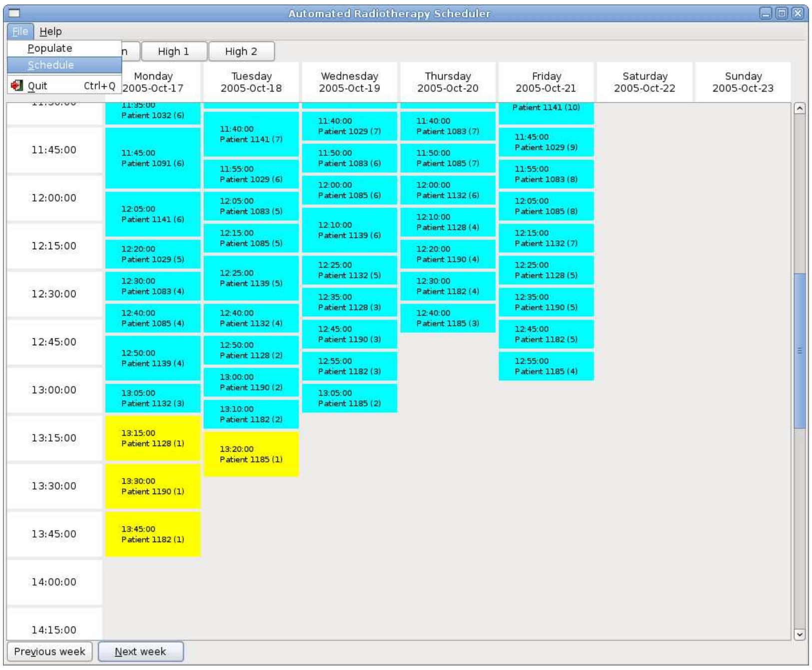


Figure B.5: Menu for scheduling new patients

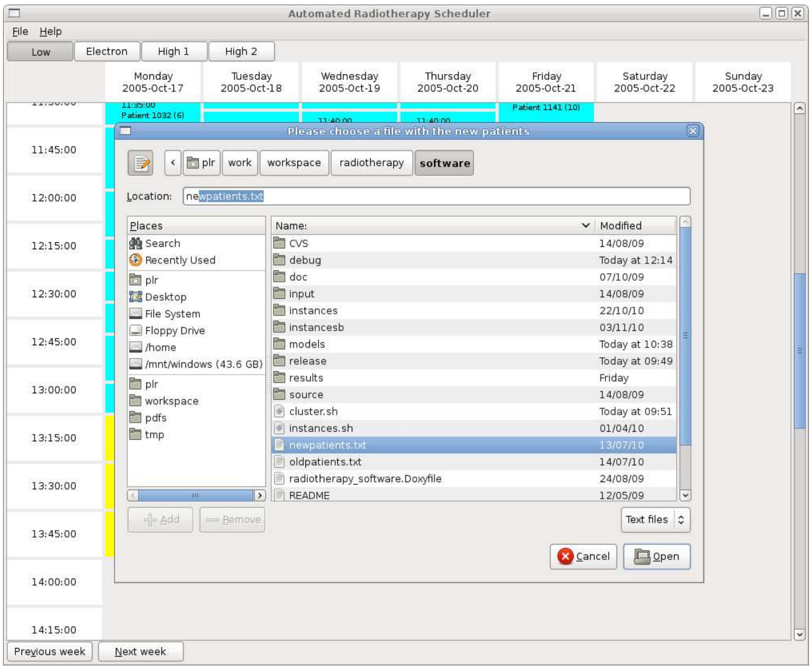


Figure B.6: Choosing the file with the new patient data

After the user has chosen a file, the *Create schedule* window opens displaying a list of the patients present in that file, seen in Figure B.7. The list shows the name, waiting list status, treatment intent, required radiation type(s) and total number of sessions for each patient.

Clicking on *Forward* takes the user to the configuration screen shown in Figure B.8, where option values of the scheduling algorithms can be changed from the default. Currently, it is possible to configure the maximum number of days in advance (MNDA values), size of the pre-scheduling horizon, size of the rescheduling horizon, number of times the patients of each waiting list status can be rescheduled, size and number of scenarios in the rolling horizon.

Clicking on *Create schedule* generates possible schedules for the patients in the file, and displays a list of the schedules generated seen in Figure B.9. For each schedule, it is possible to see the values of the objective functions analysed and its TOPSIS score.

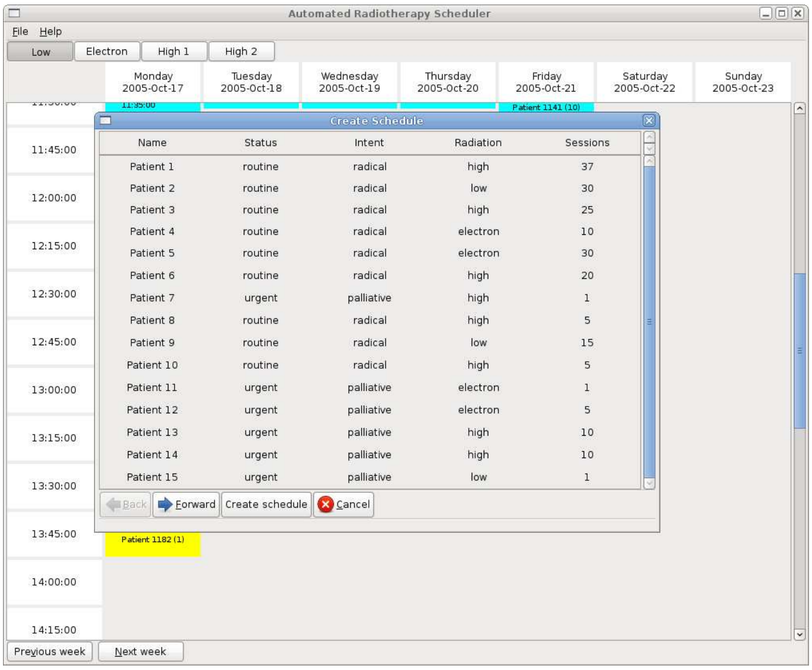


Figure B.7: List of new patients to be scheduled

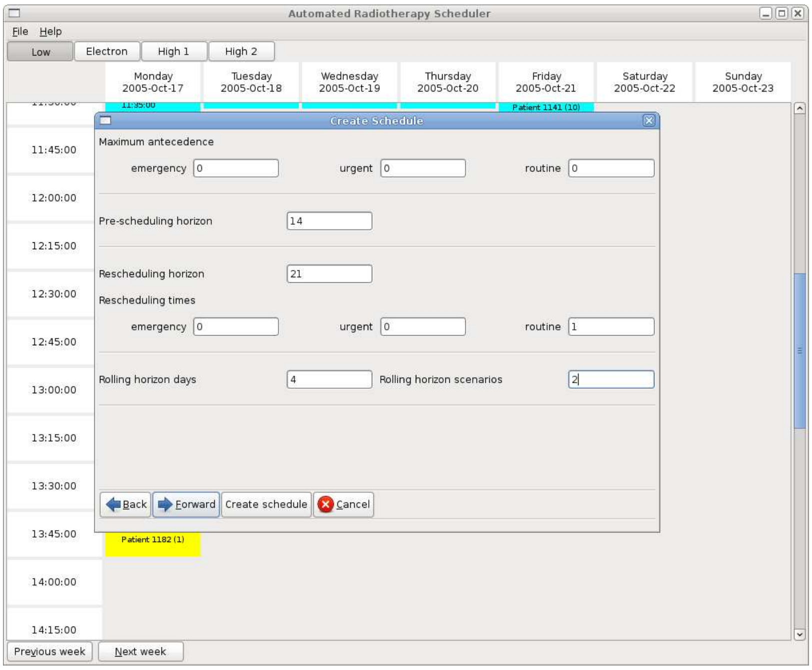


Figure B.8: Configuration of the scheduling algorithm

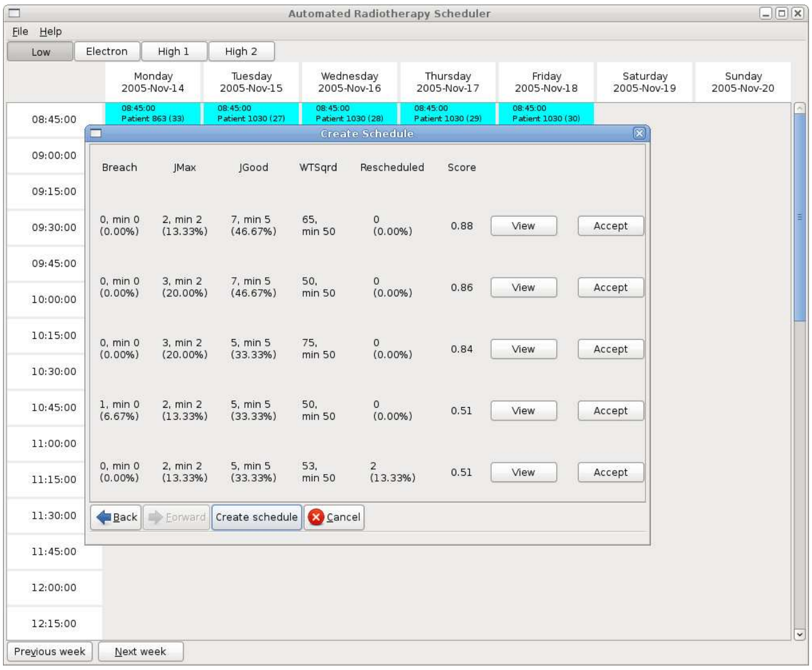


Figure B.9: List of created schedules

It is possible to view details of each schedule by clicking on *View*. A new window opens showing the list of patients who have been scheduled, including which dates they have missed in the last column to the right, as can be seen in Figure B.10. The breach date is represented by a *B*, the JCCO maximum acceptable is represented by a *J*, and the JCCO good practice is represented by a *G*. A green letter indicates that the due date is met, and a red letter indicates that the due date is missed.

By clicking on *Schedule* on the top left of this window, the user is able to view the created schedule, as shown in Figure B.11. Once the user has chosen the schedule of their preference, they can click on the *Accept* button next to the chosen scheduled. The result is the complete schedule written in the booking system, as can be seen in Figure B.12.

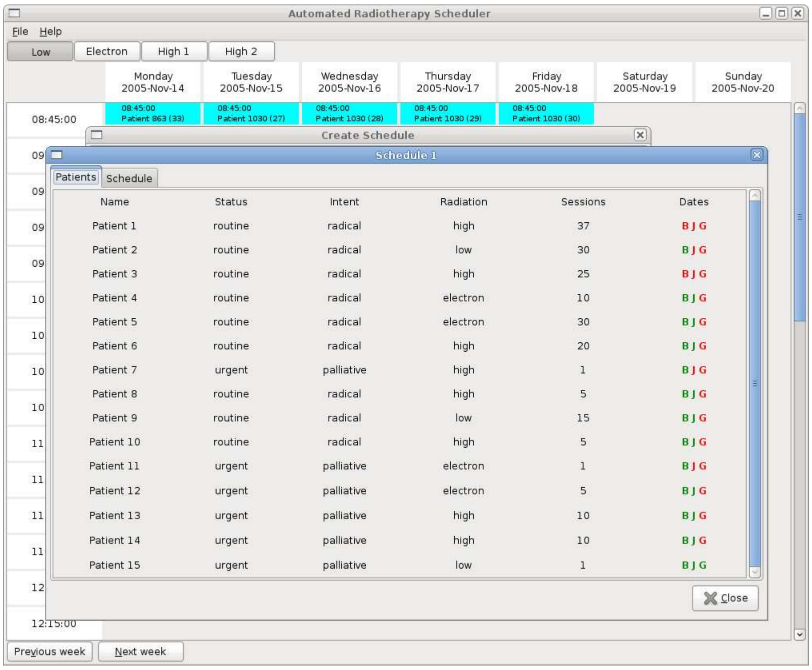


Figure B.10: List of patients with the due dates they meet and miss in a new schedule

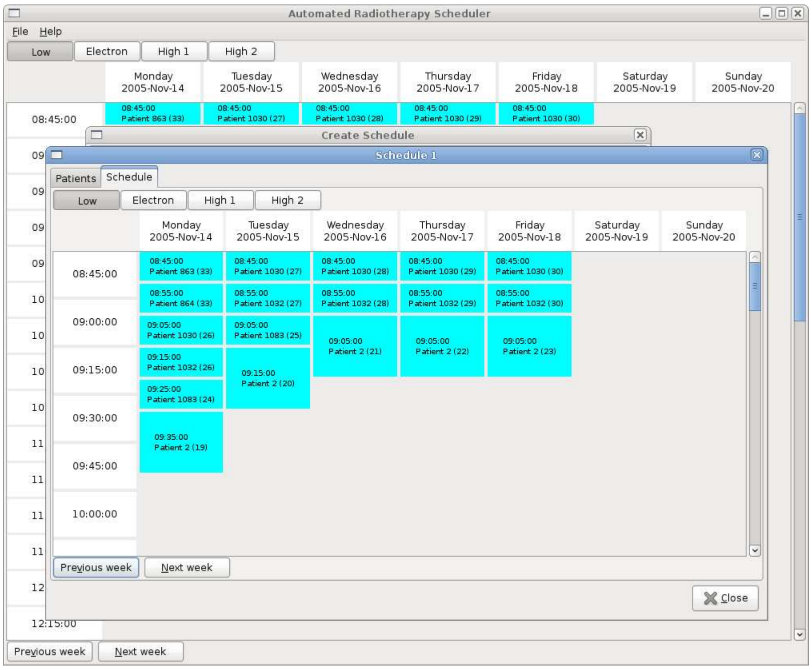


Figure B.11: Visualisation of the new schedule before it is accepted

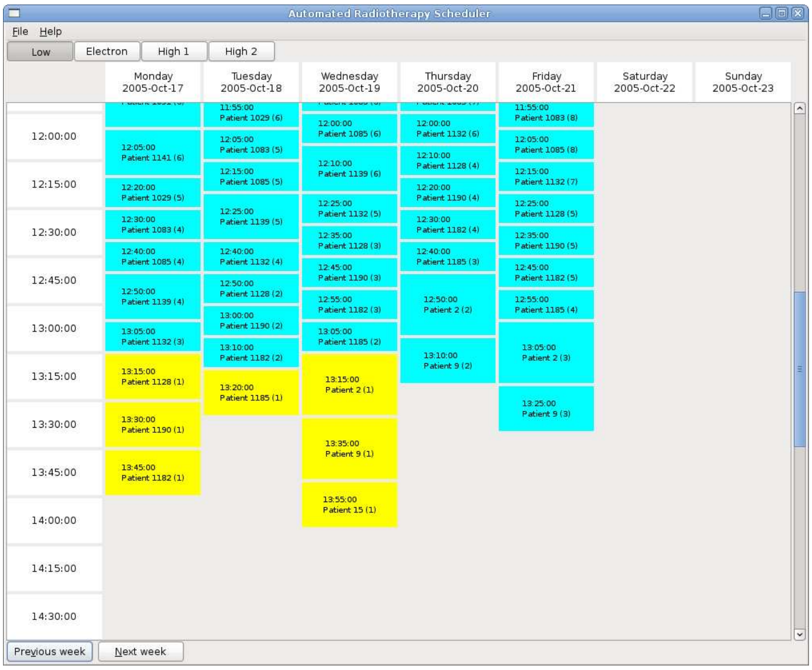


Figure B.12: Visualisation of the new accepted schedule

Appendix C

Letter of Acknowledgement

This appendix presents the letter of acknowledgement given to us by the hospital staff. As previously stated, this tool would greatly facilitate their work, and they would be happy to start using the scheduling system once the pre-treatment module has also been included.

Nottingham University Hospitals NHS
NHS Trust

City Hospital Campus
Radiotherapy Department
Hucknall Road
Nottingham
NG5 1PB

Tel: 0115 969 1169 ext 57264
Fax: 0115 9627994
Email: russell.hart@nuh.nhs.uk
Minicom: 0115 962 7749
www.nuh.nhs.uk

Please ask for: Russell Hart

Ref:

15th June 2010

Dear Sir / Madam

Re: Automated scheduling project.

The staff in the Radiotherapy Department have been involved in providing the retrospective scheduling data for the University of Nottingham Automated Scheduling, optimisation and Planning (ASAP) Research Group. As a result the ASAP group have undertaken various experiments in order to develop an automated scheduling model to maximise the treatment capacity within a tightly resourced service.

The radiotherapy department will be very pleased to utilise such an automated approach to scheduling. This scheduler will assist the radiographers and administrative staff to book patient appointments for both the pre-treatment and treatment phase of the radiotherapy pathway. The objective being to pre-book a series of appointments with predictable duration and specific time intervals between appointments.

Yours sincerely

Russell Hart
Radiotherapy Services Manager