VRIJE UNIVERSITEIT

A Careful Solution

patient scheduling in health care

ISBN: 978-90-9027274-0

Cover design by Joeri Lambert

Printed by Universal Press, The Netherlands

VRIJE UNIVERSITEIT

A Careful Solution
patient scheduling in health care

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad Doctor aan
de Vrije Universiteit Amsterdam,
op gezag van de rector magnificus
prof.dr. L.M. Bouter,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
van de Faculteit der Exacte Wetenschappen
op 13 februari 2013 om 13.45 uur
in de aula van de universiteit,
De Boelelaan 1105

door

Pauline Maria Koeleman-Out

geboren te Beverwijk

promotor: prof. dr. G. M. Koole

Acknowledgements

This thesis is the product of a number of years of work. Years with some ups and downs, some nice trips to conferences, and full of great experiences. During this period many people have been very important in helping me to get to this point. I would like to name them here.

First, everyone at CC Zorgadviseurs, and especially Marcel de Jong, for providing me with the opportunity to do research. Working with him was a lot of fun and a great learning experience. And thank you Joeri Lambert for designing the cover of this thesis.

I want to thank Ger Koole for his support and enthusiasm. He never stopped encouraging me to move forward, and always believed I could do it.

It was wonderful to work with René Bekker and Sandjai Bhulai. I learned a lot from Sandjai and he was a great inspiration with his enthusiasm and positivity. I want to thank René for the discussions we had, and for letting me force him to have lunch on so many Fridays.

I also want to thank all (former) members of the OBP research group for the great atmosphere and the fun discussions during lunches and on other occasions.

My family have always supported me, even though they didn't always understand what I was doing. But most of all I want to thank my wonderful husband Bertram. You were a great help, with improving my English and suggesting the title of this thesis, but mostly by just being there to encourage me. It was great finishing our books together. I love you.

Paulien Koeleman
Beverwijk, november 2012

Table of Contents

1 Introduction

This thesis, as the title already suggests, considers the topic of patient scheduling in different settings within health care. It is part of the larger field of operations research applied to health care, which is just one of the application areas of operations research and management science. This application area has been growing rapidly since its inception in the 1950s, and there is no sign that its growth is slowing down. Within health care many different problems are being addressed using operations research methods and probably many more that are not but could and should be studied in a quantitative way. The large variation of problems leads to quite an array of methods employed. In this thesis we study a number of different problems that involve the scheduling of patients.

In this introduction we place the research this thesis contains in the larger perspective of the field of health care OR, the historical developments in this field and in health care itself. We start with a short overview of the development of health care OR and the main themes and issues that have been studied and some of the challenges and open issues for health care OR that we feel have not yet received adequate attention and that are important issues for the near future. Then we give an overview of the patient scheduling problems that occur in health care and the OR methods available to attack these problems. We end this introduction by giving an outline of the remaining chapters in this thesis and describing how they fit within the bigger picture.

1.1 OR and health care: a short overview

The field of operations research as applied to health care has existed just about as long as operations research itself. Concerning health care, operations research is generally applied in two fields. One of these concerns the actual care and treatments. An example is optimising the settings for radiotherapy treatment of cancer cells. Work like this is often a collaboration of mathematicians with physicians or medical researchers. The second application area addresses the organisation of care and processes within and between institutions. This thesis falls into the second field of applications within health care.

To the best of my knowledge the first paper that addresses the field as such is by Norman Bailey [**10**]. This paper describes the main themes and findings of a conference held in March of 1950. The author defines operations research in medicine as "concerned with the organisation of existing clinical techniques and facilities so as to make them more widely available to patients in need; with the replanning of wards, clinics and practices so as to employ present resources to the greatest advantage; and with the general administration and planning of medical services. All the disciplines of scientific research in general and medical research in particular can be made to subserve these ends." The research efforts described here are largely concerned with collecting data for specific objectives, such as collecting data on time spent on different activities by nurses in order to give a good description of the actual job of a nurse. Design and dimensioning hospitals or departments is another important subject that Bailey mentions. He stresses the importance of a multidisciplinary approach to these issues.

From that time onward the number of papers and studies published in this field has been growing at an increasing rate, and there is no reason to expect this to change any time soon. The most recent comprehensive literature review by Brailsford et al. [**24**] on simulation and modelling in health care makes it clear that it has become impossible for a department, let alone a single researcher, to stay current with the literature. This makes good literature review papers all the more desirable. Sadly, there are not that many that address the field as a whole. The general reviews we have been able to find are not very recent, we have found one by Boldy and O'Kane [**21**] from 1982, and an older one by Fries [**44**] from 1976 with an update three years later [**45**].

How can the fast increase in attention for the field of health care OR be explained? Probably at least part of it is due to the rising costs of health care, both in absolute monetary value as in percentage of the gross domestic product, in almost every developed country. This is due to the ageing population, thanks to newly developed treatment methods. More and more problems and diseases can now be treated. This increases the costs directly, but also lengthens the patients' lives which in turn leads to more health care demands.

Another part may be due to the shortage of nursing and other personnel caused by the decreasing percentage of employed people in relation

to the whole population. This means that the demand for care increases more than the personnel available, regardless even of the costs of extra staff. Some of this could be relieved by measures to make working in health care more attractive, so as to get more young people into the profession, but the fact remains that more and more work will have to be done per employee.

A third possible factor explaining the increasing interest in health care OR could be technological advancements, which are changing the way patient care is handled. Exciting examples are available from operations being done long distance, with cameras and other equipment guiding the doctor to new treatments with for instance stem cells. But some more mundane developments can have even more impact on day to day events, like patients with diabetes being able to do their own checks for blood sugar at home and sending them to the doctor by phone or some web interface instead of having to go to the hospital for checks.

These and other factors are constantly changing the processes and demand for different types of care, and institutions and employees need to adapt to those changes. The increasing costs and personnel shortages are making it more and more necessary to do so in a way that uses available resources effectively and efficiently.

There are a few topics that have received a lot of attention in the health care OR literature. The most notable are the scheduling of treatments in the operating theatre and scheduling for diagnostic facilities such as MRI scanners. At probable reason for the popularity of surgery scheduling in the literature is the fact that the operating theatre, with all its special equipment and highly trained personnel, is the most cost-intensive department of a hospital, and it is used by roughly half of the patient who stay in a hospital. This makes running an efficient operating theatre crucial to a smooth and cost-effective organisation of a hospital. The attraction to researchers may also come from the many uncertain factors and variables, which makes surgery scheduling an interesting problem. Three recent overviews of the literature concerning surgery scheduling and operating theatre management are May et al. [**81**], Guerriero and Guido [**55**] and Cardoen et al. [**27**].

Appointment scheduling is also a widely studied topic, starting with one of the first papers in health care OR by Welch and Bailey [**106**] where

the well-known Bailey-Welch rule was first introduced. This problem is further discussed in Chapter 2.

Another field that has received a lot of attention is the admission process of patients to the hospital, or the use and dimensioning of hospital wards. Many of these are either simulation studies or applications of queueing theory.

Nurse scheduling is also a popular topic, particularly in the deterministic setting using combinatorial techniques. This is a challenging field, because of the large number of variables and constraints. This problem differs from those mentioned above in that most of the models used are deterministic in nature. An overview of the literature on this problem is Cheang et al. [**29**].

A few notable points emerge. The first of these is the apparent lack of actual application or implementation of research findings in health care organisations. This lack has been noted and lamented from the beginning of the research in this field, and is still noted in the latest review by Brailsford et al. One explanation given is the fact that researchers are interested in complex, new and challenging models and techniques and that this is also where the research funding goes. As a result, certain techniques are given a lot of attention. These more technical studies most often are not directly applicable, but the interest of many researchers and providers of funds appears to end here. Probably a lot of implementation is being done by consulting firms or people working in health organisations themselves, but these are based on simple models and techniques and are mostly not written up for publication in scientific journals. Many can be found in the so-called "grey" literature, that may well be even larger than the actual scientific literature.

Another point to note is that almost all studies use either data analysis and statistical methods or simulation as their main method. Of course the spectrum of techniques available is much larger and many are also used, but they come nowhere near the popularity of statistical analysis and simulation. A possible reason is that these methods can be applied in many situations where other methods would be difficult to use because of the size of the problem or assumptions that are necessary to use a method but are not met in reality. Another reason could be that there are many software tools available for data analysis and simulation, and less for other methods. The upside of simulation as a method is the flexibility to incor-

porate all kinds of details and the relative ease of explaining the model to health professionals. On the other hand, analytical models may well be more generic and require less data to give results. Also, usually all details of an analytical model are made clear in papers, which is often not the case with simulation models.

Finally, it is clear from the reviews that the models and analyses are disproportionately directed at hospital care and much less at for example care-at-home and nursing home. A possible reason for this is the fact that processes at hospitals are often more complex than in other kinds of health care institutions. Also hospitals are relatively expensive to operate and have larger budgets, and so they have the funding for research and there may be more gains to be made. The lack of hard data in many health care settings probably plays a role as well. Nevertheless other kinds of care have received limited attention and this is a gap to be addressed by the operations research community.

Despite all the interest in health care OR over more than sixty years, there are still some areas and issues that appear not to have received all the attention needed. First, as was already mentioned, the large majority of the literature concerns hospital care. It seems clear that other types of care will be more and more in demand in the coming years with the ageing population and the increase in the number of chronic health issues like diabetes. This is especially the case for nursing homes and care-at-home.

The care in those institutions has different characteristics from those in hospitals. For one, the length of time that patients stay within the system can be very long, sometimes years as opposed to hours or days in the case of hospitals. This leads to less turnover in the system. Another large difference is the fact that care in nursing homes and home care is often repeated periodically, and does not consist of one-time events like surgeries. Decisions made now can have an impact every day or week for the next few years, so it is even more important to consider the longer-term effects when for example prioritising patients for admission or making staffing decisions.

These characteristics require different models to address problems like the scheduling of activities or visits of employees to the patient, designing good schedules for nurses and other staff and prioritising the admission of patient groups to the facilities. Some of these problems are related to other issues in OR. For example, the problem of scheduling and routing visits to

patients by home-care nurses has been addressed using techniques from the vehicle routing problem by for example Eveborn et al. [**40**]. But a lot of these issues have not been sufficiently addressed, or even not at all as in the case of admission prioritisation.

Another issue that has not received much attention in the literature is the question of how to handle multi-step processes. Many of the questions discussed above consider one department or one resource. Optimising one part without relating it to other parts of the complete health care system that the patients experience will almost certainly lead to suboptimal use of resources. However, most times the problems are complicated enough as they are, and considering multiple steps is even harder. Another factor that explains this lack of attention in the literature may well be the fact that when it comes to managing steps between organisations or departments interests are often conflicting. This makes it difficult to formulate a common goal, as many modelling techniques require, and leads to issues when implementing changes to the system as a whole.

An example of a multi-step process are those of patients who first need emergency care in a hospital after a brain seizure, and afterwards have to rehabilitate for a few months in a separate facility. The coordination of the transfers of these patients is not always as efficient as it should be, because rehabilitation facilities treat many different types of patients that arrive through various routes, and they do not have a (financial) incentive to prioritise hospital patients over other patients. Also, rehabilitation is in itself a multi-step process as patients receive care from various disciplines during their stay.

Multi-step processes also occur within organisations. A good example is patients who stay at an intensive care after a surgery or an emergency admission, and when they recover can be moved to a normal ward. Even processes like these are challenging to analyse and optimise. But the growing complexity of care and need for efficiency ask for good ways to handle multi-step processes. An overview of the research done on processes within one institution can be found in Vanberkel et al. [**101**].

Another challenge to the research community is the development of models and solutions that strike a balance between being detailed enough to be of practical value and generic enough to be applicable in different organisations. Often studies reported are case studies where data is analysed and then a model is developed that best describes what happens in this particular case. Due to the complexity of the care process in a particu-

lar case, this is often a simulation model. This model is then analysed and provides solutions or improvement steps. While this can be very valuable for the organisation involved and maybe provide some insights for other cases, the results cannot be easily used by other organisations to study and solve their problems.

On the other hand there are studies in the literature that consider models that are interesting from a theoretical perspective, but need assumptions that are not met in practical situations. Examples are models that require the duration of a hospital stay to be exponentially distributed, or that take the length of a surgery as deterministic.

There is a need for models that hold a middle ground between these two extremes. This may require closer collaboration between people from health care and OR researchers.

A final challenge is related to this point. Since the origin of health care OR, the lack of implementation of research results has been a point of concern. This is still largely the case today. Both the scientific community and the health care managers and professionals should work to change it. This could be beneficial to everyone, including the patients. Models which are sufficiently detailed but still generic can probably help to achieve this.

Patient scheduling problems appear in many different settings and types of health care. Some of these have received a lot of attention already, while some are essentially new to the field. In all cases however, we feel there is room for improvement to make the models more realistic. This will hopefully make the results easier to implement so that the research can actually achieve its desired effect of improving health care processes. In the next section we introduce some of the most important settings where patient scheduling plays a role.

1.2 Patient scheduling problems

Within the field of health care OR, patient scheduling problems are situated at the operational level, where day-to-day decisions are made. There are different scheduling problems in each type of care institution or facility, and also some problems that are common to all situations. In this section we give an overview of the most important patient scheduling problems in different sectors of health care.

Surgery scheduling
This area addresses the question of how to schedule surgical procedures into the available time in one or more operating theatres. In most cases there is a weekly repeating block schedule that states which specialism or specialist is assigned the operating theatre with its staff at which days. Then the surgeries belonging within each block need to be assigned a date and time. Complicating factors are the stochastic duration of the surgeries, the need to keep time available for emergency surgeries, and of course the fact that before and after the surgery the patients require other resources such as ward beds. The main (quantitative) criteria by which to judge a schedule are the efficient usage of the time available, the overtime and the fraction of scheduled surgeries that need to be cancelled or rescheduled, and the waiting time for emergency surgeries.

Appointment scheduling
This problem occurs in a few different settings both in health care and in other areas. In health care, good examples are the offices of general practitioners and dentists, and outpatient clinics. In appointment scheduling the goal is to schedule a certain number of appointments into a finite block of time. Important performance criteria here are the waiting times of patients, the overtime at the end of the day and efficient use of the doctor's time. There can be more complicating factors than just the appointment duration, such as emergency arrivals, patients who don't show up for their appointment, or patients who are late.

There are, of course, similarities with the surgery scheduling problem, since in both cases activities with an uncertain duration have to be scheduled into a block of time of fixed length and overtime should be minimised. The main difference lies in the fact that in surgery scheduling the waiting time usually is of little importance, since the patients are in the hospital anyway, and overtime is much more costly for an operating theatre than for most appointment scheduling settings.

Elective admission scheduling
Admission scheduling is the question of how to decide which patient should be scheduled for admission at which point in time. This is a complex problem, because of the many sources of uncertainty in this situation and the many different patient requirements such as maximum time to admission. Uncertainties are for example the length of stay of a patient,

emergency admissions that use the same wards, and often also the availability of nurses, or staffed beds. To add to the complexity many patients need more resources than only a ward bed, for example some time in the operating room or intensive care, or complex medical treatments for cancer that have to be prepared right before administering the treatment. So in scheduling admissions, these other resources have to be taken into account. Important criteria to judge the quality of the admission schedule are the occupancy of the beds, the number of emergency admissions that have to be blocked and the number of admissions that have to be rescheduled.

Priority scheduling
This problem occurs when there is a limited number of servers i.e. doctors or beds, available, and patients of different types are waiting to be admitted once a server becomes available. The question is how to decide which patient to allocate the free server to, based on the number and the types of patients waiting. This is different from admission scheduling in that the admissions are not scheduled in advance, but are decided one by one each time a patient leaves the system. Examples are emergency departments, nursing homes and rehabilitation facilities. The goal here is to balance the interests of various different types of patients, who can each differ in their level of urgency. In many cases it is possible to keep a server idle even when there are patients waiting, in order to serve really urgent patients quickly when they arrive. Important performance measures are the occupancy of the servers, the mean waiting times for each patient type, or the fraction of patients that wait longer than some threshold value.

Home care visits and routing
In the setting of home care the scheduling of patient visits is combined with a routing problem for the employees. The setting is usually that of several nurses and other employees with different skill levels, that need to visit patients at home to carry out one or more tasks. These tasks can range from cleaning help to administering cancer treatments, and the skill level determines which employee can perform which tasks. For each task a certain time window is available in which the visit should take place. The employees are given a schedule or route with an order and timeline of the patients they will visit during their shift. Usually there are several factors to be taken into account, like the number of different employees

each patient sees, or the fact that employees with allergies cannot visit patients who have a cat or dog. This problem is related to the well-known problem of vehicle routing, and can be addressed using similar methods.

Chapter 2 discusses the problem of appointment scheduling. In Chapter 3 the problem of admission scheduling is addressed. Both Chapter 4 and Chapter 5 deal with the problem of how to prioritise patients for admission in different settings.

1.3 Available methods

In this section we take a consider main methods and techniques available for solving patient scheduling problems. The diversity of problems and their characteristics discussed in the previous section leads to a diversity in techniques used to solve them. All of these methods have their strengths and limitations, and we discuss these together with the most important applications of each method. From the literature review by Brailsford et al. **[24]** we see that the most popular methods are statistical analysis and modelling, followed by simulation. Mathematical modelling and optimisation methods are used in only a small part of the literature.

Queueing analysis
Queueing systems are systems where customers, in health care most often patients, arrive at a system and then receive some type of service from a server. In health care the server often represents a doctor or a bed. Well-known queueing models are the Erlang loss and delay models, both of which assume patients arrive randomly according to a Poisson process. Some queueing models can be analysed exactly, which means that there exist closed-form expressions for important performance measures like the mean waiting time, or the fraction of blocked customers. When this is not possible, approximations can sometimes be developed. This is for example the case when the arrival intensity varies, as is the case in emergency departments.

Examples where queueing analysis has been used in health care are deciding on the number of beds for a hospital ward and analysing the waiting times in an emergency department. For scheduling problems queueing models are sometimes used as an approximation for scheduled

arrivals, or they can be used to evaluate a given schedule, possibly in combination with some optimisation method. More examples of applications of queueing analysis can be found in Green [**53**].

Mathematical programming
Mathematical programming is the name for a set of techniques to find an optimal combination of multiple decision variables under a set of constraints. The form of the objective function often gives the name to the technique, like in linear or quadratic programming. The decision variables can be real or integer valued. Methods to solve mathematical programs are widely available and can solve programs with many decision variables, which makes it a very powerful set of techniques. The stochastic nature of many health care settings is hard to incorporate when using mathematical programming, which is often no issue for high level decision making. Alternatively, it can be implicitly considered in the goal function. A good example of an problem addressed using mathematical programming within health care is surgery scheduling, see for example Beliën and Demeulemeester [**16**] and Denton and Gupta [**35**].

Markov processes
Markov processes, especially Markov decision processes, are often used in situations where something is measured or decisions are taken each time something changes in the system. For example, when a new patient arrives or a patient leaves the system. A Markov process is described by a state space, an action space, a reward or cost function and transition probabilities. In the case of decision processes, the transition rates can be different for each action that can be chosen in a state. In reward processes a reward is earned per unit of time the system spends in some state. These processes can be used to track the number of customers waiting for service, or to decide which of several types of patients to admit when a server becomes available. Markov decision processes have been successfully applied to various diagnostic services, see for example Patrick et al. [**89**] and Day et al. [**34**].

Simulation
When a situation or problem is hard to fit into one of the usual models, or when the model becomes intractable due to a large number of variables, simulation remains open as a way to analyse systems. The power of simulation lies in the flexibility to model assumptions or setups that are hard

to fit into a queueing model. This makes it an excellent method to model complex situations, and gain insight in the performance. Optimisation can be done using simulation, but not using the same optimisation techniques used in deterministic settings.

In health care simulation is often used to model processes that have more than one step, for example an emergency department where first the patients are triaged, then wait, see a doctor, have some tests done, get treatment and are discharged or admitted to a ward. It has also been used for all kinds of scheduling problems, from appointment scheduling to designing a good staff schedule. A good overview of literature and applications can be found in Jacobson et al. [**66**].

Several of these methods are used in this thesis. In Chapter 2 we use local search for finding optimal schedules when all patients arrive on time, and simulation for evaluating schedules in the case with non-punctual patients. In Chapter 3 we use queueing analysis and quadratic programming for addressing the problem of elective admission scheduling. Then in Chapters 4 and 5 we use Markov decision theory to study prioritisation problems.

1.4 Outline of this thesis

In the remaining chapters in this thesis we treat the subject of patient scheduling in a few different health care settings. The one thing that all models in this thesis have in common, apart form the scheduling aspect, is that every problem has been encountered in a practical setting. So the models are not only interesting from a theoretical perspective, but also address a real problem. Even though simplifications have been made, we consider the models detailed enough to be of practical value while still being sufficiently generic to be used by different organisations. Of course actual implementation of these and many other models remains a challenge for organisations as well as for OR researchers.

Chapter 2 concerns one of the oldest problems in health care OR, that of scheduling outpatient appointments. This problem occurs in many settings within health care, and also in other areas of application. We extend the literature by the incorporation of emergency arrivals into the model. Also we address the issue of unpunctual patients, and how to anticipate

this in a good schedule. This chapter is based on the work in Koeleman and Koole [**72**] and [**73**].

In Chapter 3 the issue of scheduling elective admissions to hospital wards is considered. A medical specialty usually schedules its own admissions, or in some cases even individual physicians. This scheduling is often done without considering the occupancy of the wards, and without regard to other groups of patients that use the same wards. The main goal for scheduling of surgical specialties is often to use the available time in the operating theatre as efficiently as possible. These practices cause more variation in demand for beds than is necessary, and so has a negative influence on performance and utilisation of beds. This chapter presents a way to schedule admissions with the goal of aligning the demand for beds with the number actually available, and reducing the unnecessary variation, without deteriorating the utilisation of operating room capacity. This chapter is based on Bekker and Koeleman [**14**].

Chapter 4 treats the problem of how to prioritise patients in the setting of home care and a rehabilitation facility, when there are several types of patients with different service needs and waiting costs. All these patients require service or treatment from different specialties during some length of time. These are not one-time events, but they repeat, for example daily or weekly during a patient's rehabilitation process. In this chapter we address this problem using Markov decision theory, and develop approximation methods. This is necessary because the many variables lead to intractable problems in most practical situations; the so-called curse of dimensionality. This chapter is based on the research in Koeleman et al. [**71**], Koeleman and Bhulai [**70**] and Haensel et al. [**57**].

Chapter 5 deals with the question of how to prioritise in a setting with one type of server and several patient types, where not the mean waiting times are important but the fraction of patients that wait longer than some target length of time. This situation occurs for example in an emergency department, but is also applicable in situations outside of health care. Optimising such a metric does not work with standard models, so here a special type of state space description is used in Markov decision theory to handle the problem. This chapter is based on Bekker et al. [**12**].

Chapter 6 summarises the main points of the thesis, and contains a discussion on the findings, their applicability, practical implications and questions for future research.

Outpatient appointment scheduling

2

Outpatient appointment scheduling is a subject of great interest to hospitals and other medical institutions. Most doctors, dentists, general practitioners and diagnostic facilities use appointments. Outside of the medical world the problem occurs as well, for example in the scheduling of loading and unloading ships. No wonder that the problem has been an object of study for a long time, starting with the work of Welch and Bailey in the early fifties [**106**]. From that time on many papers have been written studying this appointment scheduling in many settings, and with many different assumptions and methods.

The goal of appointment scheduling is to balance the interests of the patients with those of the doctors. The patients want long intervals between appointments, as this minimises their waiting time. The doctors on the other hand wish to have as little idle time and overtime as possible, and therefore they prefer shorter intervals between two consecutive appointments.

If the durations of all appointments were known in advance with complete certainty, and there were no no-shows and other unexpected events, there would not be a problem at all. The challenge in appointment scheduling comes from the different factors of uncertainty. Indeed, the most obvious uncertain factor is the duration of the appointments. Generally something is known about the appointment durations, for example an average duration with standard deviation or some idea about the form of the probability distribution. Another factor of possible uncertainty is the unpunctuality of patients, or even no-shows where the patient does not show up at all. To have an effective schedule, this behaviour needs to be taken into consideration to avoid unnecessary doctor idle time. If there are also emergency arrivals, which have to be seen to as soon as possible, this complicates the situation further.

2.1 Literature

As already noted above, there is a long history of research on appointment scheduling. In this section we describe the main themes and results, and

explain how the approach in this chapter adds to the existing literature and results.

As noted above, the research on appointment scheduling started with the work of Welch and Bailey. Their most famous result is the so-called Bailey-Welch appointment rule, which states that two patients should be planned at the start of the day, and the other patients evenly spaced throughout the day, to offset the undesirable effects of no-shows, patient lateness and the doctor starting late. They made the assumption that all appointment durations were identically distributed and independent of each other.

A large part of the literature concerns simulation models for evaluating the performance of appointment schedules, e.g.. Fetter and Thompson [**42**] and Vissers and Wijngaard [**104**]. Ho and Lau [**63**] compare different rules for making appointments in different settings using simulation, and conclude that no single rule works best in all situations, though the Bailey-Welch rule works fine in many cases.

Other works consider finding and optimal schedule, which minimises some combination of the patient waiting time, doctor idle time and overtime. Stein and Côté [**100**] use an analytical method for finding the optimal schedule in a case with exponentially distributed service times and the restriction that the schedule be even-spaced. Wang [**105**] finds an optimal schedule without this restriction and with a Coxian distribution for the service times, and shows that this is an improvement over an even-spaced schedule. He gives the optimal schedule in terms of inter-arrival times in continuous form. For larger numbers of appointments he gives an approximation for the optimal inter-arrival times.

One of the few practical implementations is the one by Rising et al. [**94**]. They try to smooth the number of appointments over the week and over the day to accommodate the number of walk-in patients and emergencies, and they find a good schedule by trial and error in a simulation model. The implementation of their schedule gave good results in terms of waiting times and overtime of clinics.

An overview of the important issues to consider when designing an appointment system can be found in Gupta and Denton [**56**]. For a thorough review of the literature on appointment scheduling we refer to Cayirli and Veral [**28**]. They present the research done in the second half of the last century and mention some directions for future work.

The research presented in this chapter can be divided into two parts. First, in Section 2.2, we present a method to find the optimal appointment schedule in a situation with emergency arrivals and general service times. It has been found by O'Keefe [**87**] that the coefficient of variation of the service times in practice is considerably smaller than 1 (more in the order of 0.5), as in the exponential distribution used in many studies. The exact form of the distribution can differ, but Ho and Lau [**63**] find that only the coefficient of variation has a significant influence on the performance of the appointment schedule. According to Denton and Gupta [**35**] higher moments of the service time distribution are only important in the case were the costs of waiting time of patients are high relative to the cost of server idle time. Emergency arrivals or other disturbances such as phone calls are also known to have a considerable influence on the waiting times of patients as discussed by O'Keefe [**87**], which can also be seen from our experiments presented below.

The method we use here is a generalisation of the local search method used by Kaandorp and Koole [**69**], who study the case with only scheduled arrivals and exponential service times. Because we use the amount of work present in the system as a state description instead of the number of patients, the service times can have any positive distribution, and can differ for the scheduled and the emergency patients. Related to [**69**] is the work of Vanden Bosch et al. [**103**]. They solve the problem with Erlang distributed service times using a different method that is much faster. Neither of these papers include emergencies. Begen and Queyranne [**11**] include optimisation with emergencies, but only if they arrive during the service of scheduled patients. This can be a restriction if the service times of emergency patients are longer than that of scheduled patients. We do not make this assumption.

In the somewhat related area of surgery scheduling more work has been done on accounting for emergency arrivals. Examples are Gerchak et al. [**51**] and Lamiri et al. [**76**]. However, this work differs by assuming emergencies should be done on the same day instead of as soon as possible, as in the case of appointment scheduling.

Then in the second part, in Section 2.3, uses a different method to address the appointment scheduling problem, optimisation via simulation. This choice was made for three reasons. First, we wanted to eliminate the unrealistic assumption that all patients are punctual. Doing this leads to difficulties in evaluating a given schedule numerically, let alone giving

a closed-form expression for important performance criteria. A second reason to switch to using simulation is the opportunity this gives to using other ways to measure performance than just expectations. In many cases the fraction of time a given threshold value is exceeded is more interesting, because these actually influence the cost of running the system and patient and doctor satisfaction more than the averages. Think of the fraction of time overwork is needed, or the fraction of patients that waits longer than a certain amount of time. Also fairness can be taken into account, such as the difference in lowest and highest expected waiting times per patient for a given time period. Finally, we have noted that analytical methods like local search can sometimes be quite slow and simulation algorithms might give good results in less runtime.

A drawback is that there is no longer a guarantee that the solution found is actually the optimal solution, like we prove for the local search method. However, from the experiments we can see that often there is a very small difference between the results from simulation and those from local search, or even none at all.

The remainder of this chapter is organised as follows: in Section 2.2 we describe the model and the local search method for finding the optimal schedule, and in Section 2.3 we discuss the optimisation via simulation approach. Both methods are illustrated with numerical examples. We end this chapter with some conclusions and suggestions for future work in Section 2.4.

2.2 The case with punctual patients

To model the problem we divide the day (or part of a day) that the doctor is seeing patients into T intervals of length d minutes. In this time window we want to schedule N patients, and we assume emergency patients arrive according to a Poisson process with rate λ per interval. This can easily be generalised to interval-dependent arrival rates, but to avoid further complication in notation we only present the results for homogeneous emergency arrival processes.

Emergency patients are served as soon as possible, meaning that they wait only for the current patient in service to be finished. If more than one emergency patient is present, they are served in order of arrival. All

scheduled patients wait for emergency patients arriving during their waiting time, and they are also served in order of arrival.

Scheduled patients have a service time that has a known distribution with mean β_s, and emergency patients have a service time that is distributed according to a known distribution with mean β_e. Each scheduled patient is assumed to have a probability q of not showing up for his appointment.

The number of patients scheduled at the beginning of interval t is denoted by $x_t \in \{0, \ldots, N\}$, $t = 1, \ldots, T$. A complete schedule is then described by a vector $x = (x_1, \ldots, x_T)$ with $\sum_{t=1}^{T} x_t = N$.

Based on the schedule and the parameter values we calculate the expected waiting time $W(x)$, the expected idle time $I(x)$ and the expected overtime $O(x)$. Then the cost function for the schedule becomes $C(x) = \alpha W(x) + \beta I(x) + \gamma O(x)$, for any $\alpha, \beta, \gamma \geq 0$. The weights can be used to give relative importance to the three objectives. We are looking to minimise this cost function, so the problem then becomes

$$\min\{C(x) | \sum_{t} x_t = N, x_t \in \mathbf{N}_0\}.$$

In many cases it can be desirable to give longer waiting times or overtime a higher weight; for example when we prefer two patients waiting ten minutes to one patient waiting 20 minutes and one not waiting at all. For this case we can define $W^n(x)$ as the sum of the n-th power of the waiting times, for $n \geq 1$. Choosing n larger than 1 can lead to more fairness in the schedule, with very high values of n being practically equal to minimising the maximum waiting time over all patients.

If all patients that show up for their appointments do so exactly on time, we can calculate the three parts of the cost function exactly. For the case with early or late arrivals, this becomes impossible and we use simulation instead. For now, we assume all scheduled patients who actually show up for their appointment to arrive exactly on time. To calculate the results for a given schedule, we use the probabilities that there is a certain amount of work in the system at the moment just before or just after an arrival time. These are:

$p_{t^-}(i) = \mathbf{P}(i \text{ minutes of work in the system just before any arrivals at time } t)$,

$p_{t^+}(i) = \mathbf{P}(i \text{ minutes of work in the system just after any arrivals at time } t)$.

The probabilities can be calculated as follows. Let

$v_k(i) = \mathbf{P}$(number of arriving minutes of work including emergency work is i | k patients scheduled to arrive).

Then

$$p_{1^-}(0) = 1,$$

because we assume the system starts empty, and

$$p_{1^+}(i) = v_{x_0}(i),$$

$$p_{t^-}(0) = \sum_{k=0}^{d} p_{(t-1)^+}(k), \quad t = 2, \ldots, T+1,$$

$$p_{t^-}(i) = p_{t-1^+}(i+d), \quad t = 2, \ldots, T+1,$$

$$p_{t^+}(i) = \sum_{j=0}^{i} p_t(j) v_{x_t}(i-j), \quad t = 2, \ldots, T.$$

To compute $v_k(i)$ we need to compute the number of minutes of arriving work coming from emergency patients and the k scheduled patients for one interval, and then take the convolution of these two to get the distribution of the total amount of arriving work.

Let us first consider the amount of work related to emergency patients. Because emergency patients are assumed to arrive according to a Poisson process, but are modelled to only arrive at the start of intervals, the number arriving at the start of an interval has a Poisson distribution with expectation λ. The assumption is that if our intervals are small enough, the difference between this method and arrivals at any moment will be negligible. Let the number of arriving emergency patients be Y. Then the amount of work arriving is the Y-fold convolution of the vector representing the service time for emergency patients, $s_e^{(Y)}$. In this vector the jth elemen denotes the probability that the service time of an emergency patient is j minutes. Then the distribution of the amount of emergency work arriving at the start of any interval is given by:

$$v_0(i) = \sum_{y=1}^{\infty} s_e^{(y)} \mathbf{P}(Y = y) = \left(\sum_{y=1}^{\infty} \mathbf{P}(Y = y) s_e^{(y)} \right)_i.$$

The amount of work of a scheduled patient is 0 with probability q, the probability of a no-show, and otherwise his service time distribution is represented by the vector s_s. This holds for every patient arriving at any interval independently. So, if we denote by e_0 the vector with 1 at the first element and 0 at all other elements, the total amount of work arriving at a given interval with k patients scheduled to arrive becomes:

$$v_k(i) = \left(\left(\sum_{y=1}^{\infty} \mathbf{P}(Y = y) s_e^{(y)} \right) * \left((1-q)s_s + qe_0 \right)^{(k)} \right)_i .$$

Note that if λ can differ from interval to interval, $\mathbf{P}(Y = y)$ becomes time-dependent. This means that for every interval $v_k(i)$ needs to be computed separately. The rest of the analysis is not influenced by this modification.

The overtime is the expected amount of time the doctor has to work later than time $T + 1$, or the scheduled end of the day. This is the same as the expectation of the number of minutes of work in the system at time $T + 1$:

$$O(x) = \sum_{k=1}^{\infty} k p_{(T+1)^-}(k).$$

To calculate the expected idle time, we use the expected overtime. The total time the doctor is working is this overtime plus the scheduled duration of the day, which is $T * d$. From this we subtract the expected time the doctor has to work, and we get:

$$I(x) = Td + \sum_{k=1}^{\infty} k p_{T^-}(k) - N\beta_s - \lambda T \beta_e.$$

It should be noted that looking at idle time and overtime at the same time does not make sense in this case, because they are strongly related. They are however not equivalent, as time from the last service to the end of the day is counted as idle time because the doctor has to wait for possible emergency arrivals. In most practical situations one of the two should be chosen as a performance measure, according to the objective in the situation in question. For example, overtime can be very costly in some cases, while in other cases high utilisation or low idle time is considered more important. For this reason we include both overtime and idle time in our analysis.

The waiting time of a patient depends on the number of minutes of work in the system at the time of his arrival, any patients arriving simultaneously with him, and any emergency patients arriving before the start of his service. This makes it harder to compute the waiting time of a patient, as it depends not only on the amount of work present at the time of his arrival, but also on the interval in which he arrives.

The first patient to arrive at any given interval waits for at least all the work already present and the emergency work arriving simultaneously with him. If we denote the waiting time of the ith patient in the schedule by w_i, this is given by:

$$\mathbf{P}(w_1 = k) = \sum_{j=0}^{k} p_t^-(j) v_0(k-j).$$

For the ith patient to arrive at any interval the waiting time for the service the $i-1$ patients before him has to be added:

$$\mathbf{P}(w_i = k) = \sum_{j=0}^{k} \mathbf{P}(w_{i-1} = j)\mathbf{P}(s_s = k-j).$$

Now that we know $\mathbf{P}(w_i = k)$ we can compute the distribution of the complete waiting time of patient i arriving at interval t with the following iterative procedure. We first note that if the work present before a patient arrives is less than the length of an interval, he only waits for this length of time. Otherwise we have to add the probability of a given amount of emergency work arriving later than this patient but still served earlier, combined with the probability that this patient is actually still waiting at that time. This can continue in the same way until the end of the day, when no more emergency work arrives. Let $wt_i(k)$ denote the probability that the actual waiting time of patient i arriving at interval t is k minutes. Then we can derive the values of $wt_i(k)$ as follows:

(1) $time = t$.

(2) for $k = 0, \ldots, d-1$ $wt_i(k + (time - t)d) = w_i(k)$.

(3) $w_i^*(k) = \sum_{j=0}^{k} w_i(j+d)v_0(k-j)$ for $k = d, d+1, \ldots$.

(4) $wt_i(k) = w_i^*(k), \forall k \geq (time - t + 1)d - 1$.

(5) $time = time + 1$; if $time = T + 1$ then stop, else go back to step 2.

Even for relatively small numbers of patients to schedule and small intervals to schedule them in, the number of possible schedules becomes too large to make enumeration possible. The number of possible schedules is $\binom{N+T-1}{N}$. This means that another method of finding the optimal schedule has to be found.

The method we use here is local search, which has been used before by Kaandorp en Koole [**69**] in a setting with exponential service times and without emergencies. The local search method starts with some feasible solution, and improves this step-by-step by finding the best solution in its neighbourhood. This is repeated until a local optimum is reached. This local optimum is not necessarily the overall best solution, but for a certain suitable neighbourhood it can be shown that the local search algorithm finds the global optimum starting from any initial solution.

The neighbourhood for the local search algorithm is chosen as follows. Define

$$V^* = \left\{ \begin{array}{c} u_1, \\ u_2, \\ \vdots \\ u_{T-1}, \\ u_T \end{array} \right\} = \left\{ \begin{array}{c} (-1, 0, \ldots, 0, 1), \\ (1, -1, 0, \ldots, 0), \\ (0, 1, -1, 0, \ldots, 0), \\ \vdots \\ (0, \ldots, 0, 1, -1, 0), \\ (0, \ldots, 0, 1, -1) \end{array} \right\},$$

and take as the neighbourhood of a solution x all vectors of the form $x + v_1 + \cdots + v_k$ with $v_1, \ldots, v_k \in V^*$ such that $x + v_1 + \cdots + v_k \geq 0$. Adding one vector u_t is equivalent to moving the arrival of one patient from interval t to interval $t - 1$. The neighbourhood of x consists of all possible combinations of these one-interval shifts of patient arrivals with respect to x. The algorithm consists of the following steps:

(1) Start with some schedule x.

(2) For all $U \subsetneq V^*$:
for $y = x + \sum_{v \in U} v$ such that $y \geq 0$ compute $C(y)$;
if $C(y) < C(x)$ then $x := y$ and start again with step 2.

(3) x is the optimal schedule.

2.2.1 Multimodularity and local search

A property needed to prove that local search does indeed find the global optimum is multimodularity. For completeness we repeat the definition of multimodularity and its relations to local search, which were already given in **[69]**. Let

$$V = \left\{ \begin{array}{c} v_0, \\ v_1, \\ v_2, \\ \vdots \\ v_{m-1}, \\ v_m \end{array} \right\} = \left\{ \begin{array}{c} (-1, 0, \ldots, 0), \\ (1, -1, 0, \ldots, 0), \\ (0, 1, -1, 0, \ldots, 0), \\ \vdots \\ (0, \ldots, 0, 1, -1), \\ (0, \ldots, 0, 1) \end{array} \right\}.$$

Then multimodularity is defined as follows:

2.2.1. Definition. A function $f : \mathbb{Z}^m \to \mathbb{R}$ is called multimodular if for all $x \in \mathbb{Z}^m, v, w \in V, v \neq w$,

$$f(x + v) + f(x + w) \geq f(x) + f(x + v + w). \tag{2.1}$$

We also need the concept of an atom, as it forms the basis of our neighbourhood choice.

2.2.2. Definition. For some $x \in \mathbb{Z}^m$ and σ a permutation of $\{0, \ldots, m\}$, the atom $S(x, \sigma)$ is defined as the convex set with extreme points $x + v_{\sigma(0)}, x + v_{\sigma(0)} + v_{\sigma(1)}, \ldots, x + v_{\sigma(0)} + \cdots + v_{\sigma(m)}$.

In Koole and Van der Sluis **[75]** it is shown that for a multimodular function f a certain point x is a global minimum if and only if $f(x) \leq f(y)$ for all $y \neq x$ such that y is an extreme point of $S(x, \sigma)$ for some permutation σ.

This means that if we choose as neighbourhood for our local search algorithm all extreme points of all atoms $X(x, \sigma)$ for all possible permutations σ, we are guaranteed to find the globally optimal solution if our cost function is multimodular. The multimodularity of our cost function is what we prove next.

Because in our problem x_T is determined by $x_T = N - \sum_{t=1}^{T-1} x_t$ for given $x_1, \ldots, x_{T-1}$, our problem is $(T-1)$-dimensional. The set of possible solutions is $\{x \in \mathbb{Z}^{T-1} | x \geq 0, \sum_{t=1}^{T-1} x_t \leq N\}$. This is of course not equal

to $\mathbb{Z}^{T-1}$, but it is shown in Koole and Van der Sluis [75] that the above theorem still holds for this subset of $\mathbb{Z}^{T-1}$.

This means that we have to prove that our cost function is multimodular for the $(T-1)$-dimensional problem, which is the same as showing that the T-dimensional cost function satisfies Equation (2.2.1) with $v, w \in V^*$.

2.2.3. Theorem. *The cost function $C(x) = \alpha O(x) + \beta I(x) + \gamma W^n(x)$ is multimodular for all $u_i, u_j \in V^*$ for which $i \neq j$, for all α, β, $\gamma \geq 0$ and $n \geq 1$.*

Proof. We prove multimodularity separately for $O(x)$, $I(x)$ and $W(x)$. If two functions are multimodular then so is their sum, and that means we have a multimodular cost function. The idle time is related to the makespan, the timespan from the start of the schedule until the end of service of the last patient, for which multimodularity is easier to prove than for idle time. If the makespan is multimodular then so is the idle time, which means that we have to prove the multimodularity of the waiting time, overtime and the makespan for every possible i and j for which $1 \leq i < j \leq T$. We use the coupling method to prove this; we compare the different paths the system follows when patients are shifted to an earlier time slot or not, for given realisations of the service times and emergency arrivals. We then compare the numbers of minutes of work present to see differences in the waiting time, overtime and makespan.

We distinguish a number of different cases. These cases differ in the characteristics of their paths, and therefore need to be considered separately. Let (1), (2), (3) and (4) represent the paths using schedule x, $x + u_i$, $x + u_j$ and $x + u_i + u_j$ respectively, and let $W_{(i)}$, $O_{(i)}$ and $M_{(i)}$ represent the waiting time , overtime and makespan for path i.

I: $2 \leq i < j \leq T$

First we consider the case where $2 \leq i < j \leq T$. See Figure 2.1 for an illustration of the four paths. In this case schedules (1) and (3) are equal up to time $j-1$, and schedules (2) and (4) also follow the same path. Let the sum of the n-th powers of the waiting times of all patients arriving before or at time $j-1$, without the shifted patient, be α_1 in (1) and (3), and α_2 in (2) and (4). Now we have to make a distinction between the case where all systems become empty at some point between times i and $j-1$ and the case where this does not happen.

IA: systems empty between i and $j-1$

If the system becomes empty, schedules (1) and (2) follow the same path

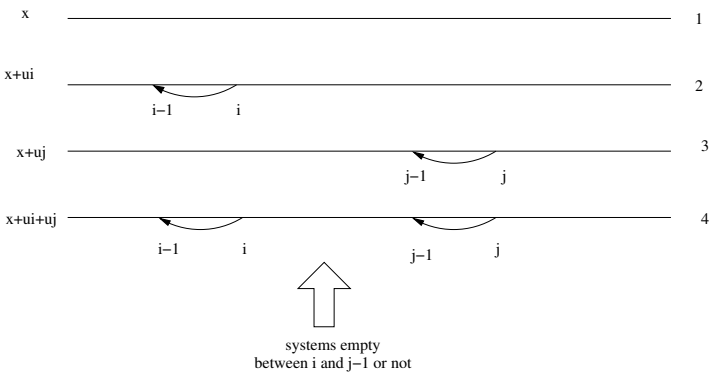

Figure 2.1: Schedule for case I.

from that moment until the last patient finishes. The same holds for schedules (3) and (4). Let the sum of the n-th powers of the waiting time of all patients from the one shifted from j to $j-1$ until the end of the schedule be β_1 in (1) and (2) and β_2 in (3) and (4). Then for the waiting time we have

$$W^n_{(2)} + W^n_{(3)} = \alpha_2 + \beta_1 + \alpha_1 + \beta_2 = \alpha_1 + \beta_1 + \alpha_2 + \beta_2 = W^n_{(1)} + W^n_{(4)}.$$

For the makespan and overtime we have that $M_{(2)} + M_{(3)} = M_{(1)} + M_{(4)}$ and $O_{(2)} + O_{(3)} = O_{(1)} + O_{(4)}$, because the end of the schedule is the same in (1) and (2), and also in (3) and (4).

IB: systems do not empty between i and $j-1$
In this case the paths of the schedules do not become equal. But we can see that the patient shifted from j to $j-1$ has a waiting time that is at most d longer in (3) than it is in (1), and also at most d longer in (4) than in (2). So if the waiting time of this patient in (1) is β_1, than it is $\beta_1 + m$ in (3) with $0 \leq m \leq d$, and if it is β_2 in (2) then it is $\beta_2 + m$ in (4), while $\beta_1 \geq \beta_2$.
For all patients arriving at time j and later, not considering the shifted patient, the waiting times are longer in (1) than they are in (3), because in (3) some work can potentially be done on the shifted patient already between times $j-1$ and j. For the same reason, the waiting times for these patients are longer in (2) than in (4). The difference in the waiting times for each patient is larger between schedules (2) and (4) than between (1) and (3), because just before the arrival of the patients at time $j-1$ more work is present in (3) than there is in (4), so in schedule (4) potentially more work

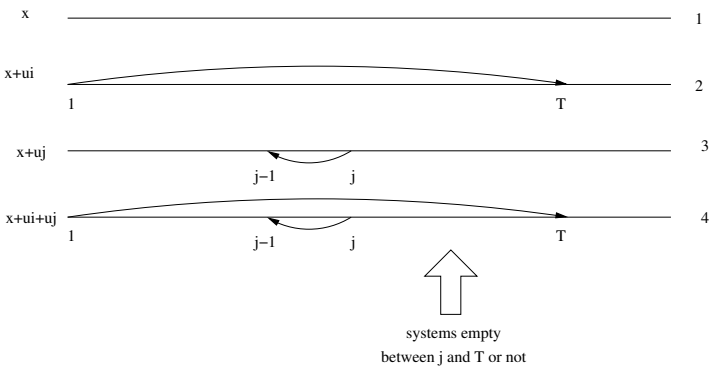

Figure 2.2: Schedule for case II.

can be done on the shifted patient between times $j-1$ and j compared to schedule (3). Let γ_i be the sum of the n-th powers of the waiting times of all customers after the one shifted from j to $j-1$ in schedule (i). Then we have $\gamma_1 - \gamma_3 \leq \gamma_2 - \gamma_4$. Now for the waiting time we get

$$\begin{aligned} W^n_{(2)} + W^n_{(3)} &= \alpha_2 + \beta_2^n + \gamma_2 + \alpha_1 + (\beta_1 + m)^n + \gamma_3 \\ &\geq \alpha_1 + \beta_1 + \gamma_1 + \alpha_2 + (\beta_2 + d)^n + \gamma_4 = W^n_{(1)} + W^n_{(4)}. \end{aligned}$$

For the overtime and makespan we can use the same reasoning as for the waiting times of the patients arriving after the one shifted from j to $j-1$, so we get $M_{(2)} + M_{(3)} \geq M_{(1)} + M_{(4)}$ and $O_{(2)} + O_{(3)} \geq O_{(1)} + O_{(4)}$.

II: $1 = i < j \leq T$
Now we consider the case where $1 = i < j \leq T$. In this case the paths of schedules (1) and (3) are equal up to time $j-1$, and also schedules (2) and (4) follow the same path up to that time. Let the sum of the n-th powers of the waiting times of all patients arriving before or at time $j-1$, excluding the patient shifted from j to $j-1$, be α_1 in (1) and (3), and α_2 in (2) and (4). See Figure 2.2. Now we have to make a distinction between the case where all systems empty at some point between times j and T and the case where this does not happen.

IIA: systems empty between j and T
From the moment the system becomes empty until time T schedules (1) and (2) follow the same path, and (3) and (4) follow the same path as well.

Let the sum of the n-th powers of the waiting times of all patients starting with the one shifted from j to $j-1$ up to and including those arriving at time T except the one shifted from 1 to T be β_1 for schedules (1) and (2) and β_2 for (3) and (4).
For the patient shifted from 1 to T we can see that in schedule (2) there is at least as much waiting time as in schedule (4), because in schedule (4) the patient that is shifted from j to $j-1$ can potentially start his service earlier compared to schedule (2). In schedules (1) and (3) any waiting time for this patient is included in α_1. Let the n-th power of the waiting time of this patient be γ_1 in schedule (2), and γ_2 in schedule (4), with $\gamma_1 \geq \gamma_2$.

$$\begin{aligned} W^n_{(2)} + W^n_{(3)} &= \alpha_2 + \beta_1 + \alpha_1 + \beta_2 + \gamma_1 \\ &\geq \alpha_1 + \beta_1 + \alpha_2 + \beta_2 + \gamma_2 = W^n_{(1)} + W^n_{(4)}. \end{aligned}$$

For the makespan and overtime we have that $M_{(2)} + M_{(3)} \geq M_{(1)} + M_{(4)}$ and $O_{(2)} + O_{(3)} = O_{(1)} + O_{(4)}$, because the end of the day is equal in schedules (1) and (3), and the work can finish earlier in schedule (4) than in schedule (2).

IIB: systems do not empty between j and T
When the system does not become empty the paths for the different schedules do not become equal. In this case we can see that the patient shifted from j to $j-1$ has a waiting time that is at most d longer in (3) than it is in (1), and also at most d more in (4) than in (2). So if the waiting time of this patient in (1) is β_1, than it is $\beta_1 + m$ in (3) with $0 \leq m \leq d$, and if it is β_2 in (2) then it is $\beta_2 + m$ in (4), while $\beta_1 \geq \beta_2$.
For all patients arriving at time j and later, not considering the shifted patient, the waiting times are longer in (1) than they are in (3), because in (3) some work can potentially be done on the shifted patient already between times $j-1$ and j. For the same reason, the waiting times for these patients are longer in (2) than in (4). The difference in the waiting times for each patient is larger between schedules (2) and (4) than between (1) and (3), because just before the arrival of the patients at time $j-1$ more work is present in (3) than there is in (4), so in schedule (4) potentially more work can be done on the shifted patient between times $j-1$ and j compared to schedule (3). Let γ_i be the sum of the n-th powers of the waiting times of all customers after the one shifted from j to $j-1$ in schedule (i). Then we

have $\gamma_1 - \gamma_3 \leq \gamma_2 - \gamma_4$.
Now for the waiting time we get

$$\begin{aligned} W^n_{(2)} + W^n_{(3)} &= \alpha_2 + \beta^n_2 + \gamma_2 + \alpha_1 + (\beta_1 + m)^n + \gamma_3 \\ &\geq \alpha_1 + \beta 1 + \gamma_1 + \alpha_2 + (\beta_2 + d)^n + \gamma_4 = W^n_{(1)} + W^n_{(4)}. \end{aligned}$$

For the overtime and makespan we can use the same reasoning as for the waiting times of the patients arriving after the one shifted from j to $j - 1$, so we get $M_{(2)} + M_{(3)} \geq M_{(1)} + M_{(4)}$ and $O_{(2)} + O_{(3)} \geq O_{(1)} + O_{(4)}$. ■

2.2.2 Complexity

The complexity of local search algorithms depends on the number of evaluations necessary to check if a given solution is a local optimum. This number is equal to the size of the neighbourhood. For an m-dimensional problem with a multimodular cost function the number of neighbours is $2^{m+1} - 2$, so for our problem this is $2^T - 2$.

This is polynomial in T, which means that it is not possible to check the local optimality of a solution in polynomial time. So our local search algorithm does not belong to the complexity class PLS as defined by Johnson et al. **[68]**, and we have to assume that (in the worst case) our algorithm has an exponential running time. However, from the numerical experiments we have performed it appears that the running times are still acceptable for problems of realistic size. For example, the running times of the experiments in Subsection 2.2.3 took one to two hours each. These experiments are for the schedule of half a day, which is most often used in practice as a time block for appointments. Of course this would be unacceptable for online algorithms, but since the problem we study here provides a blueprint for a schedule to use repeatedly this is no problem. Note that part of the runtime depends on the granularity of the time units used in the calculations. For our experiments we used minutes as the units for the waiting times, overtime and idle time, but if units of say five minutes are used the running times would be much shorter.

The neighbourhood we use in the local search algorithm is exact, which means that if a solution is better than any solution in the neighbourhood it is a globally optimal solution. This guarantees that the algorithm converges to the global optimum. It is also possible to use a smaller neighbourhood. In this case take for a solution x as neighbourhood $y = x + v$

for all $v \in U$ in step two in the local search algorithm. This gives much faster results, but there is no guarantee that the solution found is also the optimal solution.

2.2.3 Numerical experiments

The starting point for our experiments is a half-day time window, or four hours. This time is split up in 24 intervals of 10 minutes each, so $T = 24$ and $d = 10$. We want to schedule 240 minutes of work in this time period, where each regular appointment takes on average 20 minutes, and each emergency service takes on average 30 minutes. As already mentioned above, it is not really logical in the case with emergencies to look at both overtime and server idle time simultaneously, so we choose one of the two here, namely the overtime. However, the results hold as well if the idle time is taken into account instead.

In this subsection we first look at how standard even-spaced schedules behave when service times have different distributions, and how performance can be improved by changing the schedule. Then we consider the influence of the amount of emergency work on the performance of the optimal schedule, and also at what times during the four-hour period space should be reserved in the schedule. Then we study how the schedule changes with the relative importance of patient waiting times and server overtime, and how emergency arrivals influence the optimal schedule for different weights of the two performance measures. Finally we look at how the waiting times for different patients within the schedules compare to each other.

Standard practice and variability in service times

The most commonly used schedule is the even-spaced schedule. However, the performance of this schedule degrades if there are emergency arrivals and variability in service times. In Table 2.1 we compare the waiting times and overtime using the standard schedule in the case with nine scheduled patients and on average two emergency arrivals per half-day. This leads to a Poisson distributed number of emergency arrivals per interval with parameter $\frac{1}{12}$ at each time interval. The service times are deterministic in the first case, exponentially in the second case and normally distributed in the third case with standard deviation equal to that of the exponential distribution.

Distribution	Waiting time	Overtime
Deterministic	29.7	18.7
Exponential	44.0	39.8
Normal	30.6	20.2

Table 2.1: Performance of standard schedule with different service time distributions

In the deterministic case the only variability is that of emergency arrivals, and we can see that this already has a large impact on performance. After all, without emergencies both the waiting time and the overtime would be zero with deterministic service times. With the normal distributions for the service times the performance is slightly worse, but in the case with exponential service times the difference is much larger. This might be because the probability of long service times is larger in this case than with normally distributed service times. However, we can see from the results that the largest part of the waiting times and overtime is caused by the uncertainty in emergency arrivals.

In Table 2.2 we show the optimal schedule and the performances in the same three cases. The three schedules are not equal in the three cases, so the schedule is adjusted to the service time distribution. There is improvement in performance for all three scenarios. Also, for these scenarios we kept the weights for the overtime equal, but of course one could adjust the schedule by changing these weights to reflect actual priorities. This is not possible with a fixed schedule. Another thing to note is that these results seem to contradict the statement of Denton and Gupta **[35]** that higher moments only influence the optimal schedule when waiting costs are high relative to the idle time costs, or in our case overtime costs.

To make the difference between the schedules more clear we depicted the cumulative number of patients scheduled in Figure 2.3. From this picture we can see that in the case with exponential distributions the patients are shifted more towards the start of the day. The cases with normal and deterministic service times follow the same schedule for the first intervals, and toward the end the patients in the case with normal service times are shifted to earlier intervals compared to the deterministic case.

Distribution	Waiting time	Overtime	Optimal schedule
Deterministic	20.6	21.5	10101010100001010100010
Exponential	39.0	43.3	11010100100100100100100
Normal	22.6	23.8	10101010010010010010100

Table 2.2: Optimal schedules with different service time distributions

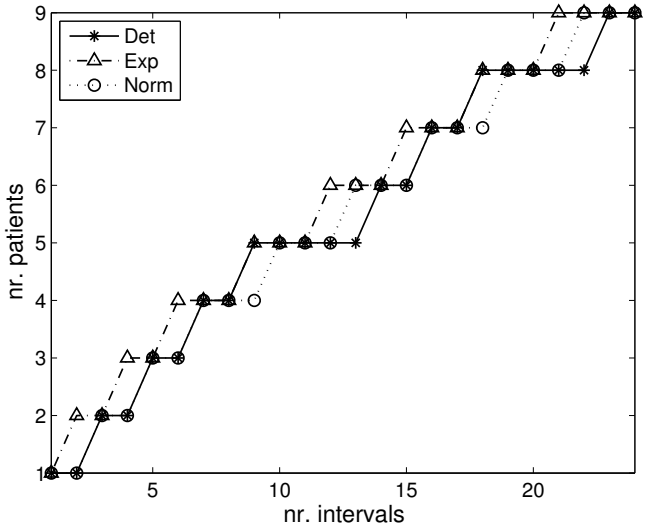

Figure 2.3: Cumulative number of patients for different service time distributions

Influence of emergencies

We noted above that emergency arrivals can have a large impact on the performance of the schedule. To further investigate how the performance changes when adding emergency work, we look at three different scenarios. The first one is the base scenario described above, with deterministic service durations and 12 scheduled patients. In the second scenario we schedule three patients less, and we expect on average 2 emergency arrivals. In the third scenario we schedule 6 patients, and expect on average

4 emergency arrivals. In all three scenarios the waiting time and overtime have equal weight. We choose deterministic service times, so the emergency arrivals are the only source of uncertainty and their influence can be seen more clearly. The results and the optimal schedule in these scenarios are given in Table 2.3.

Nr. emergencies	Waiting time	Overtime	Optimal schedule
0	0.00	0.00	101010101010101010101010
2	20.59	21.40	101010101000010101000010
4	39.08	31.13	101010000100001000010000

Table 2.3: Influence of emergencies on schedule performance

The first scenario is the ideal scenario where the whole schedule is executed according to plan, because there is no variation in either arrival moments or service durations. In this case the optimal scenario is of course to schedule time equal to the service duration for all patients. In the second and third scenario we can see that performance decreases with emergency arrivals, as is to be expected. The open space in the schedule is concentrated more towards the end of the day than at the beginning, because the probability that an emergency patient has arrived is very small early in time and open space there would often lead to unnecessary server idle time.

Relative importance of performance measures
Again we look at three different scenarios, all without emergencies and now with exponential service times. We change the weights for the waiting time and the overtime to see how the schedule and performance change. The results can be seen in Table 2.4.

From the results we can see that for higher relative weight for the waiting time, the arrival times for the patients move more towards the end of the day. This gives longer interarrival times, and so less waiting time for each patient. For the overtime we see exactly the opposite effect.

To make the difference between the schedules more clear, in Figure 2.4 we compare the cumulative number of patients that have been scheduled over time. From this picture it is clear that when the waiting time

α_W	α_L	Waiting time	Overtime	Optimal schedule
1	1	28.18	35.26	20101010101010010101010 0
10	1	19.21	58.07	11010010010100100101010 2
1	10	46.92	28.12	21110110101010010100000 0

Table 2.4: Influence of weights on the optimal schedule

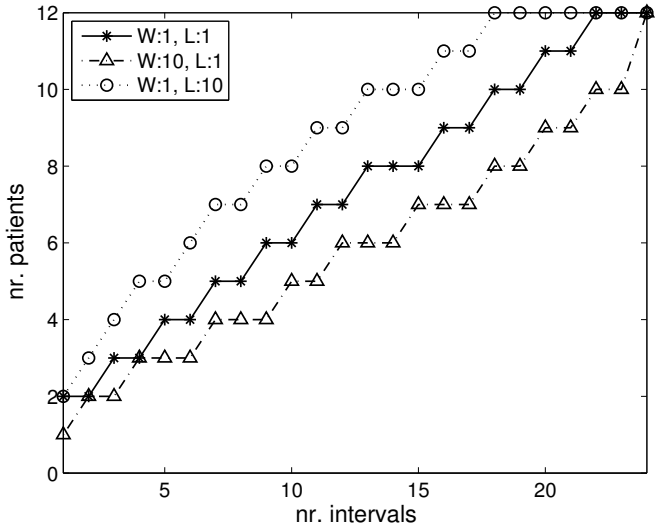

Figure 2.4: Cumulative number of patients

is weighed more heavily, the number increases slower over time. In other words, the patients are shifted more towards the end of the time available.

To see if the same principles hold in the case with emergencies, we also give the results for three scenarios with 2 and 4 expected emergency arrivals, again with exponential service times. The results can be seen in Table 2.5. We can see from the results that the performance generally gets worse on both waiting time and overtime as the portion of emergency work increases. It turns out that the scheduling principles that hold in

Emergencies	α_W	α_L	Waiting time	Overtime	Optimal schedule
0	1	1	28.18	35.23	20101010101010010101010
2	1	1	39.00	43.26	11010100100100100100100
4	1	1	49.72	52.15	11001000100000100001000
0	10	1	19.21	58.07	11010010010100100101010
2	10	1	30.86	60.60	11001000100010001001011
4	10	1	43.01	66.32	11000010000000010000010
0	1	10	46.92	28.12	21110110101010010100000
2	1	10	58.83	36.14	21101010100100100000000
4	1	10	68.06	45.12	21010010010000000000000

Table 2.5: Influence of weights on the optimal schedule

the case without emergencies have an even stronger influence if there are emergencies: more emphasis on waiting time moves arrivals towards the start of the schedule, and more emphasis on overtime has the opposite effect.

No-shows and emergencies

The effect of no-shows and emergency arrivals are opposite in some sense; emergency arrivals cause more work than scheduled, while no shows decrease the amount of work that has to be done. On the other hand, since both factors are usually accounted for in the schedule somehow, they both have the effect of adding uncertainty to the process. So it would be interesting to see what will happen if both occur.

In Table 2.6 we compare the effect of no-shows and emergencies for different distributions of the service times. The mean service times are the same as in the other experiments. The parameters are chosen in such a way that the total amount of work is equal in each case to ensure a fair comparison. The weights for waiting time and overtime are equal in all cases.

In general we can see that having emergency arrivals is worse for performance than no-shows. Having both at the same time will not cancel out the effects of either, but rather has an even larger effect on performance, although this effect will not be as large as the sum of the effects of both. Having only no-shows already gives results that are a lot worse than the

Nr. sched	Nr. em	No-show %	Dist	WT	OT	Optimal schedule
12	0	0	Det	0	0	10101010101010101010101010
16	0	25	Det	16.74	17.69	20102010101020101020101010
9	2	0	Det	20.6	21.5	10101010100001010100001010
12	2	25	Det	27.67	27.58	20101010100101010100101010
12	0	0	Exp	28.18	35.23	20101010101010010101010100
16	0	25	Exp	33.79	39.10	21110110101101011010110100
9	2	0	Exp	39.0	43.3	11010100100100100100100010
12	2	25	Exp	42.77	47.14	21010101010010101001010100
12	0	0	Norm	4.00	7.09	10101010101010101010101010
16	0	25	Norm	17.92	19.35	20110110110101101101101010
9	2	0	Norm	22.6	23.8	10101010010010010010010010
12	2	25	Norm	29.03	27.67	20101010101001010101010100

Table 2.6: No-shows and emergencies

case without no-shows and the same expected amount of work. The same has already been noted about emergencies. For both emergencies and no-shows the effect is larger when the variation in service times is smaller. This might be because variation in service times does have a smoothing effect throughout the time period.

Waiting times per patient
The objective function contains the expected waiting time averaged over all scheduled patients. It does not take into account how the total waiting time is distributed over the patients. We consider a case with 9 scheduled patients and on average 2 emergency patients per day, and deterministic service times and exponential and normal distributions for the service times with equal variances. In Table 2.7 we show the expected waiting times for the nine scheduled patients with the patients numbered in order of arrival when using the optimal schedule shown in Table 2.2.

We see in all three cases that the waiting times differ considerably per patient. The first patients have low expected waiting times, since they only wait if there is an emergency arrival immediately at the start of the day. In all three cases the waiting time per patient then increases rapidly and stabilises or decreases slightly for the last patients. As we had already

Distribution	1	2	3	4	5	6	7	8	9
Deterministic	3.33	9.99	16.65	23.28	29.85	21.04	27.23	32.81	21.13
Exponential	3.20	21.57	32.58	42.70	45.76	48.72	51.07	52.51	52.90
Normal	3.21	11.25	18.84	26.16	27.23	28.51	29.35	29.66	28.94

Table 2.7: Expected waiting times per patient

seen for the average waiting times, those with exponentially distributed service times are much higher than in the case with normally distributed served times.

2.3 Adding early and late arrivals

In the previous section we described a method for finding the optimal schedule with the assumption that every patient who shows up for his appointment does so exactly on time. When we add early and late arrivals to the model the cost function is not multimodular in this case. The reason for this is that now shifts of patients from one time slot to the preceding slot can cause changes in the order in which the patients are served. This means that the cost function is no longer more or less smooth. So we have no analytic method for evaluating schedules, let alone finding optimal schedules. Here we use simulation as a method to evaluate a given schedule. First we explain how the different performance measures are calculated in the simulation. We assume that patient unpunctuality follows some known distribution with the scheduled arrival time as a reference point. We take this distribution to be identical for all patients and independent of the scheduled appointment time and of the arrivals of all other patients. From research it is known that most patients arrive early, see for example Brahimi and Worthington [**23**], so the mean of the distribution should probably have a negative value.

The reason why it is not as straightforward to evaluate the performance of a given schedule as it was in the case of punctual patients is that the order in which the patients are being seen is no longer necessarily equal to the scheduled order. If one patient is late and the one scheduled next is early, the later patient might arrive first. Then, when an appointment finishes, the doctor has a choice between waiting for the patient with

the earlier appointment time to show up or start the next patient's appointment. The first choice leads to more idle time and overtime as well as increased waiting time later in the day, while the second option can lead to unfairness if one patient is really early and the other one is not actually late for his appointment time. And even if both patients are present in the waiting room there is a choice between sticking with the scheduled order or to see patients in order of arrival.

Here we make the following choice for the order of service: when an appointment ends, the doctor sees the patient with the earliest appointment time from all patients who are present in the waiting room. If there is no patient in the waiting room, the doctor has idle time until a patient arrives and he starts that patient's appointment. We made this choice because in this way we have the fairness of sticking as close to the schedule as possible without any avoidable idle time. In a case with emergency arrivals, all emergency patients take precedence over the scheduled patients in choosing the next patient to see.

Now we have to define how to calculate the idle time, overtime and waiting time. To start with overtime, this is again defined as the time the doctor spends working after the scheduled end point of the period. Note that if the unpunctuality distribution has no cutoff point there could potentially be patients arriving later than this time point. We make the extra assumption that the distributions are chosen in such a way that this does not occur, but an alternative would be to just have the doctor choose to stop as soon the waiting room becomes empty after the end of the clinic day.

For the doctor idle time we again use the end time of the day plus the overtime, minus the expected amount of work. This is the same as in the case without early and late arrivals. If an unpunctuality distribution is chosen in such a way that there is a possibility of patients arriving after the scheduled end time, the expected amount of work needs to be decreased by the expected number of patients arriving after this time times the mean appointment duration for scheduled appointments. Again, idle time and overtime should not be used at the same time to evaluate a schedule because they are two ways of looking at the same thing.

The last point we need to discuss is the computation of the waiting time. There are two ways to measure the waiting time for a given patient. The first is to start counting from the moment the patient arrives, regardless of the scheduled time. So if a patient is very early, it might result in a

long waiting time even though the schedule would not give this result if the patient had been on time. The other way to measure waiting time is from the maximum of the scheduled time and the time the patient arrives. We choose the latter in our experiments.

2.3.1 Optimisation via simulation

Now that we can evaluate a given schedule, we can use optimisation via simulation techniques to find optimal or near-optimal solutions. As a side note, there is no agreement in the literature on the term for this set of techniques; other terms that are used are "optimisation for simulation" and "simulation optimisation". All these terms are used for the same set of techniques and theories.

The range of techniques available for optimisation via simulation is large and varied. The reason that they are not equal to optimisation methods used in a deterministic setting is that there is a difference between finding an optimal solution and estimating the objective function value for the optimal solution. In a deterministic problem both are done simultaneously. If simulation is used to evaluate solutions, computational effort must be divided over both goals and it will always be impossible to decide with absolute certainty if one solution is better than another. See Fu [**47**] for a thorough discussion of this topic.

Good overviews of the available techniques are given by Andradóttir [**8**] and Fu [**46**]. A number of well-known techniques for optimisation via simulation are only suitable for problems with continuous decision variables. These methods are based on gradient estimations, and examples are finite difference stochastic approximation and simultaneous perturbation stochastic approximation.

Since the decision variables for the appointment scheduling problem are discrete, these methods are not suitable for the problem at hand. It would of course be possible to formulate the problem in a continuous way using the appointment times of the individual patients as decision variables, but this would result in solutions with appointment times that have to be rounded to be useful in practice. And if that is the case, it appears to be more logical to work with discrete variables in the first place. There are methods that can be used for both discrete and continuous problems, and methods that are especially suited to discrete variables. We will discuss

the most important of these two types of methods here and explain our choices.

For problems with only a small number of possible solutions a suitable method to use is ranking and selection. This method is essentially a way to divide limited computing budget over all possible solutions and to maximise the probability of choosing the best solution. All solutions are quickly simulated and the promising ones are given more computing budget. This is of course only possible for problems with small solution spaces, even with the more enhanced ranking and selection procedures that have been proposed in the literature and that can handle more solutions. For the problem considered here this method is not really suitable, as the number of possible solutions grows exponentially both with the number of intervals and the number of patients to be scheduled.

Other methods for optimising over discrete decision variables are often some form of random search method. This term covers a whole range of methods. A good overview can be found in Andradóttir [**7**]. There are global and local random search methods. Both types have in common that the algorithm starts with some solution, chooses a possible next solution according to a sampling strategy, and then makes a decision about how to proceed based on simulation results of the current and candidate solutions according to a rule that may involve randomness. The many algorithms differ in how they choose solutions to be considered, what the rule is that determines the next step, and how the estimate for the optimal solution and its objective function value are determined.

The important decisions to make in setting up a random search algorithm are how to select one or more candidate solutions in each step of the algorithm, the way of keeping track of and selecting the final solution or answer, and how to allocate computing budget. In its simplest form, the random search algorithm randomly draws one new candidate solution from the neighbourhood of the current solution, samples both current and candidate solution a few times and chooses the one with the best average performance as the new current solution in the next step. The final solution is then the solution the algorithm has visited most often during the course of running. If the neighbourhood of a candidate consists of the complete solution space the algorithm is called global random search, otherwise it is a form of local random search. Both options were described by Andradottir [**5**]. It has been shown a few years later by Andradóttir [**6**] that these naive versions of random search can easily be improved by not

keeping track of the number of times a solution has been visited by the algorithm, but of the sample mean of each solution instead. When the algorithms were first developed, this was harder because of the lack of memory, but that problem has since been resolved.

The difference between local and global random search algorithms is that in local search the candidate solutions are chosen randomly from a certain neighbourhood of the current solution, instead of from the whole solution space as in global random search algorithms. The choice of neighbourhood is essentially open, as long as any solution can be reached from any starting point with positive probability. The algorithms for global random search are designed in such a way that in the limit, or in other words with infinite computation effort, all solutions are visited and simulated infinitely many times to completely ensure convergence to a global optimum. In practice they give good results with finite computational effort because they favour good solutions in the algorithm and estimate the optimal solution using information gathered along the way. But of course actually visiting and making enough simulation runs for all possible solutions is impossible for many problems, including the one considered here, so that global convergence to the optimum has little practical value.

Another way to search for good solutions instead of through definition of a neighbourhood, is through some stochastic version of branch-and-bound. The basic idea here is to divide the solution space into continuously smaller subsets, and choosing to explore one of these subsets in more detail while also keeping an eye on the rest of the solution space. It is therefore not the same as splitting the solutions space by relaxing integrality constraints as is often done in deterministic problem solving. The best known example of stochastic branch-and-bound methods are the nested partitions method by Shi and Olafsson [**98, 99**] and Pichitlamken and Nelson [**91**].

Another possible method for finding solutions using simulation is the ordinal optimisation approach described by Ho et al. [**64**]. This method is very well suited to problems with a very large solution space, and no known structure to the problem. The approach is based on the idea that comparing solutions using simulation is much easier, i.e. faster, than accurately estimating the performance of a solution. The other crucial point is that it relaxes the goal of optimisation: we are no longer looking for the very best solution, but for one of the top $n\%$ solutions. The method consists of randomly selecting a reasonable number of solutions from the

solution space and determining the best of them using simulations. Then, for a given number of solutions drawn and an idea of the size of the solution space, we can determine the probability that our selected best solutions contain at least one of the overall top $n\%$ solutions. Note that this does not mean that the performance of our selected solution is more than a given percentage away from the optimal one, but only of the rank of the solutions found.

We chose to use random search to address the appointment scheduling problem with unpunctual patients. As was already explained above, we feel it is more natural to use methods suited for optimising over discrete variables. This rules out all gradient-based methods. The methods of ranking and selection are not scalable enough to use in real-sized problems, so they are not that interesting in this case. The ordinal optimisation method is suitable for our problem, but it does not make use of the structure in the form of the solutions, or schedules. The random search method is also a natural extension of the local search algorithm used in the case with all punctual patients. In Section 2.3.2 we give the details of the random search algorithm used, and in Section 2.3.3 we present numerical experiments to demonstrate the performance of the methods.

2.3.2 Random search algorithm

The first decision to be made in defining random search is how to define the neighbourhood of a given solution. Because the structure of the problem and the solution is not known, we will not be able to use this information in designing a good algorithm. From Section 2.2 we know that local search works well for the same problem when all patients are assumed to be punctual; it is guaranteed to find the optimal schedule within a reasonable amount of time. With this knowledge, we decided to try a random search algorithm with a similar neighbourhood.

We chose local random search over global random search because from a practical viewpoint finding a good schedule in a reasonable running time, even if not optimal, is more valuable than having guaranteed convergence to the optimal schedule. The algorithm we use works as follows: we start with a schedule chosen randomly from all possible schedules. This schedule is simulated a few times, and a next schedule is chosen randomly from a neighbourhood in which all schedules have equal probability of being drawn. This is also simulated a few times, and then with a certain

(high) probability we choose the best of the two solutions as the next one, and with small probability we choose the other one. This ensures that the algorithm improves steadily, but the small probability of choosing the worse solution provides a way out of a local optimum.

As neighbourhood of a schedule we choose all possible schedules made up of a shift of a patient from one interval to the one just before it compared to the current schedule, or if the patient is now scheduled at the first interval, to the last interval. Note that this is not the same neighbourhood as used in Section 2.2, but smaller. We made this choice because the performance of all schedules in the original neighbourhood would differ so much that the algorithm started to drift aimlessly as it were, and the smaller neighbourhood does not result in that behaviour. Since it is still possible to reach every possible schedule from any starting point, this restriction does not give any problems.

To decide on the final solution we use all information available up to that point, as this is shown in [**6**] to accelerate the algorithm and to lead to better results than just taking the final solution or the schedule visited most often as the outcome of the algorithm. Information about the number of times an algorithm is simulated and the average objective value over all simulations is needed to compute this. It is only necessary to keep information on schedules that have actually been visited; there is no need to save information on all possible schedules which could become problematic for real-sized problem instances. If necessary we can restrict the final outcome to only those schedules that have been simulated a minimum number of times, so as to lessen the influence of randomness in the performance evaluations. Because of the higher probability of choosing the better solution in each step of the algorithm these are probably the better solutions in any case.

This algorithm can end up in a local optimum, and it might be very hard to get out of there again if the difference in performance between this local optimum and its neighbours is large. Whether this happens or not depends on the randomness in the algorithm, but it can also be influenced by the choice of the starting point. We don't know the structure of the value function over the solution space, so we cannot be sure that this does not happen. This is why we choose to restart the algorithm a few times in a new solution which is again randomly chosen from all possible schedules. This gives a greater chance of finding a global optimum, and if not, then at least a better local optimum.

2.3.3 Numerical experiments

To demonstrate the performance of the algorithm described in the last section, we perform some experiments with a very small example. In this example we try to optimally schedule $N = 3$ patients in $T = 6$ time slots of $d = 10$ minutes each. This combination of T and N leads to 56 possible schedules. The reason for choosing such a small example is that we can actually simulate the performance of all schedules often enough to be reasonably certain which one is the optimal schedule. Then we can compare the outcome of the optimisation algorithm to these results.

For the service time we assume an exponential distribution with a mean of 20 minutes. For the non-punctuality we assume a normal distribution around the scheduled arrival time of the patient, with a mean of 10 minutes earlier and standard deviation 10. The arrival time and service time are assumed independent of each other and of those of other patients. The objective function is the sum of the total waiting time with weight $\alpha_W = 3$ and the tardiness with weight $\alpha_T = 1$. The optimal schedule in the case where all patients are punctual is 1-0-1-0-0-1, with an objective value of 35.97. We can compare this with the results from the simulation optimisation, but this schedule may not be optimal in the case with early and late arrivals.

First we simulate each possible schedule a large number of times, to see which results we would like to see from the optimisation algorithm. The best schedules and their performance are shown in Table 2.8.

There is quite some variability in the performance of the schedules from one simulation run to another. Also, we can see that the optimal schedule in this case is 2-1-0-0-0-0. The schedule that is optimal in the case with punctual arrivals is not among the ten best schedules. Next we run a local random search algorithm ten times, and the results from the runs are given in Table 2.9.

We can see that the algorithm does not find the optimal schedule, but it finds the second-best one twice and the next one three times. The objective values are very close, so the result is near-optimal.

Now we look at a larger problem, with 6 patients scheduled into 12 time slots and all other parameters the same as in the smaller example. Again we run the local random search algorithm with ten randomly chosen starting points, we get the results in Table 2.10.

In many of the resulting schedules there are no patients scheduled in the first time slot. This can be explained by the large fraction of patients

Schedule	Mean	Standard deviation
2-1-0-0-0-0	85.428	93.955
2-0-1-0-0-0	86.058	91.901
0-2-0-1-0-0	86.884	94.550
2-0-0-1-0-0	86.991	89.863
0-2-0-0-1-0	89.021	93.718
3-0-0-0-0-0	90.214	98.265
0-3-0-0-0-0	90.296	102.840
2-0-0-0-1-0	90.458	91.703
0-0-2-1-0-0	90.947	99.703
0-2-0-0-0-1	91.294	93.165

Table 2.8: The ten best schedules with mean and standard deviation of the objective function after $100,000$ simulation runs.

that arrives early for their appointment. Then when two or more patients are scheduled at the second time slot, there is a high probability that at least one of them will already be present at the start of the day.

Of course in this case there are too many schedules to simulate them all enough times to get a good estimate for the objective value. This means that we do not know if the algorithm found the optimal schedule, or one very close to optimal. But from the smaller example we get an indication that the solutions found will be at least good enough for practical use.

2.4 Conclusions

In this chapter we have first presented a method for finding the optimal appointment schedule in a setting with emergency arrivals or interruptions when all patients who arrive do so exactly on time. The method uses general service time distributions and can handle no-shows, which makes it suitable for use in practice. It finds the optimal arrival times for a weighted combination of patient waiting time, doctor idle time and over-

Schedule	Objective value
2-0-1-0-0-0	85.596
0-3-0-0-0-0	85.771
1-2-0-0-0-0	85.571
0-2-0-1-0-0	85.917
0-1-1-0-1-0	85.748
0-0-3-0-0-0	85.524
0-2-0-1-0-0	85.747
0-2-0-1-0-0	85.754
0-2-1-0-0-0	85.792
2-0-1-0-0-0	86.045

Table 2.9: The ten schedules and objective values resulting from ten random search runs with random starting points.

time as the objective. The method makes use of a local search algorithm, which for our multimodular objective function is guaranteed to find the global optimum. From the numerical examples we presented, it can be seen that in general more free space for emergencies is reserved towards the end of the day, or in other words, the inter-arrival times increase over the day. The same holds for space in the schedule reserved for dealing with variation in the appointment durations.

Second, we studied the case where patients can arrive early or late for their appointments. Here we used simulation to evaluate a given schedule, and presented a simple but effective algorithm for finding good schedules. This method used a similar neighbourhood as in the local search algorithm to randomly explore the solution space. There is of course no guarantee that the result is optimal, but numerical experiments show that the results are near-optimal, and certainly good enough for practical use.

The simulation method is very flexible and can be used to optimise more complex situations, for example a case where patients do not all have equal service time distributions. Also, even while this algorithm seems to

Schedule	Objective value
0-2-0-1-0-1-0-1-0-1-0-0	122.887
0-0-0-0-4-0-1-0-0-0-1-0	110.500
2-1-1-0-0-0-0-1-1-0-0-0	115.073
0-0-3-1-0-1-0-1-0-0-0-0	120.121
0-0-3-0-1-1-0-0-1-0-0-0	123.213
0-0-2-1-0-1-0-1-0-1-0-0	118.751
2-1-0-1-0-0-0-0-0-0-0-2	126.491
0-3-1-0-1-0-0-0-0-0-0-1	111.190
3-0-1-0-0-0-0-0-0-1-0-1	119.293
0-0-2-1-0-1-0-0-1-1-0-0	122.859

Table 2.10: The ten schedules and objective values resulting from ten random search runs with random starting points for a larger problem.

be very effective in finding near-optimal solutions, there are other methods that might do so as well. A logical candidate would be the nested partitions algorithm mentioned before. It remains to be seen whether these are also faster than the random search algorithm presented here.

3 Elective admission scheduling

This chapter treats the problem of scheduling patient admissions to the wards in a hospital. With the growing demand for health care resources, the pressure on the efficient usage of the available bed capacity on the wards is increasing, and the admission scheduling has a large influence on the efficiency. The workload at clinical wards is often highly variable, leading to the need for extra capacity to respond to peaks in demand for beds. In addition to these extra capacity requirements, the variability in workload has other negative side effects. Litvak et al. [**77**] show that reducing variability in bed demand helps to reduce the stress of the nursing staff and to improve the safety of patients.

Surprisingly, studies have shown that the variation in the number of scheduled patients admitted is generally at least as large as the variation in the number of emergency admissions, and often larger, see e.g. McManus et al. [**83**] and de Bruin et al. [**25**]. The variability in admissions leads to highly variable bed occupancy. The admission process is also largely affected by the schedule of the Operating Theater (OT). The OT schedule allocates the available operating time to the different surgical disciplines, but, in most cases, it does not specify which or how many procedures are to be executed in the allocated times and so the number of patients admitted on every weekday can vary significantly. This OT schedule results in a weekly bed occupancy pattern, but the number of occupied beds on each day can still vary significantly from week to week. Moreover, during the weekend the number of elective admissions is generally very small, leading to extra workload fluctuations over the week.

A simple way to reduce the variability would be to admit a fixed number of patients every day of the week. This could be implemented using a fixed quotum for the number of daily admissions, thereby removing any unnecessary variation in demand. The remaining variation would solely be due to emergency arrivals and variations in the length of stay in the hospital.

The absence of a substantial number of scheduled admissions during the weekend complicates the use of a fixed quotum per day. In addition, it is current practice in most hospitals that the number of staffed beds is lower during weekends, partly because of higher staffing costs. A fixed daily quotum (for every day of the week) would not accommodate this,

but yields the same expected bed demand every day of the week. An alternative approach is to use different admission quota for the days of the week, taking differences in length of stay (LOS) between patient types into account. In this chapter we determine the number of scheduled admissions for every day of the week, with the objective of keeping the bed demand as close as possible to a predetermined target load. This target can be different for every day of the week, thereby accommodating a lower number of staffed beds during the weekends. The result will be a set of quota for the number of scheduled admissions for different patient types on every day of the week.

The main goal of this chapter is to determine admission quota for scheduled admissions and the impact of variability in the number of admissions on the required bed capacity. First, we study approximations for determining the impact of the daily variability in the number of admissions, for both stationary and time-dependent admissions with a weekly pattern. This results in intuitive approximations for the variability in bed demand and for blocking probabilities. Second, we use these results in an optimisation model that minimises the weighted deviations of the load from a predetermined target load, which can differ from day to day. We incorporate emergency arrivals, routing of patients over different wards and multiple patient types, each type having a specific phase-type LOS distribution. Our primary focus is on bed demand, where the surgery scheduling may be included as a constraint.

Patient scheduling has received quite some attention in the literature, mostly focusing on the scheduling of surgeries. An example of work that studies surgery scheduling in combination with bed usage is Beliën and Demeulemeester [**16**], who try to level the bed usage by finding the best allocation of OT time blocks to surgical disciplines. They view the number of patients admitted on a day as a stochastic variable with a distribution depending on the specialty that used the OT. Van Oostrum et al. [**88**] find the optimal so-called master surgical schedule, in which they schedule all regularly performed surgeries on a specific day in the planning cycle, with a combination of OT time usage and the maximum number of beds needed on every day as the objective function. They treat the length of stay as deterministic, with the length depending on the type of surgery performed. Vanberkel et al. [**102**] study the effect of a given surgical schedule on the usage of beds, taking emergency arrivals and different ward types into account as well. However, they do not use an optimisation algorithm and

only try to improve step-by-step by trial and error. Their approach has been applied in practice with good results.

Gallivan and Utley **[49]** present a generic model for determining the distribution of bed occupation for a given cyclic admission schedule. They give an example of how these results could be used in an optimisation context. They restrict themselves to a single ward. Adan et al. **[1]** present a case study in which they apply an optimisation model. They consider both the OT usage and several other types of resources, such as different wards visited by patients consecutively. A weighted combination of the overutilisation and underutilisation of all these resources is minimised, in both a deterministic and a stochastic version. The stochastic version cannot be solved to optimality due to its size in their case study setting, although they do believe that taking randomness into account is important.

The remainder of this chapter is organised as follows. We start with quantifying the impact of variability on the required bed capacity. In Section 3.1 we use approximation methods for analysing models with non-Poisson arrivals and we analyse time-dependent arrivals, to allow for a weekly arrival pattern, in Section 3.2. In Section 3.3 we discuss admission scheduling that results in a stable bed demand by applying a Quadratic Programming model. We conclude with Section 3.4 where we discuss the contribution of this chapter and describe the main practical insights that can be derived.

3.1 Variability in scheduled admissions

The arrival process of emergency admissions is generally well approximated by a Poisson process. Although elective admissions are scheduled, our experience is that the variability in the number of elective admissions is at least as large as the variability in the number of unscheduled admissions, which is also supported by various studies, see e.g. **[25, 83]**. Given the variability in both types of admissions, the Erlang loss (or delay) model is often well applicable for giving insight in the implications of capacity decisions for clinical wards, see for example **[25, 82]**.

In this section we quantify the impact of a more stable (elective) arrival stream and the corresponding appropriate capacity. Equally, this may be used to determine a target load in Section 3.3. We build on approximations in the literature to analyse models with a general stationary arrival process

that is not necessarily Poisson. The approximations described here are further adjusted in Section 3.2, where we study systems with non-stationary arrivals.

For the Erlang loss model, the capacity is fixed at s beds. Patients are assumed to arrive according to a Poisson process with an average of λ per day. An arriving patient is admitted in case a bed is available and refused otherwise. An admitted patient stays for a stochastic duration (the length of stay) at the ward with an average of β days. By Little's formula, the above implies that the offered load is $\rho := \lambda\beta$, which represents the average number of occupied beds in case there would always be sufficient capacity.

This $M/G/s/s$ model has been well studied. The probability that an arriving patient is refused, also called the blocking probability, is then given by

$$B(s,\rho) = \frac{\rho^s/s!}{\sum_{k=0}^{s}\rho^k/k!}.$$

Moreover, the offered load (number of patients present in case of sufficient capacity) has a Poisson distribution, which can be well approximated by a normal distribution for ρ not too small. In particular, for the Poisson distribution the mean and variance are equal, which directly yields that the variance of the offered load can then be approximated by ρ.

To obtain insight in the impact of scheduled admissions it is required to eliminate the assumption of Poisson arrivals, which is crucial for most queueing models. This elimination leads to a $G/G/s/s$ queue.

We approximate the G/G/s/s queue using its infinite-server counterpart G/G/∞. We assume a stationary arrival process, where arrivals occur at rate λ. The coefficient of variation of the interarrival time is denoted by c_a. The service times (lengths of stay) are assumed to be independent and identically distributed with mean β. We also introduce the so-called Gini coefficient, which surprisingly appears in the approximations. This measure is related to the Lorenz curve, which is used in economics to represent the inequality in the distribution of wealth or income among the citizens of a country. Here we use it for the inequality in the length of stay S among patients (see also [**25**]). The Gini coefficient is defined as

the area under the Lorenz curve. For piecewise differentiable probability distributions, the Gini coefficient (G), proposed in **[37]**, is given by

$$G = 1 - \frac{1}{\mathbb{E}S}\int_0^\infty \mathbb{P}(S > y)^2 \mathrm{d}y.$$

For example, for a deterministic distribution we have $G = 0$, and for an exponential distribution $G = \frac{1}{2}$. In **[25]**, the Gini coefficients are given for the LOS at different wards.

We start with an approximation for the number of busy servers, or rather the variance and distribution of the number of busy servers, for a G/G/∞ system in heavy traffic. If we use the heavy-traffic approximation established in **[22]**, we have that the number of busy servers X_ρ approaches a normal distribution in the limit when the load $\rho = \lambda\beta$ of the system tends to infinity:

$$\frac{X_\rho - \rho}{\sqrt{\rho z}} \to N(0,1), \text{ as } \rho \to \infty,$$

with

$$\begin{aligned} z &= 1 + (c_a^2 - 1)\frac{1}{\mathbb{E}S}\int_0^\infty \mathbb{P}(S > y)^2 \mathrm{d}y \\ &= 1 + (c_a^2 - 1)(1 - G), \end{aligned} \tag{3.1}$$

where the second equality follows directly from the representation of the Gini coefficient. We note that only the first equality seems to be available in the literature, and the interesting and useful relation to the Gini coefficient has not yet been observed. The z is a measure of the peakedness of the arrival process and the service times, see **[108]** for a more elaborate discussion. Here, the variance of the number of busy servers is $z\rho$. From the peakedness we can see that the variance increases with the squared coefficient of variation of the interarrival times c_a^2 as is to be expected, but it can either increase or decrease in the Gini coefficient depending on the sign of $(c_a^2 - 1)$. This means that reducing the variability in LOS is only beneficial in cases where the arrival process is already quite stable, implying that hospital managers should focus first on stabilising the arrival process before stabilising the LOS distribution. Note that the point at which $(c_a^2 - 1)$ changes signs corresponds to a Poisson arrival process.

The variability in offered load is of prime importance for the required amount of capacity. Based on the square root staffing rule, see e.g. **[60,**

109], the required number of beds is typically the mean offered load (ρ) plus a constant times the standard deviation in offered load ($\sqrt{z\rho}$). The latter term corresponds to buffer capacity to deal with variability in bed demand. The value of the constant depends on the service level target, but is often chosen to be between 1 and 2.

The most natural performance measure for the G/G/s/s queue is the blocking probability B_c. In [**108**] the Hayward approximation is proposed, which is given by

$$B_c = B_c(s, \rho, z) \approx B(\frac{s}{z}, \frac{\rho}{z}). \tag{3.2}$$

In other words, we use the regular Erlang loss formula, but first divide both the number of servers and the load of the system by the peakedness z. This requires an extension of the Erlang loss formula to non-integer values for the number of servers, see [**67**].

From this approximation, it follows that the fraction of blocked arrivals increases as the peakedness increases. Hence, the loss probability increases with the coefficient of variation of the interarrival times, but can either increase or decrease with the Gini coefficient depending on the variability in the arrival process, just as the variance of the number of busy servers in the system with infinite capacity.

We have performed some numerical experiments to obtain insight in the impact of elective admissions on the bed occupancy in hospitals. Since the Hayward approximation is available in the literature, it is not our goal to carry out an extensive numerical analysis. As a base example, we consider an average-sized ward with 28 beds, see [**25**]. We present experiments with three different distributions for the length of stay, all with mean 4. The LOS at clinical wards can typically be represented by exponential or hyper-exponential distributions, whereas the deterministic LOS is included to obtain insight in the impact of the LOS characteristics. We consider (mixed) deterministic and Poisson arrivals, representing scheduled and emergency admissions. The average number of arrivals per day is $41/7$, giving an average offered load of about 23.43.

The standard deviation in offered load and the fraction of refused admissions (blocking probability) for the different scenarios may be found in Table 3.1. These values have been calculated using approximations and have been verified by simulation. We see that the standard deviation in offered load and the loss percentage increase with the share of Poisson arrivals.

Arrivals	LOS	Stdev. offered load	Loss fraction
Det	det	0	0.00 %
	exp	3.42	2.51 %
	H_2 ($p_1 = 0.1$)	3.93	3.63 %
Det + Poisson	det	3.42	2.51 %
	exp	4.19	4.24 %
	H_2 ($p_1 = 0.1$)	4.41	4.75 %
Poisson	any dist	4.84	5.80 %

Table 3.1: The fraction of refused admissions for (mixed) stationary arrivals.

In some scenarios, patients visit a number of successive wards before leaving the clinic. Heavy-traffic approximations for such networks are complicated, see [**52, 107**] for some extensions to networks. For some cases, the variability of downstream wards can easily be identified assuming sufficient capacity. For Poisson arrivals (unscheduled admissions) the number of patients in each node of the network has a Poisson distribution [**79**]. Furthermore, in case the LOS of the preceding wards are deterministic (e.g. pre-surgery admissions), the variability in admissions of the downstream ward inherits the variability of the original arrival process.

To illustrate the impact of a very regular admission schedule on downstream wards, we consider a specific tandem of two infinite server queues. We only consider deterministic external arrivals that arrive at queue 1 with an average of 41/7 arrivals per day. Each patient moves from queue 1 to queue 2 after which he/she leaves the network. We focus on queue 2 and choose an average length of stay (ALOS) of 4 for this queue leading to an average offered load of 23.43, such that the results for queue 2 may be compared to those of Table 3.1. The standard deviations of the offered load for queue 2 can be found in Table 3.2, which were determined using simulations. Note again that Poisson arrivals would lead to a Poisson number of patients present, yielding a standard deviation of 4.84.

	Queue 1	Queue 2 LOS distribution		
ALOS	LOS distribution	det	exp	H_2 ($p_1 = 0.1$)
1	exp	2.40	3.74	4.18
	H_2 ($p_1 = 0.1$)	2.33	3.75	4.16
3	exp	3.62	4.11	4.38
	H_2 ($p_1 = 0.1$)	3.31	4.04	4.35
5	exp	4.01	4.27	4.61
	H_2 ($p_1 = 0.1$)	3.76	4.16	4.47
1,3,5	det	0	3.42	3.93

Table 3.2: Standard deviation offered load to queue 2.

Clearly, the variability in offered load for queue 2 is considerably larger than for queue 1, except for deterministic ALOS at queue 1. The results show that the standard deviation in offered load increases with both the ALOS of queue 1 and the squared coefficient of variation of the service times of queue 2. It is interesting to note that a more variable service time at queue 1 compared to an exponential distribution, i.e. H_2 ($p_1 = 0.1$), may reduce the variability in offered load at queue 2, in particular as the ALOS gets larger. More practically, we see that the impact of a regular admission schedule rapidly fades out for wards further down the health chain unless the length of stay is more or less fixed for the upstream wards.

3.2 Impact of time-dependent admissions

In this section, we assume that the arrival process at a ward depends on the day of the week. This case is of particular interest in view of the schedule of elective patients at the OT. For example, it is well known that the number of arrivals is generally smaller during the weekend than on weekdays since hardly any elective procedures are scheduled during the weekend, see e.g. **[25, 54]**. The assignment of OT sessions to surgical disciplines typically also leads to differences in the number of arrivals.

We assume that there is a periodic (cyclic) arrival pattern. Let T be the length of a cycle and denote the average number of arrivals during $[a, b]$ by $\lambda(a, b)$, $a \leq b$. We are mainly interested in the weekly pattern, i.e., $T = 7$. Let $\lambda_d = \lambda(d-1, d)$ denote the average number of arrivals at day d, $d = 1, \ldots, 7$, where we denote Monday by day 1. Also, let $\bar{\lambda}$ be the average number of arrivals per day. As in Section 3.1, we assume that the capacity is fixed at s operational beds.

Again, we use the infinite server queue (G/G/∞) as a basis for approximating the number of occupied hospital beds. In particular, the mean number of occupied beds is $\rho = \bar{\lambda}\beta$ and the variance (in heavy-traffic) is $z\bar{\lambda}$, where z is called the peakedness reflecting the variability in arrival and LOS processes. Similar to **[65]**, we assume that the variability in arrivals consists of a random and predictable part. We decompose the peakedness into a random and predictable part as well, yielding

$$\begin{aligned} z &= z_{\text{rand}} + z_{\text{pred}} \\ &= 1 + (c_a^2 - 1)(1 - G) + z_{\text{pred}}, \end{aligned} \tag{3.3}$$

where the first part (z_{rand}) is the same as in Section 3.1. We note that the fraction of refused admissions can be determined again using the Hayward approximation, see Section 3.1. It easily follows that the loss fraction becomes larger for time-dependent arrivals (compared to a stationary arrival process) due to the increased peakedness.

We determine the second part (z_{pred}) of (3.3) based on a deterministic fluid approximation, see **[79]**. Specifically, let $z_{\text{pred}} = \mathbb{V}\text{ar}[m(t)]/\mathbb{E}[m(t)]$ with $m(t)$ the mean number of busy servers at time t in the G/G/∞ queue:

$$m(t) = \mathbb{E}\left[\int_{t-S}^{t} \lambda(s)\mathrm{d}s\right] = \int_0^\infty \lambda(t-s)\mathbb{P}(S > s)\mathrm{d}s, \tag{3.4}$$

where $\lambda(s)$ is the arrival rate at time s (see e.g. **[79]**). From (3.4) we see that the mean number of occupied beds depends on the full distribution of S, the LOS. That is, the ALOS or first two moments of the LOS distribution are not sufficient to determine the mean number of occupied beds at a particular point in time.

Of prime interest in the present setting is the case $T = 7$ with arrival rates λ_d, $d = 1, \ldots, 7$ and an exponential (or hyper-exponential) LOS distribution. The case of an exponential LOS distribution can be used as a

building block for more involved service time distributions and (feed forward) networks, see Appendix 3.A.

Exponential LOS

For later use, we indicate $m^{\exp}(t)$ for the mean number of busy servers in case of exponential service times. For convenience, consider the time instants $d = 1, 2, \ldots, 7$ corresponding to the end of each day. Then, using (3.4), we directly obtain the recursive relation, for $d \in \mathbb{N}$,

$$\begin{aligned} m^{\exp}(d) &= \int_{s=0}^{1} \lambda_d e^{-\mu s} \mathrm{d}s + \int_{1}^{\infty} \lambda(d-s) e^{-\mu s} \mathrm{d}s \\ &= \frac{\lambda_d}{\mu}\left(1 - e^{-\mu}\right) + e^{-\mu} \int_{0}^{\infty} \lambda(d-1-u) e^{-\mu u} \mathrm{d}u \\ &= \frac{\lambda_d}{\mu}\left(1 - e^{-\mu}\right) + e^{-\mu} m^{\exp}(d-1), \end{aligned} \tag{3.5}$$

where the final step follows from (3.4), see also [**13**].

Using the above relation n times, we have

$$m^{\exp}(d) = \frac{1}{\mu}\left(1 - e^{-\mu}\right) \sum_{i=0}^{n-1} \lambda_{d-i} e^{-\mu i} + e^{-\mu n} m^{\exp}(d-n).$$

Taking $n = T$ and using the periodicity of the arrival rate and, hence, $m^{\exp}(d) = m^{\exp}(d-T)$, yields

$$m^{\exp}(d) = \frac{1}{\mu} \frac{1 - e^{-\mu}}{1 - e^{-T\mu}} \sum_{i=0}^{T-1} \lambda_{d-i} e^{-\mu i}. \tag{3.6}$$

LOS as sum of exponentials

Here, we assume that the LOS can be expressed as a sum of exponential terms: $\mathbb{P}(S > t) = \sum_{j=1}^{J} p_j e^{-\mu_j t}$. This directly applies to hyper- and hypoexponential LOS distributions. For the former, we have $0 < p_j < 1$ and $\sum_{j=1}^{J} p_j = 1$, whereas the tail distribution for the latter is given by (3.17). Note that these cases may be equivalently interpreted as parallel and tandem networks where the LOS in each node is exponentially distributed. The hypoexponential case may be primarily applied for modelling series of subsequent wards, whereas the hyper-exponential distribution often

provides a better fit for the LOS distribution compared to the exponential.

For the mean number of occupied beds, we have

$$m(t) \;=\; \int_{v=0}^{\infty} \lambda(t-v) \left(\sum_{j=1}^{J} p_j e^{-\mu_j v} \right) \mathrm{d}v \;=\; \sum_{j=1}^{J} p_j m_j^{\exp}(t).$$

Now, the predictable variation in the number of occupied beds is approximated by

$$z_{\mathrm{pred}} = \frac{1}{T-1} \sum_{d=1}^{T} \left(m(d) - \bar{m} \right)^2 / \bar{m}, \tag{3.7}$$

where $\bar{m} = \sum_{d=1}^{T} m(d)/n = \lambda\beta$ is the average occupancy with $m(d)$ given by (3.6). We note that this may seem involved at first glance, but z_{pred} may easily be computed in e.g. a spreadsheet. Moreover, this derivation provides the basis for scheduling elective admissions as presented in Section 3.3.

3.2.1. Remark. A different approach in case of time-dependent arrivals is the stationary process approximation, see **[80]**. The main idea of that approach is to capture the additional variability in the arrival process in c_a^2, i.e., the time-dependent process is approximated by a stationary process that is more variable. The disadvantage of this approach is that the impact of the service time (LOS) distribution on non-stationary arrivals cannot always be properly taken into account.

To verify the modified peakedness approximation (3.3) numerically, we consider the following modification of the clinical ward introduced in Section 3.1. The average number of arrivals during weekdays and during the weekend is assumed to be 7 and 3, respectively. The ALOS is 4 days again, yielding an average load of roughly 23.43. For now, we assume the number of operational beds fixed at 28. In Table 3.3, we present approximation and simulation results for the standard deviation in offered load and the fraction of refused admissions for different LOS distributions and for both a deterministic and a Poisson arrival process. As can be observed from Table 3.3, the approximations are quite similar to the simulation results, indicating that the modified peakedness approximation (3.3) works well.

		Stdev. offered load		Loss fraction	
Arrivals	LOS	Approx	Simulation	Approx	Simulation
Det	det	3.60	3.21	2.89 %	2.44 %
	exp	3.91	3.80	3.58 %	3.36 %
	H_2 ($p_1 = 0.1$)	4.31	4.26	4.52 %	4.10 %
Det + Poisson	det	5.00	4.66	6.18 %	6.09 %
	exp	4.61	4.54	5.24 %	5.22 %
	H_2 ($p_1 = 0.1$)	4.76	4.63	5.61 %	5.52 %
Poisson	det	6.03	5.76	8.76 %	9.6 %
	exp	5.20	5.11	6.67 %	6.75 %
	H_2 ($p_1 = 0.1$)	5.15	5.05	6.57 %	6.71 %

Table 3.3: The fraction of refused admissions for time-dependent arrivals.

Clearly, the weekly arrival pattern leads to increased variability in offered load and refused admissions compared to stationary arrivals (Table 3.1). This weekly pattern is most prominent for a deterministic LOS [**13, 33**]. As a consequence, the impact of the variability in LOS distribution on the offered load can go either way depending on the type of arrival process. This further strengthens the conclusion from Section 3.1 that the arrival process is of primary importance for a stable bed occupancy and should be considered first before focusing on the variability in LOS.

Health chains

In Section 3.1 we illustrated that the benefits of a relatively stable arrival process to the first ward rapidly fades out for downstream wards due to the variability in LOS (except for deterministic LOS). Here we give an example of the opposite effect in case of time-dependent arrivals. In particular, we consider two wards in tandem with sufficient capacity (infinite number of servers) in which the arrival process to the first ward is as described above. The LOS at both wards is exponentially distributed, where the LOS at ward 2 equals 4. The weekly pattern in average offered load can be found in Figure 3.1 for an ALOS of 1, 3 and 5 days at ward 1. We also included the case in which the LOS at ward 1 equals 0, meaning that

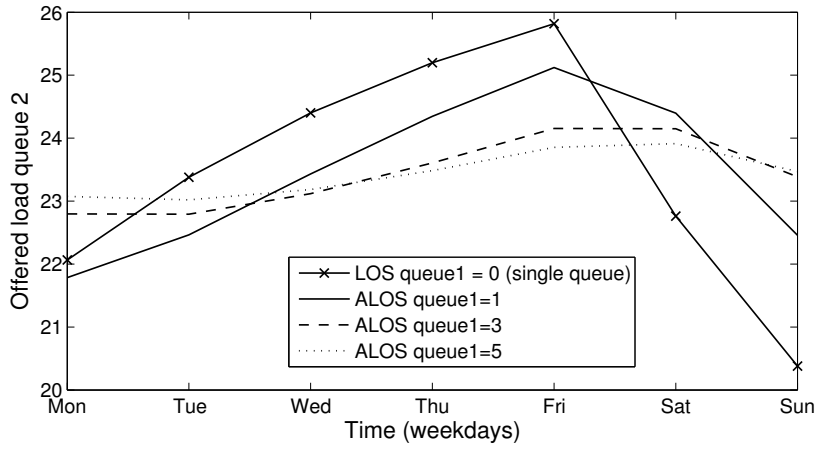

Figure 3.1: The average offered load across the week for ward 2 in a tandem with different ALOS at ward 1.

there effectively is a single ward. The different ALOS of ward 1 has the following implications for ward 2: (i) the peak in offered load shifts due to the LOS at ward 1, and (ii) the difference in average offered across the week becomes smaller as the ALOS at ward 1 increases. Combined with the observations from Section 3.1, we note that, for non-deterministic LOS, the arrival process to downstream wards tend to look more like a stationary Poisson process.

3.3 Scheduling elective admissions

In most hospitals patient admission scheduling is done per medical discipline and independently of possible effects on the bed occupancy at clinical wards or intensive care units. As indicated, this often leads to high variability in bed occupancy and weekly patterns in the number of patients present at wards. The latter is also caused by the reduced number of (elective) admissions during the weekend (see Section 3.2). One way to deal with this weekly pattern is to adapt the staffing according to the offered load as in, e.g., [**13**] or [**41**]. A different approach is to schedule

admissions such that undesired predictable fluctuations in the bed occupancy are avoided as much as possible. In this section, we propose a quantitative method for the latter option.

Specifically, the scheduling of elective admissions is done in two steps:

Step 1: Determine target load for each day, $m^*(d), d = 1, \ldots, T$.

Step 2: Determine an admission schedule such that the difference between the offered load and target load is minimised, using an optimisation model.

We note that Sections 3.1 and 3.2 play a key role in Steps 1 and 2, respectively. Here, we restrict ourselves to a single ward and K types of patients. This may, for instance, represent a ward for one medical discipline with various procedures leading to structural differences in the length of stay. This is just a base example and the model may be extended in various directions along the same lines. For implementation purposes, focusing on a single medical discipline may be a good starting point, as admissions are now generally scheduled per medical discipline and coordination between disciplines is not yet common. Moreover, in case the offered loads of all disciplines are well balanced, this immediately holds for the overall offered load.

We assume that the length of stay of patient type k is exponentially distributed with ALOS $1/\mu^k$. (Here we use exponential LOS, for hypo- or hyper-exponential LOS one ward is represented by more than one node.) Let the average number of admissions on day d of type k be λ_d^k and let the offered load of patients of type k on day d be $m^k(d)$. The target number of admissions for type k during T days is denoted by Λ^k.

Step 1: Target load

Determining the target load mainly concerns a managerial decision at a tactical level. It involves two parts: (i) The capacity in relation to variability in offered load, and (ii) the weekly pattern for available number of beds. Regarding (i), the models discussed in Section 3.1 can be applied to support decisions related to the trade-off between occupancy levels and blocking probabilities. More specifically, the average load per day m^* may be determined using (3.2). For instance, for a given throughput the required capacity may be determined such that the blocking probability

does not exceed some target. Alternatively, given a fixed capacity, a target occupancy level may be determined such that the blocking probability does not exceed some chosen value.

The number of available beds depends on the staffing, which is not necessarily the same for every day of the week. A typical example for (ii) is a different staffing during weekdays compared to the weekend, which generally means that during the weekend some beds are closed due to reduced bed demand (that is a consequence of the limited number of scheduled admissions). Denote the target load on day d by $m^*(d)$, $d = 1, \ldots, T$. Clearly, it should hold that $m^* = \sum_d m^*(d)/T$. The target load during a cycle should also be equal to the offered load following from the admission target and corresponding ALOS, i.e., $m^*(d)$ should satisfy

$$m^* = \frac{1}{T}\sum_{d=1}^{T} m^*(d) = \frac{1}{T}\sum_{k=1}^{K} \Lambda^k \times \frac{1}{\mu^k}.$$

For identical targets on all days, we evidently have $m^*(d) \equiv m^*$. In case the number of open beds during the weekend is reduced by x (assuming that T is a multiple of 7), it follows after some straightforward calculations that

$$\begin{aligned} m^*(d) &= m^* + \frac{2}{7}x \qquad \text{for } d \text{ multiple of } 1, \ldots, 5 \\ m^*(d) &= m^* - \frac{5}{7}x \qquad \text{for } d \text{ multiple of } 6, 7 \end{aligned}$$

Step 2: Optimal admission schedule
In this step, we translate the admission scheduling into a mathematical model, using results from Section 3.2. Specifically, we formulate the problem as a Quadratic Programming model with linear constraints, which is in the spirit of [**1**]. (We note that it can be formulated as a Linear Program as well using a different objective function.) The key element is that the time-dependent offered load as determined in Section 3.2 is linear in λ_d.

The objective here is to minimise the total squared deviation of the offered load from the target load. This is represented in (3.8) where the squared deviation between the target and offered load is summed over all days of the planning cycle.

The offered load for each patient type for the first (3.11) and all consecutive days (3.12) of the planning cycle are derived from the time-dependent analysis in Section 3.2, i.e., Equation (3.6) for exponential LOS.

Note again that the full LOS distribution is required to determine the mean offered loads, and not just the average LOS. The total offered load on a particular day is the sum of the loads generated by the different patient types, as can be seen in (3.10).

The constraint (3.9) assures that the total number of scheduled admissions is equal to the target number of admissions for each patient type. The number of admissions on each day should be non-negative, as represented by (3.13). Moreover, it might be desirable or current practice that no scheduled admissions occur on some days, for instance, during the weekends or on days when no OT-time is available for a certain patient type. For such days, λ_d^k should be set to 0, as represented in (3.14).

Finally, we note that the choice of decision variables depends on whether patients of type k, $k = 1, \ldots, K$, represent scheduled or unscheduled admissions. In case type k patients are scheduled, then λ_d^k, $d = 1, \ldots, T$ are decision variables, whereas the λ_d^k should be determined from historical data in case of emergency admissions. (For the latter, $m^k(d)$ can also be determined directly from the data.)

$$\text{Minimise} \quad \sum_{d=1}^{T} [m(d) - m^*(d)]^2 \tag{3.8}$$

$$\text{subject to} \quad \sum_{d=1}^{T} \lambda_d^k = \Lambda^k, \qquad k = 1, \ldots, K \tag{3.9}$$

$$m(d) = \sum_{k=1}^{K} m^k(d), \qquad d = 1, \ldots, T \tag{3.10}$$

$$m^k(1) = \frac{1}{\mu^k} \frac{1 - e^{-\mu^k}}{1 - e^{-T\mu^k}} \sum_{i=0}^{T-1} \lambda_{1-i}^k e^{-\mu^k i}, \qquad k = 1, \ldots, K \tag{3.11}$$

$$m^k(d) = \frac{\lambda_d^k}{\mu^k} \left(1 - e^{-\mu^k}\right) + e^{-\mu^k} m^k(d-1), \quad d = 2, \ldots, T, \quad k = 1, \ldots, K \tag{3.12}$$

$$\lambda_d^k \geq 0, \qquad d = 1, \ldots, T, \quad k = 1, \ldots, K \tag{3.13}$$

$$\textit{(case-dependent)} \ \lambda_d^k = 0, \qquad \text{for some} \quad d \in \{1, \ldots, T\}, \quad k \in \{1, \ldots, K\} \tag{3.14}$$

As mentioned, the admissions scheduling can be modeled as a Linear Programming problem by modifying the objective function. In that case, the objective (3.8) is to minimise $\sum_{d=1}^{T} |m(d) - m^*(d)|$, which can be made linear using standard LP arguments. Here we opt for a quadratic objective function because we assume that the consequences of a deviation from the target load will not be linear in the size of the deviation. It is considerably more difficult for the medical staff to handle larger deviations.

Extensions and modifications

The QP model as introduced above is an elementary model that may be extended in different directions depending on the specific situation. Two important extensions are multiple (consecutive) wards and the impact of the Operating Theater for surgical patients. These extensions are discussed below.

A prime example where multiple wards are involved concerns medical disciplines for which a considerable fraction of the patients needs care

at an ICU, after which they join the Normal Care Unit. The development of clinical pathways has also increased the interests in health chains. The time-dependent performance of health chains may again be found in Section 3.2 and Appendix3.A, which is one of the key elements to extent the QP model, i.e., extend (3.11) and (3.12). Moreover, the objective function should then be modified such that the sum of the deviations from the target load of each ward is minimised. Depending on its type, different wards may be assigned a different weight to represent its relative importance. For example, the weight for an ICU will typically be larger than the weight for other wards, as ICU capacity is more costly and the options for the transfer of patients in case of insufficient capacity are limited.

For surgical patients, the number of admissions is restricted due to the schedule of the OT. In general, each surgical discipline is assigned one or more rooms for some specific days, i.e., the OT sessions. For a given OT schedule, the maximum number of admissions of type k on some day d thus depends on the surgical time of type k patients and the available OT time for the medical discipline of type k. Such restrictions can be straightforwardly included in the QP and thus easily allow for modifications in the admission planning without (strongly) affecting the OT schedule. Finally, we like to emphasise that the admission scheduling applies to both surgical and non-surgical patients.

We will describe some numerical experiments to illustrate the process outlined above. The scenario for these experiments is comparable to the scenarios considered for the analysis of stationary and time-dependent admissions. Specifically, emergency patients arrive with an average of 3 per day and have an ALOS of 4 days. For the elective patients, we assume that two groups can be distinguished: patients with short (ALOS of 2 days) and long (ALOS of 6 days) hospital stay. These groups can, for instance, be determined based on medical procedures. The target number of admissions for both groups is 10 patients per week. For simplicity, the length of stay is exponentially distributed for all groups.

We consider three different target scenarios: no reduction of beds during the weekend and closing 2 and 4 beds during the weekend. For all scenarios, no elective admissions during the weekend are allowed. The required number of elective admissions that follow from solving the QP are given in Table 3.4. The resulting offered loads, along with the targets, are displayed in Figure 3.2. We note that the presented numbers of admissions in Table 3.4 are fractional. To find the admission quota, these

numbers could be rounded to the nearest integer. If it is infeasible to guarantee identical number of elective admissions for each patient group for a considerable time period, the "admission planner" could work with a small bandwidth. In practical situations the fractional numbers therefore provide a guideline in which direction the actual numbers of admissions should deviate from the prescribed admission quota.

Weekend reduction		Mon	Tue	Wed	Thu	Fri	Sat	Sun
No weekend reduction	short	5.2	3.6	1.2	0	0	0	0
	long	0	0	2.2	3.0	4.7	0	0
Reduction of 2 beds	short	6.2	3.8	0	0	0	0	0
	long	0	0.1	3.6	3.0	3.3	0	0
Reduction of 4 beds	short	4.0	3.2	0.5	1.0	1.1	0	0
	long	3.3	0.3	2.9	2.1	1.4	0	0

Table 3.4: Elective admission quota for different bed occupancy targets.

Observe that in all scenarios, the target and offered load are not identical for all days of the week. Because there are no admissions during the weekend, there is limited control over the offered load during that period. For instance, in all cases the load decreases considerably from Saturday to Sunday. To compensate for the relatively small load on Sunday, the number of admissions is largest on Mondays for all three scenarios. In case the reduction in beds is limited (here 0 or 2), the patients with relatively long hospital stay should be admitted at the end of the week, often on Fridays, thereby filling the beds during the weekend. As a consequence, the patients with short hospital stay are mainly admitted at the beginning of the week, often on Monday.

We note that aiming for a constant bed occupancy target might be undesirable in this example. For the scenario of no weekend reduction, the offered load has a peak on Friday that is implicitly caused by the relatively large target during the weekend. In case 2 beds are closed during the weekend, there remains a smaller peak in offered load on Friday, whereas this peak is absent in the scenario where 4 beds are closed. Although we presented a specific case, the admission principles apply to a broader health care setting.

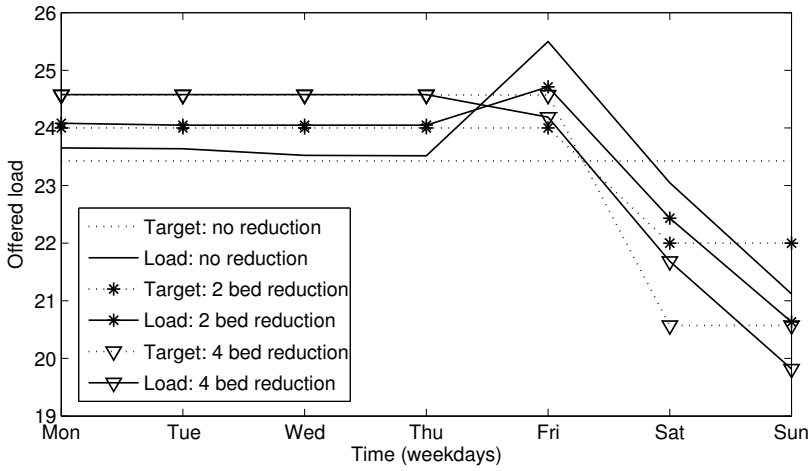

Figure 3.2: The target load and offered load across the week for different target scenarios.

3.4 Practical implications and discussion

The main goal of this chapter is to provide quantitative methods to determine admission quota for scheduled admissions and to analyse the impact of variability in scheduled admissions on the required bed capacity. For the impact of variability, we used approximation methods that build on heavy-traffic results in the literature and presented an interesting relation to the Gini coefficient. Moreover, we modified this peakedness approximation to allow for time-dependent arrivals, which is exploited in the step of admission scheduling. In particular, the admission quota for scheduled patients are determined using a QP model minimising the difference between the expected and desired occupancy.

Our second aim is to derive generic practical insights that apply to almost all hospital situations. A first major observation is that more variation in admissions leads to a higher variability in bed demand and to more refused admissions for a hospital ward. Variation in the LOS can have negative consequences as well, but its influence depends on the variability of the arrivals. Only for stable arrival processes reducing the variation in LOS leads to a less variable bed occupancy. Hence, stabilising the bed oc-

cupancy is best achieved by starting to smoothen the admissions. Along the same lines, the most time-stable performance is achieved when the arrivals to the hospital are as evenly distributed over the week as possible. A very uneven weekly pattern will increase variability in bed demand and the probability of refused admissions just as a variable arrival process will. If there is a clear weekly pattern, a LOS that is very stable can even be detrimental. Here, again, one should start by smoothing the admission pattern.

In practice, patients often visit more than one type of ward during their stay in the hospital.The variation in demand at the first ward influences that on subsequent wards. For relatively stable admissions, the variability in bed demand on the second ward is at least as large as that on the first one. Typically, for subsequent wards the bed occupancy starts to look more like the occupancy generally seen for emergency patients. In situations with a weekly admission pattern, a peak in demand on the first ward will be noticeable for the second ward as well, but with a shift in the time on which it occurs. The weekly pattern on the second ward becomes less noticeable as the LOS in the first ward becomes more variable.

In addition to smoothing the arrival process, it is also possible to schedule the arrivals in a better way. The first decision needed is the number of beds that will be staffed every weekday, e.g., how many beds are closed during the weekend. In general, in absence of scheduled admissions during the weekend, it is advisable to close beds during that period. The case where no beds are closed during the weekend and scheduled admissions are absent might lead to unused capacity. Scheduling patients with a longer expected LOS on Fridays can help to minimise this unused capacity, as such patients will stay throughout the weekend. The optimal schedules generated by our optimisation model typically show such patterns (although the pattern is clearly affected by the number of closed beds during weekends). The drawback is that this often results in peak demand on Friday itself.

Another general rule is that more admissions should be scheduled on Mondays compared to the other weekdays, to fill the ward after the weekend. Because patients with a longer LOS are mainly scheduled at the end of the week, the patients scheduled for Mondays typically have a shorter LOS. Tuesday through Thursday are often comparable and roughly have about the same number of admissions scheduled.

After developing this method, we worked with a hospital to implement the principles in practice. A few notable points we want to make here concern the results and limitations of the method. First, we have noticed that the method can lead to good results in terms of either less bed usage or increasing the number of patients treated without extra resources. The possible bed reduction was in the order of 10 percent for most specialties. We also noticed that having enough patients on a waiting list is key to successful implementation, so coordination with clinic sessions is necessary to influence and smoothen the numbers and types of referrals for surgery and thus admission. A last notable point that even when restrictive constraints were needed in the optimisation part, improvements were visible just from adhering to a schedule even though the schedule itself could have been better. This means that part of the gains are due to reducing the variation in arrivals, apart from scheduling the optimal number of arrivals.

We like to stress that the models presented are of a generic nature and can easily be implemented in e.g. an Excel spreadsheet to model the characteristics of a specific ward or hospital (or a specific time scale). Such models are of a deductive nature, based on a set of general principles and logical inference to derive new insights or improve decision making, see also Gallivan [**48**] for a further discussion on the role of models in health care. By definition, these models are based on assumptions regarding the structural characteristics of patient flows and admissions and, therefore, do not capture all decisions made at a hospital. A particular topic that is not captured by the models presented here is the possible dependence of the discharge process on the day of week or the occupancy. Although it is not clear whether it is desirable to incorporate such dependencies in the structural organisation of health care processes, this provides an interesting topic for further research.

3.A Phase type LOS and feed forward networks

In this section, we consider a feed forward network of nodes with an exponential LOS. This may be equivalently interpreted as a single node with a (specific) phase type LOS distribution. For convenience, we assume that the LOS rates are different for each node.

First, we define some notation, in line with Section 3.2, and restate part of a more general result of [**79**]. Let J be the number of nodes and let $\lambda_j(t)$ be the external arrival rate to node j at time t. A patient goes from node i to node j with probability p_{ij} and leaves the network from node i with probability $1 - \sum_{j \neq i} p_{ij}$. Denote a generic LOS at node j by S_j and let μ_j be its LOS rate.

The main goal is to determine the mean number of occupied beds at time t for node j ($m_j(t)$). We first restate (part of) a more general result of Massey and Whitt, see [**79**, Theorem 1.2]:

3.A.1. Theorem. *In the $(M_t/GI/\infty)^J)/M$ model, the number of occupied beds $Q_j(t)$ at time t, $1 \leq j \leq J$, are independent Poisson random variables with finite means*

$$m_j(t) = \mathbb{E}[Q_j(t)] = \mathbb{E}\left[\int_{t-S_j}^{t} \lambda_j^+(u)\mathrm{d}u\right], \tag{3.15}$$

where λ_j^+ is the aggregate-arrival-rate function to node j, defined as the minimal non-negative solution to the system of input equations, for $1 \leq j \leq J$,

$$\lambda_j^+(t) = \lambda_j(t) + \sum_{i=1}^{J} \mathbb{E}[\lambda_i^+(t - S_i)]p_{ij}. \tag{3.16}$$

For optimisation purposes and for applications in health care, we are interested in more explicit results. Therefore, we make several assumptions, while maintaining a sufficiently generic framework for modelling in practical situations. We assume that all S_j are exponential, i.e., we restrict ourselves to phase-type LOS distributions. For convenience, we also assume here that all μ_j's are different. Finally, we consider a feed forward network meaning that $p_{ij} = 0$ for $j \leq i$ and $1 \leq i \leq J$.

Below, we express $m_j(t)$ in terms of single nodes with exponential LOS, which are essentially used as building blocks. To do so, we decompose the patient flows into all possible routes through the network (that have non-zero probability). A patient on route r then uses a subset of the nodes $\{1, \ldots, J\}$. Specifically, patients on route $r = \{n_1, \ldots, n_f\}$ arrive at the first node n_1 with rate $\lambda_{n_1}(t)p_r$, where $p_r = p_{n_1 n_2} \cdots p_{n_{f-1} n_f}$ represents the fraction of traffic coming from node n_1 going through n_f via route r.

Now, consider node j and truncate the network at node j, i.e., consider the network consisting of nodes $\{1, \ldots, j\}$. Let r^j be a route in the truncated network that goes through node j and let R^j be the set of possible

routes going through j. We add a subscript s if route r^j starts at node s (we denote r_s^j and use R_s^j again to denote the set of possible routes). Using (3.15), (3.16) and the feed forward structure, we have

$$\begin{aligned} m_j(t) &= \mathbb{E}\left[\int_{t-S_j}^{t}\left(\lambda_j(u)+\sum_{i=1}^{j}\mathbb{E}[\lambda_i^+(u-S_i)]p_{ij}\right)\mathrm{d}u\right] \\ &= \mathbb{E}\left[\int_{t-S_j}^{t}\lambda_j(u)\mathrm{d}u\right]+\mathbb{E}\left[\int_{t-S_j}^{t}\sum_{i=1}^{j-1}p_{ij}\mathbb{E}[\lambda_i^+(u-S_i)]\mathrm{d}u\right] \\ &= m_j^{\exp}(t)+\mathbb{E}\left[\int_{t-S_j}^{t}\sum_{r_s^j\in R^j}p_{r_s^j}\mathbb{E}[\lambda_s(u-\sum_{l\in r_s^j}S_l)]\mathrm{d}u\right] \\ &= m_j^{\exp}(t)+\sum_{s=1}^{j-1}\sum_{r_s^j\in R_s^j}p_{r_s^j}\mathbb{E}\left[\int_{t-S_j}^{t}\lambda_s(u-\sum_{l\in r_s^j}S_l)\mathrm{d}u\right], \end{aligned}$$

where the final step follows from interchanging integrals and sum. Note that the value of the expectation is similar to the mean load in node j for a tandem network (or a single node with a hypoexponential LOS). Using the tail distribution of a hypoexponential random variable (3.17), we get, after some rewriting,

$$\begin{aligned} &\mathbb{E}\left[\int_{t-S_j}^{t}\lambda_s(u-\sum_{l\in r_s^j}S_l)\mathrm{d}u\right] \\ &= \mathbb{E}\left[\int_{t-(\sum_{l\in r_s^j}S_l+S_j)}^{t}\lambda_s(u)\mathrm{d}u\right]-\mathbb{E}\left[\int_{t-\sum_{l\in r_s^j}S_l}^{t}\lambda_s(u)\mathrm{d}u\right] \\ &= \prod_{l\in r_s^j-\{j\}}\frac{\mu_l}{\mu_l-\mu_j}m^{\exp}_{\{\lambda_s(\cdot),\mu_j\}}(t)+\sum_{i\in r_s^j-\{j\}}\frac{\mu_i}{\mu_j-\mu_i}\prod_{l\in r_s^j-\{j\}}\frac{\mu_l}{\mu_l-\mu_i}m^{\exp}_{\{\lambda_s(\cdot),\mu_i\}}(t), \end{aligned}$$

where $m^{\exp}_{\{\lambda_s(\cdot),\mu_j\}}(t)$ is the mean load at time t for a single node with exponential LOS at rate μ_j and arrival rate function $\lambda_s(\cdot)$. Combining the above yields

$$\begin{aligned} m_j(t) &= m_j^{\exp}(t) + \sum_{s=1}^{j-1} \sum_{r_s^j \in R_s^j} p_{r_s^j} \Bigg(\prod_{l \in r_s^j - \{j\}} \frac{\mu_l}{\mu_l - \mu_j} m^{\exp}_{\{\lambda_s(\cdot),\mu_j\}}(t) \\ &\quad + \sum_{i \in r_s^j - \{j\}} \frac{\mu_i}{\mu_j - \mu_i} \prod_{l \in r_s^j - \{j\}} \frac{\mu_l}{\mu_l - \mu_i} m^{\exp}_{\{\lambda_s(\cdot),\mu_i\}}(t) \Bigg). \end{aligned}$$

Again, the mean offered load at some node may seem involved at first glance, but we can still express it in terms of single exponential nodes. For a feed forward network of size J we require the time-dependent analysis of at most $J!$ exponential single nodes (for $j = 1, \ldots, J$, we need $m^{\exp}_{\{\lambda_s(\cdot),\mu_j\}}(\cdot)$, with $s = 1, \ldots, j$). However, the actual required number of single exponential nodes strongly depends on the routing probabilities in a specific practical situation and will often be much smaller than $J!$.

3.A.2. Example. An important special case is a tandem network of J nodes in series. Assuming here that all customers arrive at the first node, the sojourn time is then the convolution of J exponentials, which has a hypoexponential distribution, i.e.,

$$\mathbb{P}(S > t) = \sum_{j=1}^{J} \prod_{n \neq j} \frac{\mu_n}{\mu_n - \mu_j} e^{-\mu_j t}. \tag{3.17}$$

In this case, the mean number of occupied beds in node j reads

$$m_j(t) = \prod_{l=1}^{j-1} \frac{\mu_l}{\mu_l - \mu_j} m_j^{\exp}(t) + \sum_{i=1}^{j-1} \frac{\mu_i}{\mu_j - \mu_i} \prod_{l \neq i}^{j-1} \frac{\mu_l}{\mu_l - \mu_i} m_i^{\exp}(t).$$

4 Admission control for health care

Demand for health care keeps growing all the time, while budgets are being cut. This is not only the case in hospitals, but in all other types of facilities as well. This makes efficient use of available personnel very important, as personnel usually takes up a very large part of the expenses in health care. One of the ways to improve efficient and effective use of personnel, is to prioritise the entrance of patients into the facility in the right way. In this chapter we study the problem of admission control in home care and for rehabilitation facilities.

Advances in health care have led to an increasing number of elderly people in society who want to continue living in their own homes while needing medical care and care-at-home services (e.g., housekeeping and personal care). This trend has led to a situation in which home care providers are faced with a larger number of patients. At the same time, home care providers have to provide care-at-home services with fewer resources because of changing organisation and finance structures and increased competition. Therefore, efficient workforce management is essential to provide a high quality of service against low operational costs.

In practice, efficient workforce management is hard to achieve. The care-at-home sector typically has a very unpredictable demand for service. Moreover, the duration of the service is highly volatile. This creates a tension between the size of the workforce and the operational costs. On the one hand, having a lot of home care personnel leads to a very good service quality: all demand can be satisfied directly and no patient is turned down. However, the operational costs are high and much of the personnel will have a lot of idle time and low productivity. On the other hand, having too few personnel leads to low operational costs, but also a deterioration in quality of service.

Rehabilitation care facilities face similar problems, though for slightly different reasons. Over the last few years there have been quite a few changes in the organisation of rehabilitation care, and some changes are still to come. The largest of these changes concerns the structuring of the financial revenues rehabilitation facilities receive from the Dutch government and insurance agencies. In the past rehabilitation facilities were reimbursed for every treatment they administered to patients, but in the new system

patients are all diagnosed according to a fixed classification, and for every patient diagnosis the facility receives a fixed amount of money. This change of course leads to a necessity for rehabilitation facilities to work in an efficient way, while still providing good quality care. This system is not unique to rehabilitation care; the same system was also introduced in hospitals recently.

Again the largest part of the cost in rehabilitation consists of personnel cost. Almost all rehabilitation patients need a number of specialties for their treatment, for example a physiotherapist, a social worker and a doctor. The treatment of the patient by all these different groups of personnel needs to be coordinated for the treatment to be effective. There are many different groups of patients, all with their own need for the types and intensity of treatment and with different lengths of stay in the system. This means that deciding upon a good mix of personnel specialties is not an easy task.

On top of that there are also constraints on the waiting times that are deemed acceptable for patients. These can differ per group of patients, from a few days if the patients is currently staying in the hospital and the hospital bed is needed, to several weeks for less urgent patient groups. A patient can only enter the system if there is capacity available for all specialties he needs for his treatment. If one is not available the patient has to wait, and personnel from the other specialties experience some idle time, or may start treating another patient for whom all capacity is available. All of this leads to the problem of prioritising the patients for admission to the facility, so as to balance the waiting times of the patients already present and the waiting times of those arriving in the future with the efficient use of personnel.

The personnel planning problem is not unique to the health care sector. Many other service providers are faced with this challenging problem, e.g., call centers (Aksin et al. [**2**] and Gans et al. [**50**]) and public transport (Petrovic and Berghe [**90**]). For the problem in health care see Burke and Petrovic [**26**]. Ernst et al. [**39**] present a comprehensive collection of some 700 papers on personnel scheduling in different application areas. These surveys show that most of the literature on personnel planning in health care systems deal with appointment scheduling, shift scheduling, cyclic rostering, hospital admission and bed planning. This is mostly done in a deterministic setting for which mathematical and integer pro-

gramming methods, set covering and partitioning, local and tabu search techniques are used. The papers that deal with stochastic health care systems (mostly, appointment scheduling and hospital planning) use local search, genetic programming, simulation techniques, Markov decision theory, and queueing theory (see, e.g., the survey paper [**39**] and the special issues [**26, 90**] with the references therein).

The care-at-home sector has received little attention in the literature. Moreover, it faces challenges that the other industries do not have. First, patients have very specialised needs for home care so that home care providers are faced with a large number of very different patient and service profiles. Second, a patient may require a number of hours of home care in a week, but needs to receive that for several weeks consecutively. Hence, enough personnel capacity has to be available so that a patient will continue to receive home care once admitted. These two distinguishing features in a stochastic setting add additional complexity to the personnel planning problem, which makes many of the modelling and solution techniques intractable.

The literature on workforce management in a home care personnel planning setting can be categorised into two groups: the first group describes the imminent shortage of skilled nurses and other health workers in the coming years due to the ageing population, and the important factors for organisations in attracting new staff and retaining their current staff. Ellenbecker [**38**] mentions having a realistic workload and a stable schedule as important factors in keeping personnel retention at low levels. Flynn and Deatrick [**43**] also mention having a realistic workload, adequate staffing levels, and scheduled days off as important factors for the nurses. The second group of articles focuses on daily scheduling and routing of nurses. Cheng [**30**] models this problem as a vehicle routing problem with time windows using a mixed integer program. Bertels and Fahle [**17**] study the rostering and routing problem simultaneously. They choose not to obtain optimal solutions, because of the large computation times. Instead, they present several good solutions, by modelling different requirements by hard and soft constraints. Eveborn et al. [**40**] describe a decision support system for planning home care routes. This lets the user attach priorities to different aspects of the solution, such as travel times and preferred staff members to visit certain patients. The literature on home care personnel planning is largely deterministic in nature and does

not deal with the stochastic nature of the demand and service required that is perceived in practice.

Rehabilitation care also has not attracted much attention in the literature. There are a few studies on scheduling the treatment of patients inside a facility, focusing on the scheduling of appointments of patients with service providers. One of these is the work of Ogulata et al. [**86**], who propose a three-step model for the highly demanded physiotherapy, consisting of first selecting which patients to treat, then balancing the workload of the patients among the therapists and finally scheduling appointments over the working days. The work of Chien et al. [**31**] presents an approach to schedule different treatments of patients with different service providers, where they take into account any precedence constraints between these treatments. They model the problem as a hybrid job shop scheduling problem and solve it using a genetic algorithm. A model for balancing resource allocation among different types of patients has been studied not for rehabilitation but for care to the mentally handicapped by Heiner et al. [**61**]. They present a model for deciding how much of any resources to allocate to the different groups, while maximising the efficiency and effectiveness of care and also taking the fairness of the allocation into account.

The problem at hand here is an admission control problem. This problem arises in other areas than health care as well, for example in call centres. Models have been developed to deal with this problem: see, e.g., Altman [**3**] for examples in telecommunications, Gans et al. [**50**] in call centres, etc. The problem here differs from the models in the literature in the sense that patients need more than one type of server, i.e. specialties, simultaneously. In other settings there might be different customer types and several types of servers that can serve each customer type, but each customer needs only one server. This means that the problem of admission control for the rehabilitation setting does not fit into these models.

To our best knowledge no attention so far has been given in the literature to prioritising admissions to health facilities. A reason for the scarcity of literature in this area might be the complexity of the problem. The admission control problem can be modelled in e.g. the framework of Markov decision problems (see Puterman [**93**]), but suffers at the same time from a high-dimensional state space prohibiting the derivation of optimal solutions for already moderately-sized problem instances. This so-called curse of dimensionality can be addressed by applying approximation techniques, such as Approximate Dynamic Programming, that

lead to near-optimal solutions (see, e.g., Bertsekas and Tsitsiklis [**18**] and Powell [**92**]). The key idea of these techniques is to reduce the state space by replacing an exact step in the algorithm by an approximation, such as simulation, stochastic approximation or statistical learning. These approximations work fairly well when the problem under study has nice structural properties related to monotonicity. However, the personnel planning problem that we study lacks this feature as was shown in a special case of our model in Miller [**84**], Ross and Tsang [**95**], and Altman et al. [**4**].

In this chapter, we aim to provide a model to deal with the patient prioritisation problem in care-at-home and rehabilitation facilities in a stochastic setting. We assume that patients arrive according to a Poisson process and that they can differ in the length, types and intensity of care they require. We cast the problem as a Markov decision problem as this is sufficiently flexible to deal with different patient and service profiles in a decision framework. The objective in this system is to schedule the different types of patients such that a weighted waiting time is minimised.

For the somewhat simpler problem of home care the model still remains sufficiently tractable (see Powell [**92**] for issues on modelling and computation). We study the monotonicity properties of the Markov decision model. These results are used to derive optimal patient admission policies, given the demand for service. Moreover, we study the performance of these policies so that the size of the workforce can be determined.

The general model, used for the rehabilitation setting, is computationally intractable, since the state space is too large due to the need to store information on all the different types of patients. We develop approximation techniques to solve the admission control problem. Moreover, we compare the results with ordinal optimisation techniques to verify the quality of the results.

The model provides a first step in the workforce planning process; after the capacity has been derived by the model, one needs to make rosters in which individual personnel members are assigned tasks. Moreover, one needs to make efficient routes from the care-at-home facility to the patients, or appointments for their therapy treatments. These problems are not taken into account in our model, since many algorithms and software packages already exist to deal with this.

We start with giving a full problem description in Section 4.1. Then we consider the home care model, which is in fact a special case of the

more general model. In Secion 4.2 we describe this special case, and then study some structural properties of the value function and the state space. In Sections 4.3 and 4.4 we describe two different approaches to solve the more general rehabilitation care model, where there is no longer any structure in the value function or optimal policy. In all three sections we give some numerical examples to demonstrate the quality of the solutions.

4.1 Model formulation

We consider a care facility at which patients arrive requesting service from different specialties simultaneously for a specific duration. We assume that the facility has M different specialties and that there are C_m servers available for specialty m, $m = 1, \ldots, M$. These servers do not represent the number of different employees with a certain specialty, but we represent each unit of available time for a given specialty as a server. So if there are three employees with specialty m that have in total, say, 80 hours a week available for patient care, that would give $C_m = 80$.

The different combinations of services that can be requested are grouped into patient profiles or types, of which we assume that there are K. Patients arrive at the facility according to a Poisson process with rate λ, and an arriving patient has type k with probability p_k, $k = 1, \ldots, K$. Upon arrival of a patient with profile k, the patient requests c_k^m servers of specialty m. The service time has a Coxian distribution with r phases, where a patient of type k has service rate μ_{ki} in phase i, $i = 1, \ldots, r$, and goes to the next phase with probability p_{ki}. This means that during the length of time a patient stays at the facility, which has a Coxian distribution, he will continuously occupy c_k^m servers. The facility is constrained to serve at maximum a total of N patients simultaneously if capacity permits.

The care facility is subject to holding costs a_k for each patient in the queue with profile k, and holding costs 1 for each patient in treatment. The aim of the care facility is to minimise the long-term average costs for the system by optimally matching the capacity of the resources to the requests of the patients. Thus, at each arrival one needs to decide if the patient is going to be served immediately or if the patient is to be put on a waiting list. This decision also has to be taken when a patient finishes his treatment and leaves the facility. To answer this decision problem, we model the care facility as a continuous-time Markov decision problem.

Let $\mathcal{S} = \mathbb{N}^K \times \{0, \ldots, N\}^{K \times r}$ denote the state space of the system, with $\vec{s} = (\vec{q}, \vec{x}) \in \mathcal{S}$ where q_k denotes the number patients of type k waiting in the queue (of infinite size) and x_{ki} denotes the number of patients of type k in phase i of their service time. Let $V(\vec{s})$ be a real-valued function defined on the state space. This function will play the role of the relative value function, i.e., the asymptotic difference in total costs that results from starting the process in state $\vec{s}$ instead of some reference state. The long-term average optimal actions are a solution of the optimality equation (in vector notation) $g + V = TV$, with g the long-term average costs incurred in the system and with T the dynamic programming operator acting on $V(\vec{s})$ defined as follows

$$\begin{aligned} TV(\vec{s}) = TV(\vec{q}, \vec{x}) = \sum_{k=1}^{K} a_k q_k + \sum_{k=1}^{K} \sum_{i=1}^{r} x_{ki} + \lambda \sum_{k=1}^{K} p_k H(\vec{q} + e_k, \vec{x}) + \\ \sum_{k=1}^{K} \sum_{i=1}^{r-1} \mu_{ki} x_{ki} \big(p_{ki} V(\vec{q}, \vec{x} - e_{ki} + e_{ki+1}) + (1 - p_{ki}) H(\vec{q}, \vec{x} - e_{ki}) \big) + \\ \sum_{k=1}^{K} \mu_{kr} x_{kr} H(\vec{q}, \vec{x} - e_{kr}) + \Big(1 - \lambda - \sum_{k=1}^{K} \sum_{i=1}^{r} \mu_{ki} x_{ki} \Big) V(\vec{q}, \vec{x}), \end{aligned} \tag{4.1}$$

where

$$\begin{aligned} H(\vec{s}) = H(\vec{q}, \vec{x}) = \min\{ V(s') = V(\vec{q}', \vec{x}') \mid q'_k + \sum_{i=1}^{r} x'_{ki} = q_k + \sum_{i=1}^{r} x_{ki} \\ \text{for } k = 1, \ldots, K, \text{ and } x'_{ki} = x_{ki} \text{ for } k = 1, \ldots, K, \ i = 2, \ldots, r, \\ \text{and } x'_{k1} \geq x_{k1} \text{ for } k = 1, \ldots, K, \text{ and } \sum_{k=1}^{K} \sum_{i=1}^{r} x'_{ki} c_m^k \leq C_m \\ \text{for } m = 1, \ldots, M \}. \end{aligned} \tag{4.2}$$

The first term in the expression $V(\vec{s})$ models the holding costs for patients in the queue, whereas the second term models the holding costs for patients who are in service. The third term models the arrivals of patients to the system. The next three terms describe how patients move through the different phases of the Coxian service distribution: the first models the advance of a phase, the second the termination after completing a phase, and

the last the service at the last phase. Finally, the last term models the uniformisation constant (see Section 11.5 of Puterman [93]). To this end, we assume that the uniformisation constant $\lambda + \sum_{k=1}^{K} \sum_{i=1}^{r} \mu_{ki} N = 1$; we can always get this by scaling. Uniformising is equivalent to adding dummy transitions (from a state to itself) such that the rate out of each state is equal to 1; then we can consider the rates to be transition probabilities.

The function $H(\vec{s})$ models the decision making in the patient prioritisation problem. The first term in the condition models the fact that there is no difference in the number of patients in the system and queue before and after decision making. The next two conditions ensure that patients cannot change their phases due to the admission actions, and thus only the number of patients in the first phase can increase. Finally, the last condition requires that the admission actions do not violate the capacity constraint.

The optimality equation $g + V = TV$ is hard to solve analytically in practice. Alternatively, the optimal actions can also be obtained by recursively defining

$$V_{i+1} = TV_i, \tag{4.3}$$

for $i = 0, \ldots$ and arbitrary V_0. For $i \to \infty$, the maximising actions converge to the optimal ones (for existence and convergence of solutions and optimal policies we refer to Puterman [93]). Note that the optimal actions, which can be derived from the function $H(\cdot)$, completely depend on the relative value function $V(\cdot)$. Hence, in Section 4.3 we will focus on the relative value function $V(\cdot)$, and adopt approach (4.3) for deriving the structure of the optimal policy and to numerically compute optimal policies.

4.2 Home care: state space aggregation

In this section we address a special case of the model described above, that is relevant for home care facilities. In this case, there is only one specialty, so $M = 1$, and each patient of type k requires home care for c_k time units per week. We assume that patients stay in the system during a time that has an exponential distribution with parameter μ_k. For this special case, we also assume that there is a possibility to reject patients. We can now simplify the model description for this section.

Denote by $\vec{x} = (x_1, \ldots, x_K)$ the state of the home care employees, i.e., x_k is the number of patients of class k in service for $k = 1, \ldots, K$. When the state vector $\vec{x}$ is given, then the spare capacity in the system is given by $\text{cap}(\vec{x}) = S - \sum_{k=1}^{K} c_k \cdot x_k$. Now, let us suppose that a patient from class k arrives. The facility has several options to deal with this request. First, consider the scenario in which there is insufficient service capacity available (thus, $\text{cap}(\vec{x}) < c_k$). Then the facility can reject the request, but can also decide to admit the patient so that the patient is put on a waiting list for home care. For this purpose, let $\vec{q} = (q_1, \ldots, q_K)$ denote the number of patients of each class that are on the waiting list. In case there is sufficient capacity (i.e., $\text{cap}(\vec{x}) \geq c_k$), the facility has three options. First, it can again reject the request, because it expects other arrivals of patients that might conflict with the current request (this can happen, especially, when c_k is large), it can put the patient on the waiting list, or it can admit the patient immediately so that its service can start without delay. We assume that the waiting list can only hold B patients. If a patient is required to wait while the waiting list already has B patients, then the patient is rejected anyway.

We assume that the system is subject to rejection costs r_k for request k with $k = 1, \ldots, K$, and that there are costs for having a patient in the system (either waiting or in service). We are interested in finding a policy that balances the rejection costs and the average number of patients in the system by minimising the joint cost function. To this purpose, we denote the state of the care-at-home facility by $(\vec{x}, \vec{q})$ with state space $\mathcal{S} = \left\{(\vec{x}, \vec{q}) \,\middle|\, \sum_{k=1}^{K} c_k \cdot x_k \leq S, \sum_{k=1}^{K} q_k \leq B\right\}$. When a patient of class k arrives, the optimal action can be determined by the minimising action in H_a given by: $H_a(\vec{x}, \vec{q}, k) =$

$$\begin{cases} V(\vec{x}, \vec{q}) + r_k, & \text{if } \text{cap}(\vec{x}) < c_k, \sum_{k=1}^{K} q_k = B, \\ \min\{V(\vec{x}, \vec{q}) + r_k, V(\vec{x}, \vec{q} + e_k)\}, & \text{if } \text{cap}(\vec{x}) < c_k, \sum_{k=1}^{K} q_k < B, \\ \min\{V(\vec{x}, \vec{q}) + r_k, V(\vec{x} + e_k, \vec{q})\}, & \text{if } \text{cap}(\vec{x}) \geq c_k, \sum_{k=1}^{K} q_k = B, \\ \min\{V(\vec{x}, \vec{q}) + r_k, V(\vec{x}, \vec{q} + e_k), V(\vec{x} + e_k, \vec{q})\}, & \text{otherwise,} \end{cases}$$

with e_k the vector with zeros and a one at the k-th entry. The terms $V(\vec{x}, \vec{q}) + r_k$, $V(\vec{x}, \vec{q} + e_k)$, and $V(\vec{x} + e_k, \vec{q})$ represent the value of rejecting, delaying, and admitting a patient, respectively. A similar result holds when a patient no longer requires service so that home care capacity becomes available. In that case, a patient that is delayed can be taken into

service. We denote by H_d the term that deals with the actions after a departure of a patient given by: $H_d(\vec{x},\vec{q}) =$

$$\min_{(\vec{x'},\vec{q'})\in\mathcal{S}} \left\{ V(\vec{x'},\vec{q'}) \mid x'_k \geq x_k \text{ for } k=1,\ldots,K, \sum_{k=1}^{K}(x_k+q_k) = \sum_{k=1}^{K}(x'_k+q'_k) \right\}.$$

Note the H_d allows multiple patients to be admitted into the system, since it could happen that a patient with a large service utilisation has left. This event could free up capacity for multiple patients with a relatively small demand for capacity. The first condition between the brackets ensures that *new* patients are admitted, whereas the second condition makes sure that the total number of patients in both situations are equal so that no patients are rejected.

To fully control the system one needs a patient admission policy. This policy can be determined if the relative value function is known. Let g denote the long-term average cost in the system. For simplicity we assume that $\lambda + S\max_{\{k=1,\ldots,K\}}\{\mu_k\} < 1$; without loss of generality, we can always achieve this by scaling. The relative value function can now be determined by solving the optimality equation (in vector notation) $g + V = TV$, where T is the dynamic programming operator acting on $V(\vec{x},\vec{q})$ defined as follows

$$TV(\vec{x},\vec{q}) = \sum_{k=1}^{K}(x_k+q_k) + \sum_{k=1}^{K}\lambda p_k H_a(\vec{x},\vec{q},k) + \sum_{k=1}^{K} x_k\mu_k H_d(\vec{x}-e_k,\vec{q}) + \left(1-\lambda-\sum_{k=1}^{K}x_k\mu_k\right)V(\vec{x},\vec{q}).$$

The first term in the equality counts the number of patients in service and on the waiting list. The second term models the arrivals of patients and the optimal admission policy. The third term deals with the situation in which a home care service ends and a patient departs the system. The last term is the dummy term that follows from uniformisation of the system.

In the remainder of this section we study the optimal scheduling policies for the care-at-home problem. We distinguish between two cases: one with no waiting room for arriving patients, and one with waiting room. The case without waiting room has been studied in literature before in the setting of bandwidth allocation in telecommunication systems. We provide an overview of the results in the literature for this case. The case with

waiting room is inherently more complex and has been given little attention in the literature. We provide monotonicity results for this case and characterise part of the optimal policy.

4.2.1 The case with no waiting room

In this section we analyse the care-at-home model that has been described in the previous section. However, before studying the general model, we first study the case with no waiting, i.e., $B = 0$. The optimality equations then reduce to

$$g + V(\vec{x}) = \sum_{k=1}^{K} x_k + \sum_{k=1}^{K} \lambda p_k \Big[\mathbb{1}_{\{\text{cap}(\vec{x}) < c_k\}} [V(\vec{x}) + r_k] + \mathbb{1}_{\{\text{cap}(\vec{x}) \geq c_k\}} \min\{V(\vec{x}) + r_k, V(\vec{x} + e_k)\} \Big] + \sum_{k=1}^{K} x_k \mu_k V(\vec{x} - e_k) + \Big(1 - \lambda - \sum_{k=1}^{K} x_k \mu_k \Big) V(\vec{x}).$$

In case the policy is always to accept when possible, then the system reduces to the multi-rate blocking model. This is an extension of the Erlang blocking model and has been well-studied in a telecommunications setting. In this setting there is a certain amount of bandwidth (home care service capacity) available that arriving Internet requests (patients) can use. The Internet requests require part of the bandwidth (the number of hours per week) for a certain duration (the number of weeks consecutively). The model under this policy has a product-form solution for its steady-state distribution. Thus, let $\rho_k = \lambda p_k / \mu_k$ for $k = 1, \ldots, K$. Then the probability of being in state $\vec{x} \in \mathcal{S}$ is given by $\pi(\vec{x})$ and has the form

$$\pi(\vec{x}) = \pi(x_1, \ldots, x_K) = \frac{1}{G} \prod_{k=1}^{K} \frac{\rho_k^{x_k}}{x_k!} \qquad \text{with} \qquad G = \sum_{\vec{x} \in \mathcal{S}} \prod_{k=1}^{K} \frac{\rho_k^{x_k}}{x_k!}.$$

Let S_k denote the subset of states in which a call of class k is admitted to the system, i.e., $S_k = \{\vec{x} \in \mathcal{X} \,|\, \text{cap}(\vec{x}) \geq c_k\}$. Then the blocking probability of a call of class k is given by $B_k = 1 - \sum_{\vec{x} \in S_k} \pi(\vec{x})$. However, the numerical evaluation can be a problem and the Kaufman-Robert recursion alleviates

this problem. The long-term average cost g can then be efficiently calculated by $g = \sum_{k=1}^{K} B_k r_k$. The model with $B = 0$ which we study is an extension of the multi-rate blocking model. Blocking a patient of class k brings with it a cost of r_k that can be different for each class. The optimal policy with these different cost rates is usually different than the policy used in the multi-rate blocking model.

In case $c_k \equiv 1$ and $\mu_k \equiv \mu$, it is known that trunk reservation is optimal (Miller [**84**]). However, in a more general setting, Ross and Tsang [**95**] showed that the trunk reservation policy is not optimal anymore. Under the assumption that $c_k \leq c_{k+1}$ and $\mu_k \geq \mu_{k+1}$, Altman et al. [**4**] derived a stochastic ordering of the patient classes such that priority is given to the patient class with the smallest index. If in addition we make the assumption that $r_k \leq r_{k+1}$, then it can be shown that the trunk reservation policy is optimal again. The more general case with no assumptions has only been studied in a fluid model, in which the authors showed the optimality of trunk reservation.

4.2.2 The general case with a waiting room

In this section we treat the care-at-home model in which patients are allowed to wait as well. This case is significantly more difficult than the case previously discussed. Unlike the case with no waiting room, there is very limited literature available on the care-at-home model with waiting room. Therefore, we start with some structural properties of the relative value function V, which gives us insight into the structure of the optimal policy. We start by showing that V is an increasing function in all of its components. This is formalised by the following lemma.

4.2.1. Lemma (increasingness)**.** *For all $(\vec{x}, \vec{q}) \in \mathcal{S}$ and $(\vec{x} + e_k, \vec{q}) \in \mathcal{S}$ we have*

$$V(\vec{x} + e_k, \vec{q}) \geq V(\vec{x}, \vec{q}),$$

for $k = 1, \ldots, K$. Similarly, for all $(\vec{x}, \vec{q}) \in \mathcal{S}$ and $(\vec{x}, \vec{q} + e_k) \in \mathcal{S}$ we have

$$V(\vec{x}, \vec{q} + e_k) \geq V(\vec{x}, \vec{q}),$$

for $k = 1, \ldots, K$.

Proof. The proof is by induction on n in V_n. Define $V_0(\vec{x}, \vec{q}) = 0$ for all states $(\vec{x}, \vec{q}) \in \mathcal{S}$. Then, clearly, $V_0(\vec{x}, \vec{q})$ is increasing in all components of $\vec{x}$ and $\vec{q}$. Now, assume that the statement of the lemma holds for V_n for some $n \in \mathbb{N}$. Now, we prove that $V_{n+1}(\vec{x}, \vec{q})$ satisfies the increasingness property as well. Therefore, fix $k \in \{1, \ldots, K\}$ and assume that $(\vec{x} + e_k, \vec{q}) \in \mathcal{S}$, then

$$\begin{aligned} V_{n+1}(\vec{x} + e_k, \vec{q}) - V_{n+1}(\vec{x}, \vec{q}) &= 1 + \sum_{j=1}^{K} \lambda p_j \big[H_a(\vec{x} + e_k, \vec{q}, j) - H_a(\vec{x}, \vec{q}, j)\big] \\ &+ \sum_{j=1}^{K} x_j \mu_j \big[H_d(\vec{x} + e_k - e_j, \vec{q}) - H_d(\vec{x} - e_j, \vec{q})\big] + \mu_k H_d(\vec{x}, \vec{q}) \\ &+ \Big(1 - \lambda - \sum_{j=1}^{K} (x_j + \mathbb{1}_{\{j=k\}})\mu_j\Big) \big[V_n(\vec{x} + e_k, \vec{q}) - V_n(\vec{x}, \vec{q})\big] - \mu_k V_n(\vec{x}, \vec{q}). \end{aligned}$$

The first term of the righthand-side equals 1, since that is exactly the difference in the number of patients in both systems. The second term deals with the arrivals. Note that the optimal action in $H_a(\vec{x} + e_k, \vec{q}, j)$ can be used in $H_a(\vec{x}, \vec{q}, j)$ as well (possibly as a suboptimal action). In doing so, we get that $H_a(\vec{x} + e_k, \vec{q}, j) - H_a(\vec{x}, \vec{q}, j) \geq 0$ due to the induction hypothesis. The same holds for the term dealing with actions after departures. The fourth term cancels the sixth term. For the fifth term the induction hypothesis directly applies.

Now assume that $(\vec{x}, \vec{q} + e_k) \in \mathcal{S}$, then

$$\begin{aligned} V_{n+1}(\vec{x}, \vec{q} + e_k) - V_{n+1}(\vec{x}, \vec{q}) &= 1 + \sum_{j=1}^{K} \lambda p_j \big[H_a(\vec{x}, \vec{q} + e_k, j) - H_a(\vec{x}, \vec{q}, j)\big] \\ &+ \sum_{j=1}^{K} x_j \mu_j \big[H_d(\vec{x} - e_j, \vec{q} + e_k) - H_d(\vec{x} - e_j, \vec{q})\big] \\ &+ \Big(1 - \lambda - \sum_{j=1}^{K} x_j \mu_j\Big) \big[V_n(\vec{x}, \vec{q} + e_k) - V_n(\vec{x}, \vec{q})\big]. \end{aligned}$$

The first term of the righthand-side equals 1, since that is exactly the difference in the number of patients in both systems. The second term deals with the arrivals. Note that the optimal action in $H_a(\vec{x}, \vec{q} + e_k, j)$ can be used in $H_a(\vec{x}, \vec{q}, j)$ as well (possibly as a suboptimal action). In doing so,

we get that $H_a(\vec{x}, \vec{q}+e_k, j) - H_a(\vec{x}, \vec{q}, j) \geq 0$ due to the induction hypothesis. A similar remark holds for the next term with only a slight difference. In case the optimal action in $H_d(\vec{x} - e_j, \vec{q} + e_k)$ serves $q_k + e_k$ patients of type k, then the (sub)optimal action in $H_d(\vec{x} - e_j, \vec{q})$ should serve q_k patients of type k. The resulting state will differ by 1 for x_k while q_k is equal and so the induction hypothesis still applies. For the last term the induction hypothesis directly applies.

We conclude, by taking the limit of $n \to \infty$, that $V(\vec{x}, \vec{q})$ is increasing in x_k and q_k for all $k = 1, \ldots, K$. ∎

For the special case where all service needs of all classes of patients are equal, we can give a very simple rule that gives optimal results. This rule states that after a departure the patient with the smallest expected service time is taken into service. This minimises the number of patients in the system, and thus also the total rejection and holding costs.

4.2.2. Theorem. *Assume that $c_k = 1$ for all $k = 1, \ldots, K$. Suppose that $\mu_k > \mu_{k+1}$. Then, the optimal scheduling policy schedules people with the smallest index first.*

Proof. Because the theorem only concerns actions taken just after a departure, only H_d needs to be taken into account. Also, because the number of servers needed is equal for all types of patients, only one new patient will be taken into service at the same time after a departure and no room needs to be saved for patients with large service requirements. So what we need to prove is that $V(\vec{x} + e_i, \vec{q} - e_i) \leq V(\vec{x} + e_j, \vec{q} - e_j)$ if $i < j$.

The proof is by induction on n in V_n. Define $V_0(\vec{x}, \vec{q}) = 0$ for all states $(\vec{x}, \vec{q}) \in \mathcal{S}$. Then, clearly, $V(\vec{x} + e_i, \vec{q} - e_i) \leq V(\vec{x} + e_j, \vec{q} - e_j)$ holds for all values of i and j. Now, assume that the statement of the theorem holds for

V_n for some $n \in \mathbb{N}$. Now, we prove that $V_{n+1}(\vec{x}, \vec{q})$ satisfies the theorem as well. Therefore assume that $(\vec{x} + e_i, \vec{q} - e_i), (\vec{x} + e_j, \vec{q} - e_j) \in \mathcal{S}$, then

$$\begin{aligned}
&V_{n+1}(\vec{x} + e_i, \vec{q} - e_i) - V_{n+1}(\vec{x} + e_j, \vec{q} - e_j) = 0 \\
&\quad + \sum_{k=1}^{K} \lambda p_k \big(H_a(\vec{x} + e_i, \vec{q} - e_i, k) - H_a(\vec{x} + e_j, \vec{q} - e_j, k)\big) \\
&\quad + \sum_{k=1}^{K} x_k \mu_k \big(H_d(\vec{x} + e_i - e_k, \vec{q} - e_i) - H_d(\vec{x} + e_j - e_k, \vec{q} - e_j)\big) \\
&\quad + \mu_i H_d(\vec{x}, \vec{q} - e_i) - \mu_j H_d(\vec{x}, \vec{q} - e_j) \\
&\quad + \big(1 - \lambda - \sum_{k=1}^{K} x_k \mu_k\big)\big(V_n(\vec{x} + e_i, \vec{q}) - V_n(\vec{x} + e_j, \vec{q} - e_j)\big) \\
&\quad - \mu_i V_n(\vec{x} + e_i, \vec{q} - e_i) + \mu_j V_n(\vec{x} + e_j, \vec{q} - e_j).
\end{aligned}$$

The first term of the right-hand side is 0 since the total number of patients present in the system is equal in both cases. The second term deals with arrivals. The optimal action that is chosen in $H_a(\vec{x} + e_i, \vec{q} - e_i, k)$ can also be chosen in $H_a(\vec{x} + e_j, \vec{q} - e_j, k)$, although it is not necessarily optimal. Then, by the induction hypothesis, $H_a(\vec{x} + e_i, \vec{q} - e_i, k) - H_a(\vec{x} + e_j, \vec{q} - e_j, k) \geq 0$. The same goes for the third term dealing with departures, with the added remark that in the case when in $H_d(\vec{x} + e_i - e_k, \vec{q} - e_i)$ chooses to take q_i patients of type i in service, then in the other case $q_i - 1$ patients of that type should be taken into service. Then the induction hypothesis can be applied. To the sixth term the hypothesis can be applied directly. For the fourth, fifth, seventh and eighth term some rewriting is needed:

$$\begin{aligned}
&\mu_i H_d(\vec{x}, \vec{q} - e_i) - \mu_j H_d(\vec{x}, \vec{q} - e_j) - \mu_i V_n(\vec{x} + e_i, \vec{q} - e_i) + \mu_j V_n(\vec{x} + e_j, \vec{q} - e_j) \\
&\quad = -\mu_i \big(V_n(\vec{x} + e_i, \vec{q} - e_i) - H_d(\vec{x}, \vec{q} - e_i)\big) + \mu_j \big(V_n(\vec{x} + e_j, \vec{q} - e_j) \\
&\quad - H_d(\vec{x}, \vec{q} - e_i)\big).
\end{aligned}$$

We know that $\mu_i > \mu_j$. This is smaller than 0 because the difference between $V_n(\vec{x} + e_i, \vec{q} - e_i)$ and $V_n(\vec{x} + e_j, \vec{q} - e_j)$ or between taking a type i or type j patient into service is smaller than the difference between $H_d(\vec{x}, \vec{q} - e_i)$ and $H_d(\vec{x}, \vec{q} - e_j)$, the optimal action on departure of a type i or type j patient respectively.

Then, by taking the limit of $n \to \infty$, $V(\vec{x} + e_i, \vec{q} - e_i) \leq V(\vec{x} + e_j, \vec{q} - e_j)$ if $\mu_i > \mu_j$. ∎

The next theorem shows that if the service requirements are equal for all types of patients, it is never optimal to leave a patient in the queue while there are servers available. So in this case the optimal policy will be a work-conserving policy. This also follows from intuition, because to minimise the number of patients present in the system you will serve them as quickly as possible once they have been admitted.

4.2.3. Theorem. *Assume that $c_k = 1$ for all $k = 1, \ldots, K$. Then an optimal policy will schedule patients in service while there are idle servers available.*

Proof. We need to prove that upon arrival, if a server is available, the patient will use this server. This means that we need that $V(\vec{x} + e_k, \vec{q}) \leq V(\vec{x}, \vec{q} + e_k)$.

Again this proof is by induction on n in V_n. Define $V_0(\vec{x}, \vec{q}) = 0$ for all states $(\vec{x}, \vec{q}) \in \mathcal{S}$. Then of course $V_0(\vec{x} + e_k, \vec{q}) \leq V_0(\vec{x}, \vec{q} + e_k)$ holds. Now assume the proposition holds for some $n \in \mathbb{N}$. Now we prove that for $n+1$ the statement holds as well. For $n + 1$ we have

$$
\begin{aligned}
&V_{n+1}(\vec{x}, \vec{q} + e_k) - V_{n+1}(\vec{x} + e_k, \vec{q}) = \\
&\qquad \sum_{j=1}^{K} \lambda p_j \big(H_a(\vec{x}, \vec{q} + e_k, j) - H_a(\vec{x} + e_k, \vec{q}, j) \big) \\
&\qquad + \sum_{j=1}^{K} x_j \mu_j \big(H_d(\vec{x} - e_j, \vec{q} + e_k) - H_d(\vec{x} + e_k - e_j, \vec{q}) \big) - \mu_k H_d(\vec{x}, \vec{q}) \\
&\qquad + \big(1 - \lambda - \sum_{j=1}^{K} x_j \mu_j \big) \big(V_n(\vec{x}, \vec{q} + e_k) - V_n(\vec{x} + e_k, \vec{q}) \big) + \mu_k V_n(\vec{x} + e_k, \vec{q}).
\end{aligned}
$$

The first term concerns the arrivals. It is easily seen that when taking the same action in the second part as is optimal in the first part, the term satisfies the statement using the induction hypothesis. The second term deals with actions after departures. Because of the induction hypothesis, as many patients as possible are taken into service. This means that the resulting state will be the same in both cases, and the second term is equal to zero. To the fourth term the induction hypothesis can be applied directly. The third term is smaller than the fifth term as a result of Lemma 4.2.1. This means that $V_{n+1}(\vec{x}, \vec{q} + e_k) - V_{n+1}(\vec{x} + e_k, \vec{q}) \geq 0$ and the proof is complete. ∎

4.2.3 Numerical experiments

In this section we perform numerical experiments to illustrate the results of the previous sections. We discuss how the optimality equations have been solved efficiently to derive the optimal policies numerically. We start with a discussion on parallelisation of the implementation that has been crucial to the computations.

Parallelisation

In order to accommodate for the heavy use of memory when using value iteration, the dynamic program has been parallelised with MPI so that it can be run on the DAS-3 cluster computer [**32**]. However, this is not a straightforward procedure.We now explain how this was done and how parallelisation yields a gain in performance.

The calculation steps in value iteration are very similar to the steps used in Successive Overrelaxation (SOR). The SOR algorithm is used for calculating the value of a stable state in a two-dimensional matrix, in which the edges are given an initial value, and the other points are calculated using their neighbouring values. The total workload is divided block-wise over the available processors of the cluster computer, limiting the inter-processor communication to the neighbouring rows of these blocks. Using this scheme, each processor only needs a fraction of the data. The main difference between the SOR algorithm and our value iteration algorithm is that the matrix in our problem is not two-dimensional. This shifts the focus from computation speed to memory use. In fact, for $K = 4$ and $B = 13$, the memory requirements are about 45 Gigabytes (assuming that the relative value function evaluated in a particular state requires a double to store its value).

The parallel program has been engineered to make optimal use of the memory while retaining most of the computational speed. In the value iteration algorithm, new values of the relative value function are communicated and stored directly into the neighbouring blocks, without the use of additional communication buffers. Overall, the parallel version of value iteration uses significantly more memory than the sequential version, because some values in a block need to be replicated. The sequential program has a memory usage U_{seq} that is roughly of the size

$$U_{\text{seq}} = 2 \times 16 \times (\max\{B, S\} + 1)^{2K} \text{ bytes.} \tag{4.4}$$

The parallel version needs additional $2 \times 16 \times (\max B, S)^{2K-1}$ bytes per processor for communication. While this increases the memory usage a bit, the parallel version allows us to divide this memory in blocks over the available processors. The memory requirements U_{par} of the parallel version with p processors is then given by

$$U_{\text{par}} = \frac{U_{\text{seq}} + 2 \times 16 \times p \times (\max\{B, S\} + 1)^{2K-1}}{p} \text{ bytes.} \qquad (4.5)$$

Figure 4.1 illustrates the memory requirements for various problems with a state space that has eight dimensions. Similarly, Figure 4.2 shows the speedup in computation time as more processors are utilized in the cluster computer.

CPUs	problem size ($\times 100,000$)	U_{seq} (in Gb)	U_{par} (in Gb)
4	0.65	0.002	0.001
5	3.91	0.012	0.004
10	1,000.00	3.050	0.610
12	4,300.00	13.120	2.190
13	8,157.00	24.894	3.830
14	14,758.00	45.040	6.430

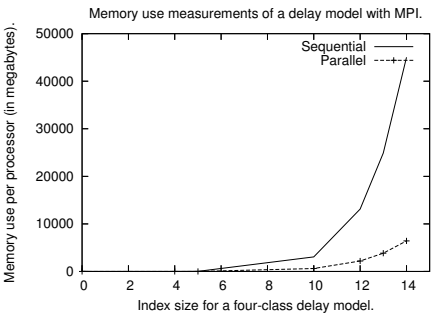

Figure 4.1: Memory requirement for various eight-dimensional problems

CPUs	time	speedup
1	64.17	0.00
2	57.81	1.11
4	32.29	1.99
8	18.60	3.45
16	11.07	5.79
32	5.87	10.94

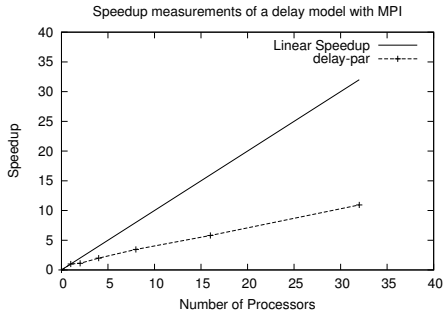

Figure 4.2: Results of the speedup for a four-dimensional problem, $S = 32$

In this section we describe some experiments we did to show how the model performs. For both the case with and without a waiting room we have performed experiments with the same parameters with regard to demand and number of servers available, so as to make the two cases comparable.

Scenario analysis

In the remainder of this subsection, we describe several scenarios and compare the results and analyse the structure of the optimal policy. The scenarios we consider are given in Table 4.1. We keep the number of arriving patients and the total workload offered (almost) the same, but increase the number of different classes of patients, with the number of servers requested for each class equal to the class number. This approach will enable us to study the influence of the variability of patient demand on the performance of the system. The scenarios also reflect the fact seen in practice that patients requiring more time per week tend both to be more rare and to have a longer service time. As these patients are also generally more urgent, they also have higher rejection costs.

In the results we can see that the average cost when using the optimal policy increases with the variation in service requirements of the patients. This is of course to be expected, because reservation of larger numbers of servers is necessary for patients that need a larger number of servers, and this causes other patients to wait for a longer period of time. However, the increase in cost is not as high as might have been expected.

Table 4.2 also shows the results for our heuristic, which is the best trunk reservation policy. With a trunk reservation policy we mean a policy in which some patient classes are blocked as a certain number of servers are occupied. For each patient class but the one with the highest priority there is such a threshold. As can be seen the relative difference in average cost between the optimal policy and the best trunk reservation policy is very small. This means that the trunk reservation policy is a very good practical alternative to the optimal policy, as it is almost as good but much easier to compute, visualise and implement.

One last remark we want to make is a point about actions taken upon departure of a patient. If a patient who uses a high number of servers leaves the system, it is possible to take into service a number of patients with lower service requirements. This can be seen in the expression for H_d. However, from our experiments it has become clear that the average cost

scenario	p_k	c_k	$\beta_k = 1/\mu_k$	r_k	scenario	p_k	c_k	$\beta_k = 1/\mu_k$	r_k
1	0.7	1	22	1	5	0.2	1	1	1
	0.3	2	28	2		0.2	2	2	2
2	0.5	1	10	1		0.1	3	2	3
	0.3	2	20	2		0.1	4	2	4
	0.2	3	25	3		0.05	5	5	5
3	0.3	1	5	1		0.05	6	5	6
	0.2	2	10	2		0.05	7	5	7
	0.2	3	12	3		0.05	8	5	8
	0.2	4	15	4		0.05	9	5	9
	0.1	5	15	5		0.03	10	5	10
4	0.2	1	2	1		0.03	11	10	11
	0.2	2	3	2		0.03	12	10	12
	0.1	3	5	3		0.03	13	10	13
	0.1	4	5	4		0.03	14	15	14
	0.1	5	5	5		0.03	15	15	15
	0.1	6	5	6					
	0.05	7	10	7					
	0.05	8	10	8					
	0.05	9	15	9					
	0.05	10	15	10					

Table 4.1: Scenarios for experiments with $\lambda = 5$.

does not increase significantly if it is assumed that at most one new patient is taken into service after a departure. The optimal policy differs only in very few cases, and then only at the boundaries of the state space. It does however speed up the computation significantly to make this assumption.

4.3 Rehabilitation model: Approximate Dynamic Programming

In this section and the next one we study the extended model with more than one specialty. For this more general model, the approach of the previ-

scenario	optimal	heuristic
1	27.611	0.52%
2	31.408	0.78%
3	31.593	0.83%
4	37.884	1.30%
5	38.778	1.82%

Table 4.2: Results for the different scenarios

ous section will not work as well, because there is more information in the state space. Aggregating too much leads to bad results, and without this aggregation the state space is not reduced enough to make for a tractable model. For this reason, we use two different approaches that do work well for this more extended case. In this section we use a method that replaces one step by an approximation. In the next section we describe a stochastic programming method that changes the stochastic aspects of the model to make the problem deterministic.

The model described in Section 4.1 gives a complete description of the system dynamics and also provides a recipe to obtain the optimal state-dependent actions. However, the recipe is numerically intractable, since the state space is of order $\mathbb{N}^K \times \{0, \ldots, N\}^{K \times r}$. The memory requirements for already moderate values of K, N, and r become the bottleneck and prohibit the derivation of the optimal policy.

In this section, we develop an approximate dynamic programming algorithm which does not suffer from the dimensionality problem of the original problem formulation. In principle, one can apply sophisticated approximate dynamic programming techniques to achieve this by exploiting structural properties of the relative value function (such as increasingness, convexity, sub/supermodularity, etc.). However, it has already been shown that there is little structure in the relative value function, since the problem can be reduced to a stochastic knapsack problem (when $r = 1$ and $M = 1$) that does not have these structural properties either (see, e.g., Ross and Tsang [**95**] and Altman et al. [**4**]). Therefore, we adopt the approach outlined in Powell [**92**] to derive an efficient algorithm which yields near-optimal results. This means there is a trade-off in the com-

putational complexity and the quality of the resulting policy. In order to study this trade-off in more detail, we now present our description of the approximation.

In the optimality equation (4.1) we modelled the system dynamics through the transition probabilities whenever an action is taken. The transition probabilities model the new state $\vec{s}'$ after an action a has been applied in state $\vec{s}$. However, there is an intermediary state, the so-called post-decision state $\vec{\tilde{s}}$, before reaching the final $\vec{s}'$. The post-decision state is the state just after an action has been taken, but before any other event has occurred. The post-decision state provides a new way to look at the decision problem such that the dimensionality can be handled. The method consists of five steps.

(1) Start with a pre-decision state $\vec{s}$.

(2) Solve the deterministic optimisation problem using an approximate value function:

$$\tilde{v}(\vec{s}) = \min_{\vec{a}} \Big\{ \sum_{k=1}^{K} a_k q_k + \sum_{k=1}^{K} \sum_{i=1}^{r} x_{ki} + V(\vec{\tilde{s}} \,|\vec{s}, \vec{a}) \Big\}, \tag{4.6}$$

where $\vec{a}$ is the decision vector that is allowed in state $\vec{s}$.

(3) Update the value function approximation $V(\vec{\tilde{s}})$ by

$$(1-\alpha)V(\vec{\tilde{s}}) + \alpha\tilde{v}(\vec{s}). \tag{4.7}$$

(4) Obtain a Monte Carlo sample using the transition probabilities to generate the next pre-decision state $\vec{s'}$.

(5) Return to step 1.

In our case, in step (1) we start with an empty system as pre-decision state. Our initial form for $V(\vec{s})$ is given by $V(\vec{s}) = \sum_{k=1}^{K} a_k q_k + \sum_{k=1}^{K} \sum_{i=1}^{r} x_{ki}$, which represents the direct costs. This choice is based upon a first-order guess of the relative value function, which does reflect the costs that are obtained in the different states. In step (3) we use an update rule based on the bias-adjusted Kalman filter. This results in adaptive step sizes α_n based on n data points that are given by

$$\alpha_n = 1 - \frac{\sigma^2}{(1+\theta^n)\sigma^2 + \beta^n}, \tag{4.8}$$

where σ^2 is the variance of the observation noise, and β^n is the bias measuring the difference between the current estimate of the value function and the true value function. The values of θ^n can be recursively computed by

$$\theta^n = \begin{cases} \alpha_{n-1}^2, & n = 1, \\ (1-\alpha_{n-1})^2\theta^{n-1} + \alpha_{n-1}^2, & n > 1. \end{cases} \tag{4.9}$$

In step (4) we draw randomly the next event according to the transition probabilities to determine the next pre-decision state.

The advantage of using the post-decision variable in the optimisation is that the problem instance now belongs to the realm of combinatorial optimisation. This can be done very efficiently. In the next section, we study the performance of this algorithm with respect to the optimal policy.

4.3.1 Numerical results

We evaluate the performance of our approximations extensively by numerical experiments. We do this by comparing the long-run average costs to the performance of the optimal policy (OPT) and to the performance of the approximate dynamic programming algorithm (ADP). We generate 100 random instances for different parameter values and compare the performance of these algorithms.

First, we describe the setup of the experiments. We look at a problem with $K = 2$ patient types and $M = 2$ disciplines, respectively. In all cases we have that λ is uniformly drawn from $[1, 5]$. Service durations are Gamma distributed with mean μ_i in $[8, 15]$, which is then approximated by a Cox(5) distribution using the EM-algorithm described by Asmussen et al. [**9**] to obtain the parameters used in model (4.1). The probability p_k is random and drawn from $[0, 1]$. The factors c_m^k are also drawn from a set of fixed values (of which the parameters are outlined below between brackets) and determine the factors a_k by $a_k = \sum_{m=1}^{M} c_m^k/\mu_k$. The other settings are given by

$$(C_1, C_2) = (20, 20) \text{ and } c = \begin{pmatrix} 1 & \{0, \ldots, 3\} \\ \{0, \ldots, 3\} & 1 \end{pmatrix},$$

where $\{\cdot\}$ denotes a set from which uniformly a value is drawn. We randomly generate 100 parameter settings.

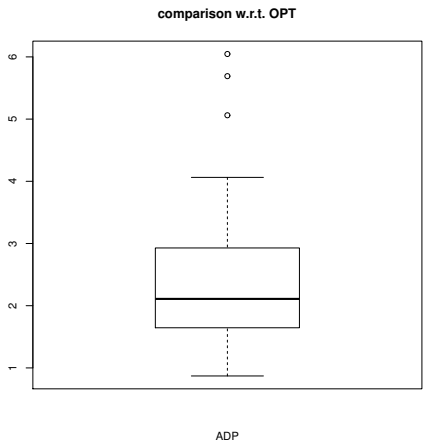

Figure 4.3: Comparison of the ADP algorithm for different settings.

Figure 4.3 shows the comparison of the different algorithms for these settings. The relative difference between the two algorithms is listed on the y-axis and is computed as

$$\frac{\text{performance(ADP)} - \text{performance(OPT)}}{\text{performance(OPT)}}.$$

In the boxplot, the thick line represents the median, surrounded by the 25% and the 75% quartile. This range is also called the IQR, the inter-quartile range. The whiskers represent the $1.5 \cdot$ IQR range (cut off by the last point that falls into that range) in which most of the points fall. In case the points do not fall into this range, the points can be considered as outliers.

In order to assess the quality of the ADP in a different manner, we also compare the relative difference of the ADP to the performance of Ordinal Optimisation (OO) methods (see Ho et al. [**64**]). The idea in ordinal optimisation is to randomly generate policies (drawn uniformly), say a 100 policies, and assess their performance by simulation. When the performance is sorted and a graph is drawn, the shape of the curve provides a clear indication of the difficulty of the problem. E.g., a shape that follows a square-root shape denotes that in the policy space there are a few good

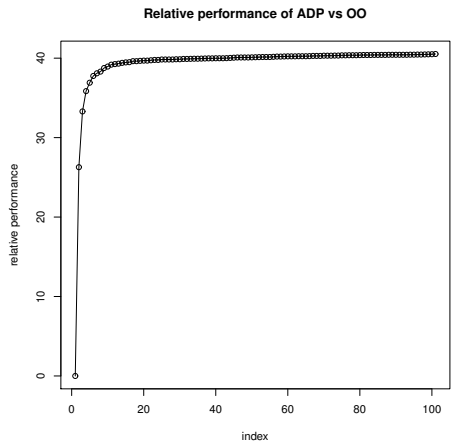

Figure 4.4: Comparison of the ADP algorithm with OO.

policies as compared to the majority of other policies; hence, the problem is hard. An inverted square-root shapes usually denotes that the problem is insensitive to minor change in the policy as many policies have similar performance.

We apply the ordinal optimisation technique in our example as well. We randomly generate 100 policies which are drawn as follows. In each state $\vec{s}$ we determine the set $\mathcal{A}_{\vec{s}}$ of all allowed actions. A policy is drawn by drawing uniformly an action from $\mathcal{A}_{\vec{s}}$ for each state. We include to the set of 100 policies also the policy that is generated from the ADP algorithm as first policy. This allows us to compare the results of OO with the ones obtained from ADP. Figure 4.3.1 displays the relative difference of the ADP and OO results. As the shape of curve shows, this admission control problem belongs to this set of hard problems in which very few good policies exist.

The figures clearly show that ADP consistently has a very good performance with little variance. The average deviation with respect to OPT is less than 3% over the whole range of the experiments. Unfortunately, the comparison cannot be done for larger systems as the optimal policy is already numerically intractable for larger systems. However, the approximate dynamic programming method is scalable and is shown to have very

good performance, making it an excellent alternative to control larger systems.

4.4 Alternative: stochastic programming approach

In this section we address the home care scheduling problem from Section 4.3 using stochastic programming. The model formulation is the same as the one used in the previous section. This method addresses the problem of the state space dimensionality by reducing the complexity in the model by replacing the stochasticity with scenarios. This gives us the opportunity to generate near-optimal solutions because all patient information is retained in the model. The method can also scale well for large problem instances.

Stochastic programming (SP) is an approach for solving optimisation problems under uncertainty, which uses distributional information on the random parameters involved. The goal of this method is to find a solution to the problem at hand that is feasible for almost all realisations of the random process and at the same time maximises or minimises the expectation of some criterion function. A general introduction to the field of stochastic programming is given by Ruszczyński and Shapiro [**96**]. Stochastic programming is applied to stochastic scheduling by Birge and Dempster [**20**], who investigate approximations at different levels of the decision hierarchy. Examples of other applications are power production and trading, see Schultz et al. [**97**], and revenue management, see Haensel et al. [**58**].

We will first introduce the stochastic programming formulation, and then we give some numerical examples to show the effectiveness of the method.

4.4.1 Stochastic programming formulation

In this section, we reformulate the Markov decision model into a stochastic programming model such that the detailed information on patient characteristics is used while retaining tractability of the model. The key idea in doing so is to remove the stochasticity in the Markov decision model by studying realisations of the stochastic variables in the process. These realisations are called scenarios. For each scenario, the problem becomes a deterministic problem that can be solved in a mathematical programming

setting. By studying several scenarios, the influence of the nature of the replaced variables can be studied so that a robust policy can be obtained.

Before going to the stochastic programming formulation, we summarise the notation that we used in the previous section. Note that some variables are slightly reformulated, and that some variables have a more general definition.

- $K \in \mathbb{N}$ – number of patient types
- $M \in \mathbb{N}$ – number of treatment disciplines
- $T \in \mathbb{N}$ – considered time horizon (in weeks)
- $\mu \in \mathbb{N}^M$ – estimated treatment duration per patient type (in whole weeks)
- $c \in \mathbb{R}^{M \times K}$ – patient type $\times$ discipline matrix
- $cap \in \mathbb{R}^{T \times K}$ – capacity per discipline and time stage
- $\alpha \in \mathbb{R}^K$ – importance weight per patient type
  $\alpha_k = \mu_k \cdot \sum_{m=1}^{M} c_{m,k}, \qquad \forall k = 1, \ldots, K$

In our stochastic programming setting, we model the stochastic patient arrival process as a discrete-time stochastic process on a probability space $(\Omega, \mathcal{F}, P)$. We approximate the random arrival process per time unit and patient type by a sample of S demand scenarios $d_s \in \mathbb{N}^{T \times K}$, $s = 1, \ldots, S$, each being realised with probability π_s. The duration of the treatment is random as well and we model this randomness by a set of L possible perturbation factors $\Lambda_{k,l}$ each realised with probability $\lambda_{k,l}$ ($k = 1, \ldots, K$ and $l = 1, \ldots, L$). To illustrate this with a small example: Consider two patient types with $\mu = \{2, 4\}$. Three possible treatment duration perturbations ($L = 3$) could be given by:

$$\Lambda = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \end{pmatrix}, \qquad \text{and} \qquad \lambda = \begin{pmatrix} 0.3 & 0.4 & 0.3 \\ 0.25 & 0.5 & 0.25 \end{pmatrix}.$$

This means that patients of type 1 have a treatment duration of one week ($= \mu_1 + \Lambda_{1,1} = 2 + (-1)$) with probability 0.3, with probability 0.4 two weeks and with probability 0.3 a treatment duration of three weeks.

Patient scheduling model without waiting queues
We first model the system with no queues, i.e., arriving patients are either taken into treatment or rejected upon arrival. The objective of the model is to minimise the number of rejected patients, where each patient type is weighted according to its importance factor α. The objective then takes the following form

$$\text{minimise} \quad \sum_{s=1}^{S} \pi_s \cdot \sum_{t=1}^{T} \sum_{k=1}^{K} \alpha_k \cdot r_{s,t,k}, \tag{4.10}$$

where $r_{s,t,k}$ denotes the number of rejected patients of type k in scenario s at time t.

At each time stage, arriving patients are either accepted and taken into treatment or they are rejected:

$$a_{s,t,k} + r_{s,t,k} = d_{s,t,k}, \qquad \forall s = 1, \ldots, S, \quad t = 1, \ldots, T, \quad k = 1, \ldots, K, \tag{4.11}$$

where $a_{s,t,k}$ denotes the accepted patients of type k in arrival scenario s at time t, and $r_{s,t,k}$ denotes the rejected patients.

The decision or control variable is the acceptance limit $u_{t,k}$, which is defined as the maximum number of patients of type k to be accepted (taken into treatment) at time t. Thus, the following constraint needs to be fulfilled:

$$a_{s,t,k} \leq u_{t,k}, \qquad \forall s = 1, \ldots, S, \quad t = 1, \ldots, T, \quad k = 1, \ldots, K. \tag{4.12}$$

We define $\mathit{OldPat}_{t_1,k}$ as the number of patients of type k who are already in treatment at time t_1: ($\forall t_1 = 1, \ldots, T, \forall k = 1, \ldots, K$) by

$$\mathit{OldPat}_{t_1,k} \geq \sum_{l=1}^{L} \sum_{t_2=t_1-\mu_k+\Lambda_{k,l}+1}^{t_1-1} \lambda_{k,l} \cdot u_{t_2,k}. \tag{4.13}$$

The variable $\mathit{UsedDis}_{t,k,m}$ denotes the number of expected units of discipline m which are in use by patient type k at time t: ($\forall t = 1, \ldots, T, \forall k = 1, \ldots, K, \forall m = 1, \ldots, M$) and is defined by

$$\mathit{UsedDis}_{t,k,m} = (u_{t,k} + \mathit{OldPat}_{t,k}) \cdot c_{k,m}. \tag{4.14}$$

The total number of used discipline quantities at each time t is constrained by the available capacity modelled as

$$\sum_{k=1}^{K} \mathit{UsedDis}_{t,k,m} \leq \mathit{cap}(t,m), \qquad \forall t = 1,\ldots,T, \quad m = 1,\ldots,M. \tag{4.15}$$

Finally, we have some non-negativity and integrality conditions

$$u_{t,k}, \mathit{UsedDis}_{t,k,m} \in \mathbb{N}, \qquad \forall s,t,k,m, \tag{4.16}$$

$$u_{t,k}, a_{s,t,k}, r_{s,t,k} \geq 0, \qquad \forall s,t,k,m. \tag{4.17}$$

Since we start with an empty system, we need to force

$$u_{t,k} = 0, \qquad \forall t \leq 0, \quad \forall k, \tag{4.18}$$

$$a_{s,t,k} = 0, \qquad \forall t \leq 0, \quad \forall s,k, \tag{4.19}$$

$$\mathit{OldPat}_{1,k} = 0, \qquad \forall k. \tag{4.20}$$

As output of the optimisation model, we obtain the optimal acceptance policy $\hat{u}_{t,k}$ for each patient type at each time. We can further retrieve the expected capacity utilisation per discipline and time, as well as the resulting rejections for each patient arrival scenario d_s.

Patient scheduling model with a queue per patient type

In a system with a queue no patient is completely rejected. Patients are entering their designated queue (per patient type) and within each queue we work with a FCFS acceptance policy only constrained by the acceptance limit $u_{t,k}$ per time and patient type. To model the queue we need to introduce an additional variable $Q_{s,t,k}$, which denotes the queue size in scenario s at time t for patient type k.

The accept-reject constraints in (4.11) have to be extended with the queue variable. Patients which are not taken into service at time $t-1$, represented by $r_{s,t-1,k}$, are now blocked from the start of treatment at $t-1$ but remain in the system ($\forall s = 1,\ldots,S, \forall t = 1,\ldots,T, \forall k = 1,\ldots,K$). Hence, we have

$$Q_{s,t,k} = r_{s,t-1,k} + d_{s,t,k}, \tag{4.21}$$

$$Q_{s,t,k} = a_{s,t,k} + r_{s,t,k}. \tag{4.22}$$

So the full program with a queue takes the following form:
objective function

$$\text{minimise} \quad \sum_{s=1}^{S} \pi_s \cdot \sum_{t=1}^{T} \sum_{k=1}^{K} \alpha_k \cdot r_{s,t,k} \tag{4.23}$$

subject to $(\forall s = 1, \ldots, S, \forall t = 1, \ldots, T, \forall k = 1, \ldots, K, \forall m = 1, \ldots, M)$

$$\begin{aligned}
Q_{s,t,k} &= r_{s,t-1,k} + d_{s,t,k} && (4.24)\\
Q_{s,t,k} &= a_{s,t,k} + r_{s,t,k} && (4.25)\\
a_{s,t,k} &\leq u_{t,k} && (4.26)\\
\mathit{OldPat}_{t_1,k} &\geq \sum_{l=1}^{L} \sum_{t_2=t_1-\mu_k+\Lambda_{k,l}+1}^{t_1-1} \lambda_{k,l} \cdot u_{t_2,k} && (4.27)\\
\mathit{UsedDis}_{t,k,m} &= (u_{t,k} + \mathit{OldPat}_{t,k}) \cdot c_{k,m} && (4.28)\\
\sum_{k=1}^{K} \mathit{UsedDis}_{t,k,m} &\leq cap(t,m) && (4.29)\\
u_{t,k}, \mathit{UsedDis}_{t,k,m} &\in \mathbb{N} && (4.30)\\
u_{t,k}, a_{s,t,k}, r_{s,t,k}, Q_{s,t,k} &\geq 0 && (4.31)
\end{aligned}$$

For a start with an empty system at $t = 1$, we need to force additional constraints

$$\begin{aligned}
u_{t,k} &= 0, \qquad \forall t \leq 0, \quad \forall k,\\
a_{s,t,k} &= 0, \qquad \forall t \leq 0, \quad \forall s, k,\\
\mathit{OldPat}_{1,k} &= 0, \qquad \forall k,\\
r_{s,0,k} &= 0, \qquad \forall s, k.
\end{aligned}$$

The current formulation does not distinguish between waiting times of patients in the queue, i.e., there is no difference between the case of ten patients waiting one week or one patient waiting ten weeks. Obviously, we would like to distinguish between these two cases. Therefore, we extend the model so that we can differentiate waiting patients to a certain level. Waiting patients are divided into two groups, patients waiting one week and patients waiting two weeks or more.

We introduce an additional variable $\mathit{LongWait}_{s,t,k}$, which denotes the number of patients of type k at time t in scenario s who have been waiting at least at time $t-1$ and t. This is mathematically expressed by

$$\mathit{LongWait}_{s,t,k} = \max\{0, r_{s,t-1,k} - a_{s,t,k}\}. \tag{4.32}$$

In order to solve the problem with standard MIP solving techniques, we need to linearise the maximum constraint at the cost of additional variables. Hence, Equation (4.32) is rewritten as ($\forall s = 1, \ldots, S, \forall t = 1, \ldots, T, \forall k = 1, \ldots, K$)

$$\begin{aligned}
\mathit{LongWait}_{s,t,k} - \hat{z}_{s,t,k} &= r_{s,t-1,k} - a_{s,t,k}, && (4.33)\\
\mathit{LongWait}_{s,t,k} - \check{z}_{s,t,k} &= 0, && (4.34)\\
0 \leq \hat{z}_{s,t,k} \leq (1 - z_{s,t,k}) \cdot \kappa, \qquad & 0 \leq \check{z}_{s,t,k} \leq z_{s,t,k} \cdot \kappa, && (4.35)\\
z_{s,t,k} &\in \{0, 1\}, && (4.36)
\end{aligned}$$

with κ being a sufficiently large constant. The optimisation model is now able to distinguish between patients waiting only one time stage in the queue or two and more. A further differentiation of waiting times is possible by a repetition of the same approach and at the cost of introducing additional variables.

The problem formulation enables us to solve the problem with respect to two objective functions:

Objective 1: Minimising the number of waiting patients in the queue:

$$\text{minimise} \quad \sum_{s=1}^{S} \pi_s \cdot \sum_{t=1}^{T} \sum_{k=1}^{K} \alpha_k \cdot r_{s,t,k}. \tag{4.37}$$

Objective 2: Minimising the number of patients in queue waiting for two time stages or longer:

$$\text{minimise} \quad \sum_{s=1}^{S} \pi_s \cdot \sum_{t=1}^{T} \sum_{k=1}^{K} \alpha_k \cdot \mathit{LongWait}_{s,t,k}. \tag{4.38}$$

Both objectives are subject to the following constraints:
$(\forall s = 1, \ldots, S, \forall t = 1, \ldots, T, \forall k = 1, \ldots, K, \forall m = 1, \ldots, M)$

$$
\begin{aligned}
Q_{s,t,k} &= r_{s,t-1,k} + d_{s,t,k} && (4.39)\\
Q_{s,t,k} &= a_{s,t,k} + r_{s,t,k} && (4.40)\\
a_{s,t,k} &\leq u_{t,k} && (4.41)\\
\mathit{LongWait}_{s,t,k} - \hat{z}_{s,t,k} &= r_{s,t-1,k} - a_{s,t,k} && (4.42)\\
\mathit{LongWait}_{s,t,k} - \check{z}_{s,t,k} &= 0 && (4.43)\\
0 \leq \hat{z}_{s,t,k} \leq (1 - z_{s,t,k}) \cdot \kappa, &\quad 0 \leq \check{z}_{s,t,k} \leq z_{s,t,k} \cdot \kappa && (4.44)\\
z_{s,t,k} &\in \{0,1\} && (4.45)\\
\mathit{OldPat}_{t_1,k} &\geq \sum_{l=1}^{L} \sum_{t_2=t_1-\mu_k+\Lambda_{k,l}+1}^{t_1-1} \lambda_{k,l} \cdot u_{t_2,k} && (4.46)\\
\mathit{UsedDis}_{t,k,m} &= (u_{t,k} + \mathit{OldPat}_{t,k}) \cdot c_{k,m} && (4.47)\\
\sum_{k=1}^{K} \mathit{UsedDis}_{t,k,m} &\leq \mathit{cap}(t,m) && (4.48)\\
u_{t,k}, \mathit{UsedDis}_{t,k,m} &\in \mathbb{N} && (4.49)\\
u_{t,k}, a_{s,t,k}, r_{s,t,k}, Q_{s,t,k} &\geq 0 && (4.50)
\end{aligned}
$$

4.4.2 Numerical Examples

In this section, we evaluate the performance of our stochastic programming model by numerical experiments. We also evaluate the complexity of the models under both the objectives (4.37) and (4.38). The computation is performed by FICO Xpress-IVE 1.20.01 (optimizer version 20.00.05). We first start with describing our input data.

Input data

As illustrative example, we consider a model with 2 patient types, 3 treatment disciplines, a planning period of 10 weeks, and 3 demand scenarios. The estimated treatment duration per patient type is given by the vector μ, the perturbation factors by Λ with corresponding probability λ, and the available capacity by c_t. The number of perturbations L and the corresponding values and probabilities need to be estimated from historical patient data; for the sake of illustration we work with $L = 3$.

$$K = 2, \qquad M = 3, \qquad T = 10, \qquad S = 3,$$

$$\mu = \begin{pmatrix} 2 & 3 \end{pmatrix}, \quad \Lambda = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}, \quad \lambda = \begin{pmatrix} 0.3 & 0.4 & 0.3 \\ 0.25 & 0.5 & 0.25 \end{pmatrix}.$$

The available capacity per discipline is equal for all times stages t and given by

$$cap_{t,\cdot} = \begin{pmatrix} 40 & 40 & 10 \end{pmatrix}, \qquad \forall t = 1, \ldots, T.$$

The patient type $\times$ discipline matrix is

$$c = \begin{pmatrix} 1 & 3 & 1 \\ 2 & 2 & 0 \end{pmatrix}.$$

The patient arrival scenarios are generated by Monte Carlo simulation, where the arrival process of patients of type one (k=1) is assumed to be Poisson(3) distributed and the arrivals of patients of type 2 (k=2) are assumed to follow a Poisson(5) distribution.

Results

First, we solve the stochastic programming problem with no waiting queues, as described in Section 4.4.1. The problem is solved for several S values, in the range between 3 and 160. The results of the test are shown in Figure 4.5.

The objective values (objval) computed by the SP model (4.10) are compared with simulation results. The simulated objective values are obtained by applying the computed acceptance levels u in a patient simulation tool and iterating 10,000 times. We find that the best acceptance level is already computed with $S = 20$, as seen in the fourth plot which compares the simulation results with the best simulation result over all test cases. The gap between the computed and simulated objective value is generally decreasing in the number of considered scenarios S. The computation times naturally increase in S. However, we observe only a small growth until $S = 130$ and afterwards a larger increase in computation time.

The optimal S depends on the actual considered problem and the choice of S requires some testing. A common way to reduce the number

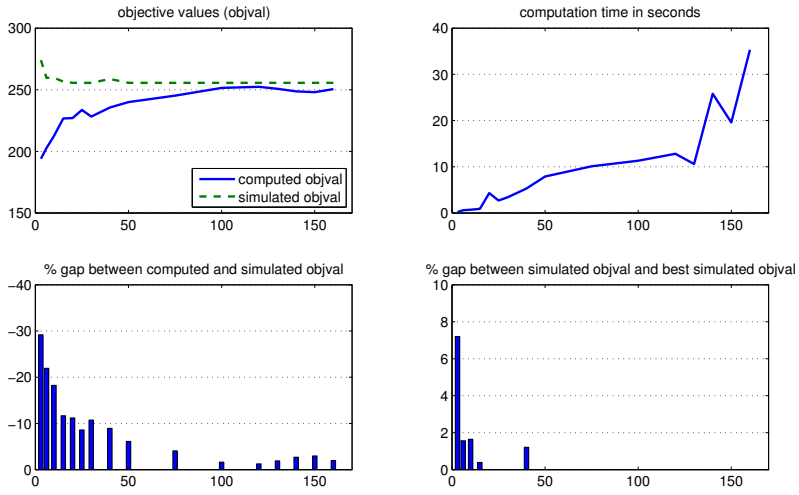

Figure 4.5: Results of SP without waiting queues for different S values.

of considered scenarios in the optimisation model without losing distributional information, is to apply scenario reduction algorithms, as discussed in Heitsch and Römisch [**62**]. Such algorithms create scenario trees by deleting scenarios with small Kantorovic distances to its neighbours and add their probability to the closest neighbour scenario. Such algorithm and their application are out of the scope of this chapter. In the following, we will work with $S = 10$.

Let us now concentrate on solving the SP problem with waiting queues. We start with the objective function (4.37), which minimises the total number of patients waiting in the queue.

$$\begin{array}{ll} \# \text{ constraints} = 1540, & \# \text{ variables} = 1525, \\ \text{Obj Value} = 742, & \text{comp. time} = 41.4 \text{ sec.} \end{array}$$

The results are shown in Table 4.3–4.5. The expected waiting times for patients of type one is 1.46 time stages and for patients of type two 0.58 time stages. The distribution of the estimated patients per waiting time are given in Table 4.4. The expected capacity utilisation per discipline and time is given in Table 4.5.

t	$u_{t,1}$	$u_{t,1}$	AQS $k=1$	AQS $k=2$	ANLW $k=1$	ANLW $k=2$
1	4	6	0	0.7	0	0
2	2	6	0.6	0.7	0	0
3	2	2	1.1	4.2	0	0.1
4	2	6	1.8	2.6	0	0.3
5	2	5	2.6	2.5	0.4	0.5
6	2	4	4.1	2.6	1.1	0.8
7	2	4	5.5	4.1	2.6	0.7
8	2	5	6.6	3.9	3.7	0.9
9	2	5	7.8	3.6	4.8	1.1
10	0	7	11.1	2.6	7.8	0.9

Table 4.3: Results of optimisation run 1, w.r.t. objective (4.37).
Abbreviation: AQS - average queue size, ANLW - average number of long waiting patients

If we solve the SP problem with respect to objective function (4.38) - minimising the number of long waiting patients:

$$\begin{array}{ll} \#\text{ constraints} = 1540, & \#\text{ variables} = 1525, \\ \text{Obj Value} = 247.6, & \text{comp. time} = 530 \text{ sec.} \end{array}$$

The results are shown in Table 4.6-4.8. The computation time increases by almost a factor of 13, compared to objective (4.37). This can be explained by the fundamental increase in complexity, because the $LongWait$ variables and thus the binary z variables are now directly influencing the objective function. The expected waiting times for patients of type one is 1.26 time stages and for patients of type two 0.85 time stages. The distribution of the estimated patients per waiting time are given in Table 4.7. The expected capacity utilisation per discipline and time is given in Table 4.8.

The performance of the stochastic programming algorithm is very good with low running times. In fact, the algorithm is scalable enough for practical purposes, whereas the Markov decision problem is already intractable for this example.

	patient type $k = 1$	patient type $k = 2$
wait 0:	2.48	8.45
wait 1:	3.07	5.64
wait 2:	1.82	1.62
wait 3:	1.22	0.07
wait 4:	0.56	0
wait 5:	0.2	0
wait 6:	0	0
wait 7:	0	0
wait 8:	0	0
wait 9:	0	0
wait 10:	0	0

Table 4.4: Expected number of patients and waiting time, generated by the results of optimisation run 1 - objective (4.37).

4.5 Conclusions and further research

In this chapter we have discussed and studied a model for home and rehabilitation care, used to prioritise admissions of patients of different types. We formulated the model as a Markov decision problem. Because of the large size of the state space, approximations are necessary to obtain results. The home care model is a special case of the more general model for rehabilitation care. The difference lies in the fact that for home care there is only one specialty. For this case we were able to prove some monotonicity properties of the value function, and the structure of the optimal policy for some cases. The experiments showed that a simple trunk reservation model gives very good results. Both for this heuristic and for the optimal policy, computation times were quite short. For obtaining the optimal policy parallel computing was used to handle to the large memory requirements.

For the more general model with more than one specialty, there are no longer any nice structural properties. Neither will obtaining the opti-

t	$m = 1$	$m = 2$	$m = 3$
1	0.31	0.42	0.23
2	0.54	0.7	0.31
3	0.59	0.78	0.36
4	0.71	0.89	0.36
5	0.75	0.93	0.37
6	0.76	0.94	0.38
7	0.73	0.92	0.39
8	0.74	0.93	0.39
9	0.79	0.99	0.39
10	0.82	0.92	0.2

Table 4.5: Expected capacity utilisation per discipline, generated by the results of optimisation run 1 - objective (4.37).

mal policy be possible, because parallelisation will not reduce the memory requirements enough. Instead, we studied two different approximation approaches, that both show good results.

The first approach uses stochastic programming. We have shown that the stochastic programming formulation works well and has the potential to scale while retaining short computation times. The second method uses approximate dynamic programming with good results. This is also scaleable in the problem instance. In both cases simulations were used to compare the performances with, as deriving an optimal policy is not possible.

There are some additions to the model that are interesting to address in future work. One of these is that in practice patient needs may change over time while they are in service. In general, patients need less intensive treatment when their condition progressively improves. Adding this to the model would mean that patients are allowed to change their type while residing in the system.

An extension that would be especially useful for the home care problem is that of different skill levels. This means that there are employees

t	$u_{t,1}$	$u_{t,1}$	AQS $k=1$	AQS $k=2$	ANLW $k=1$	ANLW $k=2$
1	4	6	0.4	1.9	0	0
2	2	6	0.6	1	0	0
3	2	2	1.4	4.7	0	0.1
4	2	6	1.9	3.3	0	0.3
5	2	5	2.7	2.8	0.4	0.5
6	2	4	4.1	2.6	1.1	0.8
7	2	4	5.7	4.3	2.6	0.7
8	3	4	5.8	5.1	2.9	1.4
9	3	3	6.3	7	3.2	2.5
10	3	4	6.7	7.8	3.4	3

Table 4.6: Results of optimisation run 2, w.r.t. objective (4.38).
Abbreviation: AQS - average queue size, ANLW - average number of long waiting patients

with different skill levels, and patients need a minimum skill level but personnel with a higher level can also perform that care, if at a higher cost. This works differently from the general model for rehabilitation care, because in that case each type of care can only be performed by one specialty, so at the moment this is not taken into account.

Another point to make the model more suitable for practice would be to consider other performance measures, for example the percentage of patients that wait more than a certain length of time. This might be a better target for the optimal policy, since the mean waiting time gives no information on the variability in the waiting times for individual patients.

	patient type $k = 1$	patient type $k = 2$
wait 0:	2.08	5.48
wait 1:	4.16	7.62
wait 2:	1.88	2.28
wait 3:	1.06	0.4
wait 4:	0.16	0
wait 5:	0	0
wait 6:	0	0
wait 7:	0	0
wait 8:	0	0
wait 9:	0	0
wait 10:	0	0

Table 4.7: Expected number of patients and waiting time, generated by the results of optimisation run 2 - objective (4.38).

t	$m = 1$	$m = 2$	$m = 3$
1	0.24	0.33	0.19
2	0.53	0.69	0.32
3	0.58	0.76	0.35
4	0.71	0.89	0.37
5	0.76	0.95	0.38
6	0.78	0.98	0.39
7	0.74	0.93	0.38
8	0.72	0.95	0.48
9	0.66	0.92	0.53
10	0.68	0.96	0.57

Table 4.8: Expected capacity utilisation per discipline, generated by the results of optimisation run 2 - objective (4.38).

Time constraints in emergency departments

5

Emergency departments (EDs) are important to a hospital for a number of reasons. A large part of the patients enter the hospital after a visit to the ED. They arrive without any appointment, and require more prompt attention than elective patients. Good access to the ED is both important for the patient and for the hospital as ED congestion leads to patients abandoning (or leaving without being seen) and ambulances rerouting to other hospitals.

One of the most important factors of patient satisfaction is the time they spend waiting in the emergency department. Also patient safety and personnel productivity are heavily influenced by the waiting time in the ED, see for example Derlet and Richards [**36**]. In many countries, emergency departments have a set-hour target in which they have to treat patients and then either admit them to a ward or discharge them. In the Netherlands this target is four hours. In general, EDs distinguish between two or more classes of patients with different emergency categories, ranging from patients who have to be treated immediately to those who can wait for a longer period of time. Upon arrival at the hospital, a triage process is used to decide to which emergency class a patient belongs. Every class of patients can have a different target for the time in which the patient should be seen by a doctor, and not meeting this target can be worse for one class of patients than for another class. EDs must prioritise the patients of all categories in the right way considering their waiting time targets, while also keeping the total time spent in the ED within the time limit, with limited available resources.

This leads to a prioritisation problem: which patient should be treated next when a doctor finishes the treatment of a patient? This prioritisation problem in itself is not unusual; there are many situations where this occurs, both in health care and in other settings such as call centres. Therefore this problem has been widely studied. What distinguishes the ED setting from most other settings where prioritisation is an issue, is the fact that it is not the mean waiting time of each patient class that is important but instead the fraction of patients from each class that do not start their treatment on time.

The mathematical modelling literature on patient prioritisation in EDs is rather scarce due to the complexity of the problem. There are some pa-

pers in which data analysis is performed to study how well EDs perform regarding their waiting times and the four-hour target (see, e.g., Locker and Mason [**78**]). The complexity in mathematical modelling lies within the fact that many decision models cannot take tail probabilities of the waiting time into account, rendering them void for realistic ED modelling. In this chapter, we overcome these difficulties by combining several techniques within the framework of Markov decision processes (MDP).

In the resulting Markov decision problem the goal is to minimise the fraction of patients that is not taken into service within the target time. To do this, it would be most effective to keep information on the waiting time of all the patients in the state space. However, this is not possible in practice. Instead, we use the state space description used by Koole et al. [**74**], where the state describes either the number of available servers or the current waiting time of the patient first in line. So we leave the information about the other patients out of the state space. This description of the state makes it possible to formulate the problem as a Markov decision process, and allows us to optimise the desired criterion. This we can do by including costs whenever a patient that starts treatment has waited too long, and no costs when the waiting time falls within the set target.

With this formulation, we find a closed-form expression for a system with a single patient type by viewing the Poisson equations as difference equations. Then we can use this expression in a one-step policy improvement approach. This approach has been shown to give good results in cases where a good approximation of the value function is available, see for example Bhulai [**19**] and Haijema and Van der Wal [**59**].

The outline of this chapter is as follows: in Section 5.1 we analyse a model with a single patient type and multiple servers, and derive an expression for the value function. Then in Section 5.2 we use this result to give a near-optimal policy. We demonstrate the performance of the solution with some numerical examples. Finally in Section 5.3 we discuss the results and directions for possible future work.

5.1 Analysis of a single queue

As the starting point for finding a good admission policy in the case with several patient types, we first provide a detailed description and analysis for a model with one patient type.

We are interested in dynamic control problems in which the objective of the system is to constrain the waiting time W of customers, more precisely, the service level in the system is defined as to minimise $\mathbb{P}(W > \alpha)$. Most standard techniques in stochastic optimisation are not well-equipped to handle such problems. Therefore, we present an alternative computational method for the calculation of the service level in this isolated Erlang delay model. This derivation will be a key building block for more complex decision models in the next sections. The alternative method is based on the analysis of the waiting time of the customer that is at the head of the queue using the Erlang approximation (EA), see Bekker et al. [**15**] and Nielsen et al. [**85**].

Let us first consider the case where our multi-server queue is non-empty. The main idea is that we analyse the waiting time of the customer that is first in line (FIL). This so-called FIL-process is linearly increasing until a service completion occurs, which happens with rate $s\mu$. Upon a service completion, the FIL-process has a negative jump as the second customer in line (if any) now becomes the first customer in line. So the jump size is determined by the inter-arrival time. Note that the FIL-process is a Markov process.

5.1.1 Dynamic programming

To cast the model into a dynamic programming framework, we discretise time into periods of length $1/\gamma$. The parameter γ governs the accuracy of the discretisation. The FIL waiting time now increases by 1 at an exponential rate γ, where state $k \in \{1, \ldots\}$ represents that the customer at the head of the queue has waited $k-1$ phases. Upon service completion, it follows from properties of the Poisson arrival process (see Nielsen et al. [**85**]) that the Markov chain jumps to state y according to

$$p_{x,y} = \begin{cases} 1 - \sum_{h=0}^{x-1} \left(\frac{\lambda}{\lambda+\gamma}\right) \left(\frac{\gamma}{\lambda+\gamma}\right)^h, & y = 0, \\ \left(\frac{\lambda}{\lambda+\gamma}\right) \left(\frac{\gamma}{\lambda+\gamma}\right)^{x-y}, & y = 1, \ldots, x. \end{cases} \tag{5.1}$$

Now, consider the case that there are no customers in line. We extend the state space to $\mathcal{X} = \{-s, -(s-1), \ldots, 0, 1, \ldots\}$ to take the number of occupied servers into account. Here, state $x \in \{-s, \ldots, 0\}$ denotes that there are $s + x$ servers occupied and no customers are waiting. The transition rates of the resulting Markov process can be found in Nielsen et al. [**85**].

Let n represent the acceptable waiting time, e.g., n is determined as $\gamma\alpha$. Then, the cost of a customer exceeding his acceptable waiting time can be expressed in the state variable by $c(x) = \mathbb{1}\{x > n\}$. Hence, the triplet $(\mathcal{X}, p, c)$ enables us to use techniques from dynamic programming for optimal control in problems where the waiting time should not exceed some threshold with a certain probability.

Let $V(x)$ be a real-valued function defined on the state space. This function will play the role of the relative value function, i.e., the asymptotic difference in total costs that results from starting the process in state x instead of some reference state. Furthermore, let g denote the long-term average costs. The dynamic programming Poisson equations can then be formulated by

$$g + \tau V(x) = \lambda V(x+1) + (x+s)\mu V(x-1) + (\tau - \lambda - (x+s)\mu)V(x), \quad (5.2)$$

for $-s \leq x \leq 0$. For states $x \geq 1$, we have

$$g + \tau V(x) = \gamma V(x+1) + s\mu \left(\mathbb{1}_{\{x>n\}} + \sum_{y=0}^{x} p_{x,y} V(y) \right) + (\tau - \gamma - s\mu)V(x). \quad (5.3)$$

Note that τ is the uniformisation constant, defined by $\tau = \max\{\lambda, \gamma\} + s\mu$. Uniformising is equivalent to adding dummy transitions (from a state to itself) such that the rate out of each state is constant (see Section 11.5 of Puterman [**93**]).

5.1.2 Performance analysis

In this subsection, we investigate the accuracy of the discretised approximation of the state space as a function of γ. Specifically, we derive the probability that the FIL process upon service initiation is in a state of at least n, approximating the probability that the waiting time is at least n/γ. Let W_{app} be the approximation of the waiting time by assuming that the waiting time in some state $j \geq 0$ is deterministic and identical to j/γ. This differs from Nielsen et al. [**85**], where the waiting time in state j is approximated by an Erlang random variable. Our approximation is considerably simpler.

The following proposition gives the tail distribution of W_{app}, enabling us to determine an appropriate value for γ. Large values of γ yield better

approximations, but lead to longer computation times. The proposition below can help in making this trade-off.

5.1.1. Proposition. *For the M/M/s queue with $\rho < 1$, the tail probability of the waiting time is approximated by*

$$\mathbb{P}(W_{\text{app}} > \alpha) = \frac{1}{1-\rho}\frac{(\lambda+\gamma)\pi(0)}{\gamma+\lambda\pi(0)}\left(1+\frac{-s\mu(1-\rho)\alpha}{\alpha(s\mu+\gamma)}\right)^{\alpha\gamma},$$

where

$$\pi(0) = \frac{(s\rho)^s}{s!}\left(\sum_{i=0}^{s-1}\frac{(s\rho)^i}{i!} + \frac{(s\rho)^s}{s!}\frac{\gamma+\rho\lambda}{\gamma}\frac{1}{1-\rho}\right)^{-1}. \tag{5.4}$$

Proof. Let $\pi(\cdot)$ be the stationary distribution of the Markov chain that describes the waiting time of the patient first in line. Using a level crossing argument and the geometric form of $p_{x,y}$ it follows that, for $i \geq 1$,

$$\gamma\pi(i) = s\mu\sum_{j=i+1}^{\infty}\pi(j)\left(\frac{\gamma}{\lambda+\gamma}\right)^{j-i}. \tag{5.5}$$

The left-hand side corresponds to upcrossings of level i, whereas the right-hand side corresponds to downcrossings. From (5.5) it is straightforward to show that $\pi(i) = \pi(1)\left(\frac{\lambda+\gamma}{s\mu+\gamma}\right)^i$. Similarly, using level crossings again, we have

$$\lambda\pi(0) = s\mu\sum_{j=1}^{\infty}\pi(j)\left(\frac{\gamma}{\lambda+\gamma}\right)^j = \frac{s\mu+\gamma}{\lambda+\gamma}\gamma\pi(1),$$

where the second equality follows from substituting the result for $\pi(i)$ given above. The analysis of the states $\{-s,\ldots,0\}$ is identical to the ordinary M/M/s queue. Finally, using normalisation, we obtain that, for $i \geq 1$,

$$\pi(i) = \frac{\lambda}{\gamma}\pi(0)\left(\frac{\lambda+\gamma}{s\mu+\gamma}\right)^i,$$

with $\pi(0)$ as in (5.4).

For the waiting time, we need to consider the states of the Markov chain at epochs of service initiations, which can either be arrival epochs with available servers or service completion epochs with customers in the queue. Let $\tilde{W}$ denote the state of the system just before a service initiation.

Using the conditioning also used by Nielsen et al. **[85]**, we get, after some rewriting, that

$$\begin{aligned}
\mathbb{P}(\tilde{W} > n) &= \sum_{i=n+1}^{\infty} \frac{s\mu\pi(i)}{\lambda \sum_{j=-s}^{-1} \pi(j) + s\mu \sum_{j=1}^{\infty} \pi(j)} \\
&= s\mu\frac{\lambda}{\gamma}\pi(0) \sum_{i=n+1}^{\infty} \left(\frac{\lambda+\gamma}{s\mu+\gamma}\right)^i \times \frac{1}{\lambda - \lambda\pi(0) + (s\mu-\lambda)\sum_{j=1}^{\infty}\pi(j)} \\
&= s\mu\frac{\lambda}{\gamma}\pi(0)\frac{\lambda+\gamma}{s\mu-\lambda}\left(\frac{\lambda+\gamma}{s\mu+\gamma}\right)^n \times \frac{1}{\lambda - \lambda\pi(0) + \lambda\pi(0)(\lambda+\gamma)/\gamma} \\
&= \frac{1}{1-\rho}\frac{(\lambda+\gamma)\pi(0)}{\gamma+\lambda\pi(0)}\left(\frac{\lambda+\gamma}{s\mu+\gamma}\right)^n.
\end{aligned}$$

Finally, we take $n = \gamma\alpha$ yielding the result. ∎

From Proposition 5.1.1 it follows that the approximating waiting time converges to the waiting time in the M/M/s queue as $\gamma \to \infty$. Moreover, it is not required that n is a multiple of $1/\gamma$. Thereby, we may obtain an approximation of the tail probability for any $\alpha \geq 0$.

5.1.2. Remark. In **[85]** the waiting time distribution for the EA has only been derived in closed form for the M/M/1 case. For comparison, we here give the result for the multi-server queue. Let W_{EA} denote the waiting time for the EA. Assuming that the waiting time in state i upon a service initiation has an Erlang distribution consisting of i periods, each with rate γ, we obtain from a similar analysis as in the proof of Proposition 5.1.1 that

$$\mathbb{P}(W_{\mathrm{EA}} > \alpha) = \frac{1}{1-\rho}\frac{(\lambda+\gamma)\pi(0)}{\gamma+\lambda\pi(0)}\exp\left(-\frac{\gamma}{s\mu+\gamma}s\mu(1-\rho)\alpha\right),$$

with $\pi(0)$ as in (5.4).

5.1.3 The value function

As a first step in the optimisation framework, we need to determine the relative value function $V(x)$ for every state $x \in \{-s, \ldots\}$. In this subsection, we derive a closed-form expression for $V(\cdot)$. In particular, we consider the cases with and without waiting customers separately. We first deal with the case of $-s \leq x \leq 0$.

5.1.3. Proposition. *The relative value $V(x)$ for $x = -s, \ldots, 0$ is given by*

$$V(x) = \frac{g}{\lambda} \sum_{i=1}^{x+s} \sum_{k=0}^{i-1} \frac{(i-1)!}{(i-k-1)!} \left(\frac{\mu}{\lambda}\right)^k.$$

Proof. This result follows directly from the relative value function of the M/M/s queue for $0 \leq x \leq s$ as given by Bhulai [**19**]. ∎

Now, consider the situation that there are customers waiting in the queue, i.e., $x \geq 1$. The relative value function $V(\cdot)$ then satisfies the following equation

$$\begin{aligned} g + \tau V(x) = \gamma V(x+1) \\ + s\mu \left(\mathbb{1}_{\{x>n\}} + \sum_{y=0}^{x} p_{x,y} V(y) \right) + (\tau - \gamma - s\mu) V(x), \end{aligned}$$

where $\mathbb{1}_{\{\}}$ is the indicator function. The first term on the right-hand side represents an increment in the waiting time of the first customer in line. The second term on the right-hand side corresponds to a service completion, where a penalty of 1 is incurred if the state is above n. In addition, the Markov chain then jumps to state y according to $p_{x,y}$. The final term is due to uniformisation.

5.1.4. Theorem. *For $x \geq 1$ the relative value function is*

$$\begin{aligned} V(x) = V(0) + \frac{g}{\gamma s\mu(1-\rho)^2} \left[\lambda x(\rho - 1) + (\lambda + \gamma) \left(\frac{s\mu + \gamma}{\lambda + \gamma} \right)^x - (\lambda + \gamma) \right] \\ + \frac{g}{\lambda} \left[\lambda - \gamma - \frac{\gamma}{\rho} \sum_{k=0}^{s-1} \frac{(s-1)!}{(s-k-1)!} \left(\frac{\mu}{\lambda}\right)^k \right] \frac{1}{\gamma(\rho - 1)} \\ \times \left[\frac{\gamma(\rho-1)}{s\mu + \gamma} \mathbb{1}_{\{x=0\}} - \rho + \frac{\lambda + \gamma}{s\mu + \gamma} \left(\frac{s\mu + \gamma}{\lambda + \gamma} \right)^x \right] \\ - \mathbb{1}_{\{x>n\}} \times \frac{1}{\gamma(1-\rho)^2} \left[\lambda(x - n - 1)(\rho - 1) \right. \\ \left. + (\lambda + \gamma) \left(\frac{s\mu + \gamma}{\lambda + \gamma} \right)^{x-n-1} - (\lambda + \gamma) \right] \end{aligned}$$

Proof. For convenience, we may here assume that $V(0) = 0$. The case that $V(0) \neq 0$ is directly obtained by adding $V(0)$ to the relative value function $V(x)$. First, rewrite the equation for $V(\cdot)$ as

$$g + (\gamma + s\mu)V(x) = \gamma V(x+1) + s\mu \left(\mathbb{1}_{\{x>n\}} + \sum_{y=0}^{x} p_{x,y} V(y) \right). \tag{5.6}$$

To determine $V(\cdot)$, we use generating functions. Define $G(z) = \sum_{x=0}^{\infty} V(x) z^x$ as the generating function of $V(\cdot)$. Note that for $x = 0$ we need to use Equation (5.2), because Equation (5.6) does not hold in this case. Multiplying both sides of (5.6) by z^x and summing yields

$$\begin{aligned} \frac{g}{1-z} - g + (\gamma + s\mu)G(z) - (\gamma + s\mu)V(0) &= \gamma \frac{G(z)}{z} - \gamma \frac{V(0)}{z} - \gamma V(1) \\ &+ \frac{s\mu z^{n+1}}{1-z} + s\mu \frac{\lambda}{\lambda+\gamma} \frac{1}{1-\gamma z/(\lambda+\gamma)} G(z) + s\mu \frac{\gamma z}{\lambda+\gamma-\gamma z} V(0), \end{aligned}$$

where the final terms follow from substituting the geometric form of $p_{x,y}$ and interchanging the summations. With $V(0) = 0$, then this gives

$$\begin{aligned} &\frac{g}{1-z} - g + (\gamma + s\mu)G(z) = \\ &\qquad \frac{\gamma}{z} G(z) - \gamma V(1) + \frac{s\mu z^{n+1}}{1-z} + s\mu \frac{\lambda}{\lambda+\gamma} \frac{1}{1-z\gamma/(\lambda+\gamma)} G(z). \end{aligned} \tag{5.7}$$

Rearranging terms gives

$$G(z) = \left(\frac{s\mu z^{n+1} - g}{1-z} + g - \gamma V(1) \right) \frac{z}{\gamma} \frac{\gamma z - \lambda - \gamma}{(z-1)(s\mu z + \gamma z - \lambda - \gamma)}. \tag{5.8}$$

The denominator of the final fraction on the right-hand side of (5.8) has two roots: 1 and $(\lambda+\gamma)/(s\mu+\gamma) < 1$. Using partial fraction expansion, we may rewrite (5.8) as

$$\begin{aligned} G(z) &= \left(\frac{s\mu z^{n+1}}{1-z} - \frac{g}{1-z} + g - \gamma V(1)\right)\frac{1}{\gamma(\lambda - s\mu)} && (5.9)\\ &\quad \times \left(\frac{\gamma(\lambda - s\mu)}{s\mu+\gamma} - \frac{\lambda}{1-z} + \frac{(\lambda+\gamma)s\mu}{s\mu+\gamma}\frac{1}{1-z(s\mu+\gamma)/(\lambda+\gamma)}\right) \\ &=: s\mu\frac{z^{n+1}}{1-z}\frac{1}{\gamma(\lambda-s\mu)}H(z) && (5.10)\\ &\quad +(g-\gamma V(1))\frac{1}{\gamma(\lambda-s\mu)}H(z) && (5.11)\\ &\quad -\frac{g}{1-z}\frac{1}{\gamma(\lambda-s\mu)}H(z) && (5.12) \end{aligned}$$

with $H(z)$ defined as the transform between the second pair of large brackets in (5.9). Note that the product of two transforms corresponds to convolution. Also observe that $z^{n+1}/(1-z)$ is the transform of a function that equals 1 on $\{n+1, n+2, \ldots\}$ and 0 elsewhere. Since $H(z)$ corresponds to a transform that is 0 on $\{\ldots, -1, 0\}$ it follows that $z^{n+1}/(1-z) \times H(z)$ is 0 on $\{0, \ldots, n\}$. The value function $V(x)$ for $x \in \{1, \ldots, n\}$ is thus completely determined by (5.11) and (5.12). We start with the inverse of (5.12). Applying inversion and working out the convolutions, we obtain, for $x \in \{1, \ldots, n\}$,

$$\begin{aligned} V(x) &= \frac{g}{\gamma(\lambda-s\mu)}\left(-\frac{\gamma(\lambda-s\mu)}{s\mu+\gamma} + \lambda\sum_{k=0}^{x} 1 - \sum_{k=0}^{x}\frac{(\lambda+\gamma)s\mu}{s\mu+\gamma}\left(\frac{s\mu+\gamma}{\lambda+\gamma}\right)^k\right) \\ &= \frac{g}{\gamma(\lambda-s\mu)} \\ &\quad \times \left(-\frac{\gamma(\lambda-s\mu)}{s\mu+\gamma} + \lambda(x+1) - \frac{\lambda+\gamma-(s\mu+\gamma)\left(\frac{s\mu+\gamma}{\lambda+\gamma}\right)^x}{\lambda-s\mu}(\lambda+\gamma)s\mu\right). \end{aligned}$$

Rewriting the above yields the first line of the expression in Theorem 5.1.4. Now, for $x > n$ we additionally require the inverse of (5.10). Similar to the inverse of (5.12), the inverse of (5.10), for $x > n$, reads

$$\frac{-s\mu}{\gamma(\lambda-s\mu)}\left(-\frac{\gamma(\lambda-s\mu)}{s\mu+\gamma} + \lambda\sum_{k=n+1}^{x} 1 - \sum_{k=n+1}^{x}\frac{(\lambda+\gamma)s\mu}{s\mu+\gamma}\left(\frac{s\mu+\gamma}{\lambda+\gamma}\right)^{x-k}\right).$$

For (5.11) finally, the inverse is

$$(g-\gamma V(1))\frac{1}{\gamma(\lambda-s\mu)}\left(\frac{\gamma(\lambda-s\mu)}{s\mu+\gamma}\mathbb{1}_{\{x=0\}}-\lambda+\frac{(\lambda+\gamma)s\mu}{s\mu+\gamma}\left(\frac{s\mu+\gamma}{\lambda+\gamma}\right)^x\right).$$

If we write the Poisson equation for $x=0$, we get

$$g+(\lambda+s\mu)V(0)=\lambda V(1)+s\mu V(-1).$$

Then we can use Proposition 5.1.3 to write $V(-1)$ as

$$V(-1)=V(0)-\frac{g}{\lambda}\sum_{k=0}^{s-1}\frac{(i-1)!}{(i-k-1)!}\left(\frac{\mu}{\lambda}\right)^k.$$

Using $V(0)=0$, we then have

$$g-\gamma V(1)=g-\frac{\gamma}{\lambda}(g-s\mu V(-1)).$$

If we put this into the inverse of (5.11) , rewriting and combining the three transforms, we obtain the final result. ■

5.2 One-step policy improvement

We have obtained expressions for the value function of a system with one type of patients for every state. The next step is to use these results to find near-optimal policies for cases with more patient types. We use one-step policy improvement for this purpose. This method needs an approximation for the value function in each state as a starting point. This is then used to make a decision in the improvement step. In this section we describe how we use the expressions derived in the previous section to approximate a system with two patient types, and give the optimality equations used in the improvement step.

We are considering a system with two patient types with separate queues. Both patient types arrive according to a Poisson process and have exponential service times. Let λ_1, λ_2 denote the arrival rates of patients of type 1 and 2 respectively, and let μ_1, μ_2 be the service rates of the patients. Note that the service times depend on the patient type, not on the servers. The patients have a target waiting time denoted by n_1, n_2 respectively, and

the costs of waiting longer than this target is weighed by w_1, w_2. Note that we use here n_i as the waiting time target expressed in the number of time units waited, which are exponentially distributed with parameter γ. To get the same result choose $n_i = \alpha_i \gamma$.

We have a group of S servers which can choose from both queues when they finish the treatment of a patient, or when there is an arrival and they are not occupied at that moment. It is also possible for a server to stay idle, even if there are patients waiting. This may happen for example if one class of patients has a very short target time, and the other one a much longer one. If there are no patients of the first type, it might be better to keep a server free in case a patient of the first type arrives.

To find an approximation of the value function in the case with two patient types, we have to incorporate the number of servers occupied by each type of patient into the state space. This is necessary because the service times differ with the patient type. So we define the state space $\mathcal{X} = (x_1, s_1, x_2, s_2)$, where x_i denotes the waiting time of the first patient in line of type i, and s_i denotes the number of servers occupied with patients of type i. So every time a patient is taken into service s_i increases by 1, and when a patient finishes his service s_i is decreased by 1. Note that all servers assigned to one of the queues are always occupied, so there is no need for the negative part of the state space as used previously. If a servers is idle, it is not assigned to either queue in the approximation, i. e. $s_1 + s_2 < S$.

Effectively this means that we are ignoring the servers that are not occupied by a patient. This makes it possible to assign less servers to the two queues than S even when there are patients waiting, or in other words, to keep a server idle. We define $V_{\lambda,\mu,n}(x, s)$ as the value function of a system with one queue, parameters λ, μ and n, x the waiting time of the patient first in line, and s available servers. Then the value function for the complete system is approximated by $V(x_1, s_1, x_2, s_2) = V_{\lambda_1,\mu_1,n_1}(x_1, s_1) + V_{\lambda_2,\mu_2,n_2}(x_2, s_2)$.

Now it remains to define the optimality equations for the improvement step. We have to distinguish four different cases based on the values of x_1 and x_2. First for $x_1 > 0$ and $x_2 > 0$ we have

$$\begin{aligned} V(x_1, s_1, x_2, s_2) = &\gamma H_{w,w}(x_1 + 1, s_1, x_2, s_2) + s_1\mu_1 H_{w,w}(x_1, s_1 - 1, x_2, s_2) \\ &+\gamma H_{w,w}(x_1, s_1, x_2 + 1, s_2) + s_2\mu_2 H_{w,w}(x_1, s_1, x_2, s_2 - 1) \\ &+(1 - 2\gamma - s_1\mu_1 - s_2\mu_2)V(x_1, s_1, x_2, s_2). \end{aligned} \tag{5.13}$$

For $x_1 > 0$ and $x_2 = 0$ we have

$$\begin{aligned} V(x_1, s_1, x_2, s_2) = &\gamma H_{w,nw}(x_1 + 1, s_1, x_2, s_2) + s_1\mu_1 H_{w,nw}(x_1, s_1 - 1, x_2, s_2) \\ &+\lambda_2 H_{w,w}(x_1, s_1, x_2 + 1, s_2) + s_2\mu_2 H_{w,nw}(x_1, s_1, x_2, s_2 - 1) \\ &+(1 - \gamma - s_1\mu_1 - \lambda_2 - s_2\mu_2)V(x_1, s_1, x_2, s_2). \end{aligned} \tag{5.14}$$

For $x_1 = 0$ and $x_2 > 0$ we have

$$\begin{aligned} V(x_1, s_1, x_2, s_2) = &\lambda_1 H_{w,w}(x_1 + 1, s_1, x_2, s_2) + s_1\mu_1 H_{nw,w}(x_1, s_1 - 1, x_2, s_2) \\ &+\gamma H_{nw,w}(x_1, s_1, x_2 + 1, s_2) + s_2\mu_2 H_{nw,w}(x_1, s_1, x_2, s_2 - 1) \\ &+ + (1 - \lambda_1 - s_1\mu_1 - \gamma - s_2\mu_2)V(x_1, s_1, x_2, s_2). \end{aligned} \tag{5.15}$$

For $x_1 = 0$ and $x_2 = 0$ we have

$$\begin{aligned} V(x_1, s_1, x_2, s_2) = &\lambda_1 H_{w,nw}(x_1 + 1, s_1, x_2, s_2)) + s_1\mu_1 V(x_1, s_1 - 1, x_2, s_2) \\ &+\lambda_2 H_{nw,w}(x_1, s_1, x_2 + 1, s_2) + s_2\mu_2 V(x_1, s_1, x_2, s_2 - 1) \\ &+(1 - \lambda_1 - s_1\mu_1 - \lambda_2 - s_2\mu_2)V(x_1, s_1, x_2, s_2). \end{aligned} \tag{5.16}$$

with $H_{w,w}(x_1, s_1, x_2, s_2) =$

$$\begin{cases} \min \Bigg\{ V(x_1, s_1, x_2, s_2) + P\mathbb{1}_{\{x_1>T \wedge x_2>T\}}, \\ \qquad \sum_{i=0}^{x_1} \left[p_{x_1,i} V(i, s_1+1, x_2, s_2) + w_1 \mathbb{1}_{\{x_1>n_1\}} \right], \\ \qquad \sum_{i=0}^{x_2} \left[p_{x_2,i} V(x_1, s_1, i, s_2+1) + w_2 \mathbb{1}_{\{x_2>n_2\}} \right] \Bigg\} & \text{if } s_1+s_2 < S, \\ V(x_1, s_1, x_2, s_2) & \text{otherwise,} \end{cases} \tag{5.17}$$

$H_{w,nw}(x_1, s_1, x_2, s_2) =$

$$\begin{cases} \min \Bigg\{ V(x_1, s_1, x_2, s_2) + P\mathbb{1}_{\{x_1>T \wedge x_2>T\}}, \\ \qquad \sum_{i=0}^{x_1} \left[p_{x_1,i} V(i, s_1+1, x_2, s_2) + w_1 \mathbb{1}_{\{x_1>n_1\}} \right] \Bigg\} & \text{if } s_1+s_2 < S, \\ V(x_1, s_1, x_2, s_2) & \text{otherwise,} \end{cases} \tag{5.18}$$

and finally $H_{nw,w}(x_1, s_1, x_2, s_2) =$

$$\begin{cases} \min \Bigg\{ V(x_1, s_1, x_2, s_2 + P\mathbb{1}_{\{x_1>T \wedge x_2>T\}}), \\ \qquad \sum_{i=0}^{x_2} \left[p_{x_2,i} V(x_1, s_1, i, s_2+1) + w_2 \mathbb{1}_{\{x_2>n_2\}} \right] \Bigg\} & \text{if } s_1+s_2 < S, \\ V(x_1, s_1, x_2, s_2) & \text{otherwise.} \end{cases} \tag{5.19}$$

Here the functions $H_{w,w}(x_1, s_1, x_2, s_2)$, $H_{w,nw}(x_1, s_1, x_2, s_2) =$ and $H_{nw,w}(x_1, s_1, x_2, s_2)$ define the possible actions you can take in each situation. For example in $H_{w,w}(x_1, s_1, x_2, s_2)$, which is used when there are patients waiting of both types, the possible actions are first to do

nothing, second to take a patient of type 1 into service, and finally to take a patient of type 2 into service. These actions are of course only possible if $s_1 + s_2 < S$, i.e., when there is a server available.

The value P can be considered a penalty for letting the waiting times increasing more than some maximum value T. This has been introduced so that the optimal policy will not be never to take any patient into service. Without this addition, that would be the optimal policy. The values for P should be significantly higher than the values for w_1 and w_2 to have the desired effect. Also T should be large enough not to influence the actual policy, so significantly above n_1 and n_2. For $p_{x_1,i}$ use Equation (5.1) with parameters λ_1 and μ_1, and for and $p_{x_2,i}$ with parameters λ_2 and μ_2.

Note that we have here given only the equations for a situation with two patient types. It is possible to generalise this to systems with more patient types using similar equations.

5.2.1 Numerical examples

In this section we illustrate the one-step improvement approach with some numerical results. We consider a system with two patient types as described in the previous section, for a few different parameter settings. All examples are chosen sufficiently small to actually compute the optimal policy and the associated long-term average costs. This means that we can compare the results for the one-step improved policy with the actual optimal values and see how much of an error we make when using the approximation.

See Table 5.1 for the results. We denote the long-term average costs of the optimal policy by $g*$ and those of the one-step improved policy by g^{ospi}. In all experiments we chose $\gamma = 50$ and $P = 10000$.

From the results we can see that the long-term average costs for the policy after one policy-improvement step are very close to the long-term average costs for the optimal policy. Of course the examples here are very small, but these examples indicate that the approach taken in this chapter would also give good results in larger problem instances, where computing the optimal policy is not practical any more.

S	λ_1	λ_2	μ_1	μ_2	n_1	n_2	w_1	w_2	g^{ospi}	g^*
2	1.0	1.0	3.0	2.0	10	10	1.0	1.0	0.3945	0.3990
2	2.0	1.0	3.0	2.0	25	50	1.0	1.0	0.3213	0.3217
2	2.0	1.0	3.0	2.0	25	50	1.0	5.0	0.3213	0.3219
2	2.0	1.0	3.0	2.0	25	100	1.0	1.0	0.3213	0.3217
3	2.0	1.0	3.0	2.0	10	10	1.0	5.0	0.5825	0.5831

Table 5.1: Numerical results of one-step policy improvement.

5.3 Conclusions and directions for further work

In this chapter we have presented a method for fractions of patients waiting longer than a certain goal, instead of the mean waiting times. This method is based on approximating the waiting times of all patients in the queues by keeping track of the patient who is first in line. We have derived an expression for the value function of a system with only one patient type, and demonstrated how this can be used to derive good policies for systems with more than one patient type. We have demonstrated the efficacy of this approach with some numerical examples.

In this thesis we focus on health care situations exclusively, but we can imagine more situations where a model like this could be interesting to apply. One obvious example is call centres, where patients are often ranked by importance.

The examples shown in the previous section are very small, and are meant only to show the efficacy of the one-step policy improvement approach. It would however be interesting to see what effect different parameter values would have on the long-term average costs of the system. Also the policy itself deserves attention in future work.

For further research within health care one possible extension could be an extra type of patient representing the severest type of emergency patient, that would preempt any ongoing treatment and cannot wait for even the shortest amount of time. This would be useful for example in the case of patients with heart failure. This could be done for example by adding a type with a very high weight and a very small waiting time target which will probably result in idling servers. This can also be done

by modifying the Poisson equations or the service time distribution of the other patients.

Another useful possibility is the inclusion of abandonments, or patients leaving without treatment. This is a problem often encountered with the less severe patients, and occurs more often the longer the waiting time is. This can be a performance measure in itself, and will of course change the dynamics of the system as well.

Conclusions

6

In the last four chapters we have discussed four different scheduling problems in healthcare. These were addressed using different methodologies.

In Chapter 2 we considered the problem of appointment scheduling, which occurs in many settings. We have presented some analytical results and an algorithm for finding the optimal schedule for the case with the limiting assumption that all patients are punctual in case they show up for their appointment. If this assumption is not realistic enough, we can use simulation-based methods to find good schedules. While there are no absolute guarantees about the quality of the solutions, the results are good enough for practical use.

Next, in Chapter 3 we have presented a method to analyse the bed demand for a given admission schedule, and described a method to find the optimal admission schedule to match bed demand to availability. This allows for a smoother demand for beds and improved performance of a hospital ward. Any special constraints can easily be taken into account in the optimisation method, and it is small enough to be implemented in something like an Excel spreadsheet. This, together with the flexibility of the method, makes is suitable for use in hospitals as it is, although the organisational side of implementation can be challenging. The results that have been achieved in one hospital look promising.

Then, in Chapter 4 the problem of prioritising patients for admission in home and rehabilitation care was solved using Markov decision theory. We used the number of time units available for care each week as the number of servers, and divided the patients into groups based on the number of time units of care they need per week. For rehabilitation care we needed to take a number of different server types into account as well. Our goal was to minimise the weighted average waiting time over all patient groups. Because of the large state space we needed to use approximation techniques, which gave good results.

Finally in Chapter 5 we look at a prioritisation problem again, but this time we optimise not the average waiting time but the fraction of patients who wait longer than some set threshold or target. This problem is often seen in emergency departments, but there are many other possible settings both within and outside of health care. To do this we use Markov decision theory again, while taking the waiting time for the patient who is first in

line as the state description. Then we use approximation techniques, because for several patient types the problem quickly becomes too large to solve directly. Small examples indicate that this gives near-optimal results. We are aware that the process at an emergency department usually consists of several steps and involves more than one health care professional, but we still believe that this model provides insight and policies for prioritising patients. It can also serve as a starting point for further analysis of a more complicated model.

We can see that in all cases significant improvements can be made by using techniques from operations research, and that the models can be flexible and generic enough to be used in practice. Sometimes assumptions are made that may not be completely realistic, or at least not in all cases, so there is still room for improvement. A good example of this is the assumption of stationary arrivals in Chapter 5, and the fact that the number of hours of care needed per week is fixed over time in Chapter 4. But in every case we were able to make less constrictive assumptions than previous models from the literature, making the models more realistic. And the models give good insight in any case, even when the assumptions don't hold completely.

The other big thing still remaining to do is to test these models out in practice. We were able to do this with the results from Chapter 3, with good results. The challenge in implementing the results often lies not only in having a good enough model, but also in making an actual lasting change in an organisation. This can be a challenge even when the results of the model are very clear and simple. This is a matter outside the scope of this thesis, but it is good to keep in mind when working with people from health care institutions on their problems. This should always be the ultimate goal when modelling health care processes.

Bibliography

[1] I Adan, J Bekkers, N Dellaert, J Vissers and X Yu, *Patient mix optimisation and stochastic resource requirements: A case study in cardiothoracic surgery planning*, Health Care Management Science **12** (2009), 129–141.

[2] O Z Aksin, M Armony and V Mehrotra, *The modern call-center: a multi-disciplinary perspective on operations management research*, Production and Operations Management **16** (2007), no. 6, 665–688.

[3] E Altman, Handbook of Markov Decision Processes, ch. Applications of Markov Decision Processes in Communication Networks, 489–536, Kluwer Academic Publishers, 2002, 489–536.

[4] E Altman, T Jiménez and G M Koole, *On optimal call admission control in a resource-sharing system*, IEEE Transactions on Communications **49** (2001), 1659–1668.

[5] S Andradóttir, *A global search method for discrete stochastic optimization*, SIAM Journal of Optimization **6** (1996), 513–530.

[6] S Andradóttir, *Accelerating the convergence of random search methods for discrete stochastic optimization*, ACM Transactions on Modeling and Computer Simulation **9** (1999), 349–390.

[7] S Andradóttir, *An overview of simulation optimization via random search*, Handbooks in Operations Research and Management Science: Simulation (S G Henderson and B L Nelson, eds.), Elsevier, 2006, 617–631.

[8] S Andradóttir, Simulation Optimization, 307–333, John Wiley & Sons, Inc., 2007, 307–333.

[9] S Asmussen, O Nerman and M Olsson, *Fitting phase type distributions via the EM algorithm*, Scandinavian Journal of Statistics **23** (1996), 419–441.

[10] N T J Bailey, *Operational research in medicine*, Operational Research Quarterly **3** (1952), 24–30.

[11] M A Begen and M Queyranne, *Appointment scheduling with discrete random durations*, Mathematics of Operations Research **36** (2011), 240–257.

[12] R Bekker, S Bhulai and P M Koeleman, *Time constraints in emergency departments*, Working paper.

[13] R Bekker and A M de Bruin, *Time-dependent analysis for refused admissions in clinical wards*, Annals of Operations Research **178** (2010), 45–65.

[14] R Bekker and P M Koeleman, *Scheduling admissions and reducing variability in bed demand.*, Health Care Management Science **14** (2011), 237–49.

[15] R Bekker, G M Koole, T B Nielsen and B F Nielsen, *Queues with waiting time dependent service*, Queueing Systems **68** (2011), 61–78.

[16] J Beliën and E Demeulemeester, *Building cyclic master surgery schedules with leveled resulting bed occupancy*, European Journal of Operational Research **176** (2007), 1185–1204.

[17] S Bertels and T Fahle, *A hybrid setup for a hybrid scenario: combining heuristics for the home health care problem*, Computers & Operations Research **33** (2006), 2866–2890.

[18] D P Bertsekas and J N Tsitsiklis, *Neuro-dynamic programming*, Athena Scientific, 1996.

[19] S Bhulai, *Dynamic routing policies for multi-skill call centers*, Probability in the Engineering and Informational Sciences **23** (2009), 75–99.

[20] J R Birge and M A H Dempster, *Stochastic programming approaches to stochastic scheduling*, Journal of Global Optimization **9** (1996), 417–451.

[21] D P Boldy and P C O'Kane, *Health operational research – a selective overview*, European Journal of Operational Research **10** (1982), 1 – 9.

[22] A A Borovkov, *On limit laws for service processes in multi-channel systems*, Siberian Mathematical Journal **8** (1967), 746–763.

[23] M Brahimi and D J Worthington, *Queueing models for out-patient appointment systems – a case study*, The Journal of the Operational Research Society **42** (1991), 733–746.

[24] S C Brailsford, P R Harper, B Patel and M Pitt, *An analysis of the academic literature on simulation and modelling in health care*, Journal of Simulation **3** (2009), 130–140.

[25] A M de Bruin, R Bekker, L van Zanten and G M Koole, *Dimensioning clinical wards using the Erlang loss model*, Annals of Operations Research **178** (2010), 23–43.

[26] E Burke and S Petrovic, eds., *European journal of operational research*, Special Issue on Timetabling and Rostering, vol. 153, Elsevier, 2004.

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

[28] T Cayirli and E Veral, *Outpatient scheduling in health care: a review of literature*, Production and Operations Management **12** (2003), 519–549.

[29] B Cheang, H Li, A Lim and B Rodrigues, *Nurse rostering problems - a bibliographic survey*, European Journal of Operational Research **151** (2003), 447 – 460.

[30] E Cheng and J L Rich, *A home health care routing and scheduling problem*, Report CAAM TR98-04, Rice University, 1998.

[31] C Chien, F Tseng and C Chen, *An evolutionary approach to rehabilitation patient scheduling: A case study*, European Journal of Operational Research **189** (2008), 1234–1253.

[32] *http://www.cs.vu.nl/das3*.

[33] J L Davis, W A Massey and W Whitt, *Sensitivity to the service-time distribution in the nonstationary Erlang loss model*, Management Science **41** (1995), 1107–1116.

[34] R W Day, M D Dean, R Garfinkel and S Thompson, *Improving patient flow in a hospital through dynamic allocation of cardiac diagnostic testing time slots*, Decision Support Systems **49** (2010), 463–473.

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

[36] R W Derlet and J R Richards, *Overcrowding in the nationï£¡s emergency departments: Complex causes and disturbing effects*, Annals of Emergency Medicine **35** (2000), 63–68.

[37] R Dorfman, *A formula for the Gini coefficient*, The review of Economics and Statistics **61** (1979), 146–149.

[38] C H Ellenbecker, *A theoretical model of job retention for home health care nurses*, Journal of Advanced Nursing **47** (2004), 303–310.

[39] A T Ernst, H Jiang, M Krishnamoorthy, B Owens and D Sier, *An annotated bibliography of personnel scheduling and rostering*, Annals of Operations Research **127** (2004), 21–44.

[40] P Eveborn, P Flisberg and M Rönnqvist, *Laps careï£¡an operational system for staff planning of home care*, European Journal of Operational Research **171** (2006), 962 – 976.

[41] Z Feldman, A Mandelbaum, W A Massey and W Whitt, *Staffing of time-varying queues to achieve time-stable performance*, Management Science **54** (2008), 324–338.

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

[43] L Flynn and J A Deatrick, *Home care nurses' descriptions of important agency attributes*, Journal of Nursing Scholarship **35** (2003), 385–390.

[44] B E Fries, *Bibliography of operations research in health-care systems*, Operations Research **24** (1976), 801–814.

[45] B E Fries, *Bibliography of operations research in health-care systems: An update*, Operations Research **27** (1979), 408–419.

[46] M C Fu, *Optimization via simulation: A review*, Annals of Operations Research **53** (1994), 199–247.

[47] M C Fu, *Optimization for simulation: Theory vs. practice*, INFORMS Journal on Computing **14** (2002), 192–215.

[48] S Gallivan, *Challenging the role of calibration, validation and sensitivity analysis in relation to models of health care processes*, Health Care Management Science **11** (2008), 208–213.

[49] S Gallivan and M Utley, *Modelling admissions booking of elective inpatients into a treatment centre*, IMA Journal of Management Mathematics **16** (2005), 305–315.

[50] N Gans, G M Koole and A Mandelbaum, *Telephone call centers: tutorial, review, and research prospects*, Manufacturing and Service Operations Management **5** (2003), 79–141.

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

[52] P W Glynn and W Whitt, *A new view of the heavy-traffic limit theorem for infinite-server queues*, Advances in Applied Probability **23** (1991), 188–209.

[53] L Green, *Queueing analysis in healthcare*, Patient Flow: Reducing Delay in Healthcare Delivery (R W Hall, ed.), International Series in Operations Research & Management Science, vol. 91, Springer US, 2006, 281–307.

[54] L V Green and V Nguyen, *Strategies for cutting hospital beds: the impact on patient service*, Health Services Research **36** (2001), 421–442.

[55] F Guerriero and R Guido, *Operational research in the management of the operating theatre: a survey*, Health Care Management Science **14** (2011), 89–114.

[56] D Gupta and B Denton, *Appointment scheduling in health care: challenges and opportunities*, IIE Transactions **40** (2008), 800–819.

[57] A Haensel, S Bhulai and P M Koeleman, *A stochastic programming formulation for the patient scheduling problem*, Submitted.

[58] A Haensel, M Mederer and H Schmidt, *Revenue management in the car rental industry: A stochastic programming approach*, Journal of Revenue and Pricing Management **11** (2011), 99–108.

[59] R Haijema and J van der Wal, *An MDP decomposition approach for traffic control at isolated signalized intersections*, Probability in the Engineering and Informational Sciences **22** (2008), 587–602.

[60] S Halfin and W Whitt, *Heavy-traffic limits for queues with many exponential servers*, Operations Research **29** (1981), 567–588.

[61] K Heiner, W A Wallace and K Young, *A resource allocation and evaluation model for providing services to the mentally retarded*, Management Science **17** (1981), 769–784.

[62] H Heitsch and W Römisch, *Scenario tree modeling for multistage stochastic programs*, Mathematical Programming **118** (2009), 371–406.

[63] C Ho and H Lau, *Minimizing total cost in scheduling outpatient appointments*, Management Science **38** (1992), 1750–1763.

[64] Y Ho, Q Zhao and Q Jia, *Ordinal optimization: Soft optimization for hard problems*, Springer, 2007.

[65] J M Holtzman and D L Jagerman, *Estimating peakedness from arrival counts*, Proceedings of ITC-9, Torremolinos, Spain.

[66] S H Jacobson, S N Hall and J R Swisher, *Discrete-event simulation of health care systems*, Patient Flow: Reducing Delay in Healthcare Delivery (R W Hall, ed.), International Series in Operations Research & Management Science, vol. 91, Springer US, 2006, 211–252.

[67] A A Jagers and E A van Doorn, *On the continued Erlang loss function*, Operations Research Letters **5** (1986), 43–46.

[68] D S Johnson, C H Papadimitriou and M Yannakakis, *How easy is local search?*, Journal of computer and system sciences **37** (1988), 79–100.

[69] G C Kaandorp and G Koole, *Optimal outpatient appointment scheduling*, Health Care Management Science **10** (2007), 217–229.

[70] P M Koeleman and S Bhulai, *Optimal admission control in rehabilitation facilities*, Submitted.

[71] P M Koeleman, S Bhulai and M van Meersbergen, *Optimal patient and personnel scheduling policies for care-at-home service facilities*, European Journal of Operational Research **219** (2011), 557–563.

[72] P M Koeleman and G M Koole, *Optimal outpatient appointment scheduling with emergency arrivals and general service times*, IIE Transactions on Healthcare Systems Engineering **2** (2012), 14–30.

[73] P M Koeleman and G M Koole, *Simulation optimization for appointment scheduling*, Winter Simulation Conference 2012.

[74] G Koole and A Mandelbaum, *Queueing models of call centers: an introduction*, Annals of Operations Research **113** (2002), 41–59.

[75] G Koole and E van der Sluis, *Optimal shift scheduling with a global service level constraint*, IIE Transactions **35** (2003), 1049–1055.

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

[77] E Litvak, P I Buerhaus, F Davidoff and M C Long, *Managing unnecessary variability in patient demand to reduce nursing stress and improve patient safety*, Joint Commission Journal on Quality and Patient Safety **31** (2005), 330–338.

[78] T E Locker and S M Mason, *Analysis of the distribution of time that patients spend in emergency departments*, British Medical Journal **330** (2005), 1188–1189.

[79] W A Massey and W Whitt, *Networks of infinite-server queues with nonstationary Poisson input*, Queueing Systems **13** (1993), 183–250.

[80] W A Massey and W Whitt, *Stationary-process approximations for the nonstationary Erlang loss model*, Operations Research **44** (1996), 976–983.

[81] J H May, W E Spangler, D P Strum and L G Vargas, *The surgical scheduling problem: Current research and future opportunities*, Production and Operations Management **20** (2011), 392–405.

[82] M L McManus, M C Long, A Copper and E Litvak, *Queueing theory accurately models the need for critical care resources*, Anesthesiology **100** (2004), 1271–1276.

[83] M L McManus, M C Long, A Copper, J Mandell, D M Berwick, M Pagano and E Litvak, *Variability in surgical caseload and access to intensive care services*, Anesthesiology **98** (2003), 1491–1496.

[84] B Miller, *A queueing reward system with several customer classes*, Management Science **16** (1969), 234–245.

[85] B F Nielsen, G M Koole and T B Nielsen, *First in line waiting times as a tool for analysing queueing systems*, Operations Research **60** (2012), 1258–1266.

[86] S N Ogulata, M Koyuncu and E Karakas, *Personnel and patient scheduling in the high demanded hospital services: A case study in the physiotherapy service*, Journal of Medical Systems **32** (2008), 221–228.

[87] R M O'Keefe, *Investigating outpatient departments: implementable policies and qualitative approaches*, The Journal of the Operational Research Society **36** (1985), 705–712.

[88] J M van Oostrum, M van Houdenhoven, J L Hurink, E W Hans, G Wullink and G Kazemier, *A master surgical scheduling approach for cyclic scheduling in operating room departments*, OR spectrum **30** (2008), 355–374.

[89] J Patrick, M Puterman and M Queyranne, *Dynamic multi-priority patient scheduling for a diagnostic resource*, Operations Research **56** (2008), 1507–1525.

[90] S Petrovic and G Vanden Berghe, eds., *Annals of operations research*, Special Issue on Personnel Scheduling and Planning, vol. 155, Springer Netherlands, November 2007.

[91] J Pichitlamken and Nelson B L, *A combined procedure for optimization via simulation*, ACM Transactions on Modeling and Computer Simulation **13** (2003), 155–179.

[92] W B Powell, *Approximate dynamic programming: Solving the curses of dimensionality*, John Wiley & Sons, Inc., 2007.

[93] M L Puterman, *Markov decision processes: Discrete stochastic dynamic programming*, John Wiley & Sons, 1994.

[94] E J Rising, R Baron and B Averill, *A systems analysis of a universty-health-service outpatient clinic*, Operations Research **21** (1973), 1030–1047.

[95] K W Ross and D H K Tsang, *The stochastic knapsack problem*, IEEE Transactions on Communications **37** (1989), 740–747.

[96] A Ruszczyński and A Shapiro, *Stochastic programming, handbooks in operations research and management science vol. 10*, Elsevier, Amsterdam, 2003.

[97] R Schultz, M P Nowak, R Nürnberg, Römisch W and M Westphalen, *Stochastic programming for power production and trading under uncertainty*, Key Technology for the Future (W. Jï£¡ger and H.-J. Krebs eds.), Springer, 623–636.

[98] L Shi and S Ólafsson, *Nested partitions method for stochastic optimization*, Methodology and Computing in Applied Probability **2** (2000), 271–291.

[99] L Shi and S Ólafsson, *Nested partitions method, theory and applications*, Springer, New York, 2009.

[100] W E Stein and M J Côté, *Scheduling arrivals to a queue*, Computers and Operations Research **6** (1994), 607–614.

[101] P T Vanberkel, R J Boucherie, E W Hans and J Hurink, *A survey of health care models that encompass multiple departments*, International Journal of Health Management and Information **1** (2010), 37–89.

[102] P T Vanberkel, R J Boucherie, E W Hans, J L Hurink, W A M van Lent and W H van Harten, *An exact approach for relating recovering surgical patient workload to the master surgical schedule*, Journal of the Operational Research Society **62** (2010), 1851–1860.

[103] P M Vanden Bosch, D C Dietz and J R Simeoni, *Scheduling customer arrivals to a stochastic service system*, Naval Research Logistics **46** (1999), 549–559.

[104] J Vissers and J Wijngaard, *The outpatient appointment system: design of a simulation study*, European Journal of Operational Research **3** (1979), 459–463.

[105] P P Wang, *Optimally scheduling N customer arrival times for a single-server system*, Computers and Operations Research **8** (1997), 703–716.

[106] J D Welch and N T J Bailey, *Appointment systems in hospital outpatient departments*, The Lancet **259** (1952), 1105–1108.

[107] W Whitt, *On the heavy-traffic limit theorem for GI/G/∞ queues*, Advances in Applied Probability **14** (1982), 171–190.

[108] W Whitt, *Heavy-traffic approximations for service systems with blocking*, AT&T Bell Laboratories Technical Journal **63** (1984), 689–708.

[109] W Whitt, *Understanding the efficiency of multi-server service systems*, Management Science **38** (1992), 708–723.

Samenvatting (Dutch Summary)

Een zorgvuldige oplossing: het plannen van patiënten

Zoals de titel al enigszins verraadt, gaat dit proefschrift over het plannen van patiënten in verschillende situaties binnen de gezondheidszorg. Het plannen van patiënten is een heel breed onderwerp, en er zijn heel veel situaties te verzinnen waarin er gepland moet worden. Een gemeenschappelijk kenmerk in alle gevallen is dat er een balans gevonden moet worden tussen de wachttijden en andere servicecriteria voor de patiënten aan de ene kant, en de hoeveelheid en benutting van de capaciteit aan de andere kant. Als de duur van alle opnames, afspraken en dergelijk nu van tevoren perfect voorspelbaar waren, was het plannen een heel stuk eenvoudiger. Helaas is dit vrijwel nooit het geval, en daarom moet in elke situatie met deze variaties rekening gehouden worden.

Dit proefschrift is opgedeeld in zes hoofdstukken. In het eerste hoofdstuk wordt ingegaan op de rol van operations research in het algemeen, en plannen of *scheduling* binnen de gezondheidszorg in het bijzonder. Ook behandelen we verschillende technieken die bij dergelijke problemen bruikbaar zijn. Dan volgen vier hoofdstukken die elk een apart planningsprobleem behandelen. In het laatste hoofdstuk worden de conclusies op een rijtje gezet. De vier inhoudelijke hoofdstukken zijn als volgt ingedeeld:

In hoofdstuk 2 wordt het probleem behandeld hoe het beste afspraken gepland kunnen worden binnen een dagdeel. Dit probleem doet zich bijvoorbeeld voor op een polikliniek. Omdat het niet zeker is hoe lang een afspraak exact gaat duren, moeten de wachttijden van de patiënten afgewogen worden tegen de uitloop en wachttijd van de arts. In het eerste deel wordt dit probleem aangepakt met een local search-aanpak, voor het geval dat er spoedgevallen tussendoor kunnen komen, die vrijwel direct behandeld moeten worden. Hierbij nemen we aan dat sommige patiënten niet komen op hun afspraak, maar dat alle patiënten die komen exact op tijd zijn. In het tweede deel maken we de situatie iets realistischer door deze laatste aanname te laten vallen. Hiervoor werkt de local search-techniek niet meer, en daarom gebruiken we optimalisatie via simulatie om het probleem te bekijken. Hiermee hebben we geen garantie meer dat de optimale oplossing gevonden wordt, maar experimenten tonen aan dat deze aanpak heel goed werkt.

Het zo goed mogelijk plannen van opnames op een ziekenhuisafdeling is het onderwerp van hoofdstuk 3. Op elke ziekenhuisafdeling is sprake van variatie in de vraag naar bedden. Dit is variatie gedurende de dag en de week, en over de verschillende weken. Een groot deel van deze variatie wordt veroorzaakt door spoedgevallen en een onvoorspelbare ligduur van de patiënten, maar het grootste deel is vaak veroorzaakt door een onevenwichtige planning. In dit hoofdstuk beschrijven we een manier om de variatie te meten en voorspellen, en vervolgens gebruiken we dit om een optimale planning voor de opnames te bepalen. Het doel bij het maken van deze planning is de vraag naar bedden zo dicht mogelijk te laten aansluiten bij een doelwaarde voor elke dag van de week. Hierdoor wordt de variatie kleiner, en hoeven er dus minder patiënten geweigerd of op een andere afdeling geplaatst te worden.

Hoofdstuk 4 gaat over prioriteren van patiënten die wachten op thuis- of revalidatiezorg. Deze patiënten hebben meestal elke week zorg nodig, waarbij per patiënt de hoeveelheid en de soorten zorg die deze nodig heeft kan verschillen. We delen de beschikbare tijd per week en per soort medewerker op in uren, en definiëren patiënttypes die elke week hetzelfde aantal uren zorg nodig hebben van de verschillende soorten medewerkers, gedurende de hele periode dat zij zorg krijgen. Dan gebruiken we Markov beslissingsprocessen om een optimale prioriteringsstrategie te bepalen, waarbij de wachttijd van de patiënten gewogen naar type geminimaliseerd wordt.

In hoofdstuk 5 ten slotte bekijken we een situatie waarin niet de gemiddelde wachttijd van belang is, maar het percentage patiënten dat te lang wacht. Het is echter niet mogelijk om de actuele wachttijden van alle wachtende patiënten bij te houden om daarop te gaan sturen; hiervoor is namelijk meer geheugen nodig dan waar de meeste computers over beschikken, en bovendien wordt het bepalen van een goede strategie binnen een redelijke tijd onmogelijk. We gebruiken daarom een benadering van de wachttijd van de patiënt die het langste wacht van elk type om het hele systeem mee te benaderen, en vervolgens gebruiken we Markov beslissingstheorie om de wachttijden van de verschillende typen patiënten te balanceren. Elk type patiënten kan hierbij een eigen verdeling voor de behandelduur hebben, en eventueel een verschillende waarde voor de toegestane wachttijd.