A policy approximation of a Markov decision process for scheduling clients in an outpatient mental healthcare clinic.

M.S. van den Berg - Vreeken BSc

Supervised by:
prof. dr. R.J. Boucherie - University of Twente
J.Jolink MBA - Mediant

September 24, 2021

Abstract

The main issues within the department of mood and anxiety issues of Mediant are long waiting times for clients and a high experienced workload for practitioners. Since scheduling clients concerns both and clients are currently scheduled ad hoc, a method to schedule clients is developed. The goal is to reduce both waiting times and experienced workload.

Since it is known how many appointments a client needs before a new client needs to be notified for an initial appointment, it is possible to delay the scheduling of a new client until a current client leaves. To make scheduling easier and reduce irregularity, a blueprint schedule is made.

To decide which type of client to plan with which type of practitioner, a Markov Decision Process (MDP) is formulated. Due to the curses of dimensionality, this is too large to solve for real-life cases. Therefore an approximation is made, which uses simple scheduling rules: a combination of a trunk reservation policy and a threshold policy. This approximation gives an expected penalty of 27% higher and an expected queue length of 29% higher compared to the MDP for small cases, while the runtime is much shorter. The approximation outperforms the current policy of assigning clients to the default type of practitioner if planned back-to-back in a realistic sized simulation.

Preface

My graduation project was at Mediant, which is an outpatient mental healthcare clinic. The project was from January 2021 until September 2021. Despite the COVID-19 epidemic, I could still work at the office, among a team of mental healthcare practitioners. I was positively surprised on how welcome I was in the team and how kind everybody was to me.
I specifically searched for a practical project in healthcare, since it motivates me when I can help people directly with my research.

The goal of the assignment was to reduce the waiting lists, but also to make the workplace less stressful for practitioners. Therefore, I first examined how the company works and why the practitioners experience stress. After that, I concentrated on improving the scheduling of clients, with as goal to both minimize the expected waiting time and reduce varying schedules for practitioners and hopefully reduce stress this way. I am very happy that the company has decided to continue with this project and test my models in practice. I hope that my work can make a difference for both goals.
I would like to thank my internship supervisors at the company, Jürgen Jolink, for his time and support, and Ingrid Höelsgens, for her explanations about the used methods within the department and for her patience to answer all my questions.
Furthermore I would like to thank my supervisor from the University of Twente, Richard Boucherie, for all the time he spent helping to come up with ideas, for encouraging me when I needed it and for providing feedback on my report.

September 24, 2021,
Enschede,
Sylvia van den Berg - Vreeken

Contents

1 Introduction

Within the mental healthcare, the issue of long waiting times has been a problem for several years. Measured 12 months after the start of the treatment, long waiting times have proven to lead to deterioration in the subdomains behaviour, impairment, symptoms and social functioning, especially when the waiting time was longer than 3 months (Reichert and Jacobs, 2018). Clients with a long waiting time are more likely to either refuse treatment or drop out prematurely (Westin et al., 2014). Furthermore, long waiting times are a barrier to enter treatment (Redko et al., 2006) and can lead to a deterioration of health, with an even greater need for help as a result (Zorgautoriteit, 2018.
In the Netherlands, a maximum of four weeks waiting time till triage is set and a maximum of ten weeks between triage and start of treatment. However, these standards are often not met. In this report, a mathematical approach to this problem is used.
This chapter discusses the background of mental healthcare, the current routing used by Mediant and the contributions, objectives and structure of this report.

1.1 Background

This section first gives some information about what is already tried to reduce waiting times in the Netherlands regarding mental health care, and then some background on Mediant and the department S&A within Mediant.

1.1.1 Waiting Times for Mental Health Services

The issue of long waiting times for mental health services has been on the political agenda for some years. To reduce waiting times, several initiatives have been set up, among which nationwide agreements on how to tackle waiting times. NZa is asked to report on this and speak with the involved parties. The main agreements are on effective use of the available capacity, enlargement of that capacity by creating extra education spots, the use of e-health, improving the cooperation between care givers and health insurers and improving availability of ambulatory, acute and secured mental healthcare. (Zorgautoriteit, 2017)
Unfortunately, these measures have not directly led to a reduction of the waiting times. However, as a result of these measures, all mental health services are now obligated to disclose their waiting times, which are publicly published by Vektis. In 2018, not all waiting times were completely disclosed and sometimes incorrect. NZa was still working on this at the time of publication. (Zorgautoriteit, 2018)

1.1.2 Mediant

Mediant is a mental healthcare organisation in Twente, a region in the Netherlands. They provide help, advice and guidance for people having mental health issues. The whole organisation has around 1150 employees. Despite the extra education spots, there is still a chronic shortage of qualified psychologists and

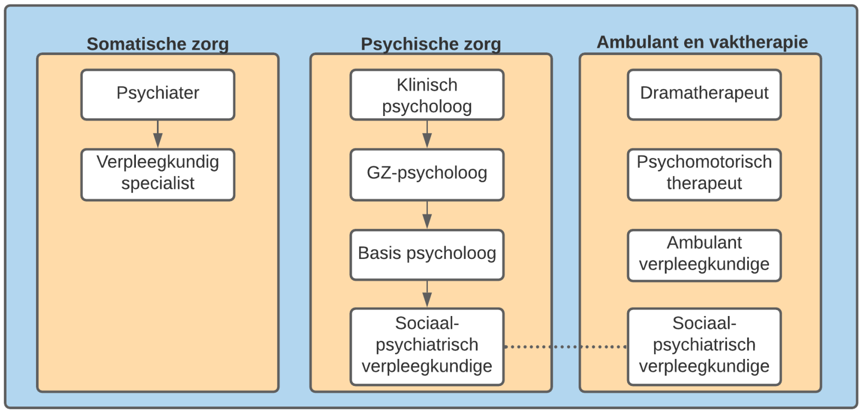

Figure 1: Overview of all practitioners.

psychiatrists, which makes it very difficult to increase the number of staff members.

Within the mental healthcare, there are two subdivisions: basic mental healthcare (BGGZ) for mild to moderate mental health problems and specialist mental healthcare (SGGZ) for severe mental health problems, often being comorbid problems, meaning that there is more than one main problem.

Currently the average waiting time for a triage is 5 weeks. The average waiting time between the intake and the start of the treatment is 4 weeks for BGGZ and 6 weeks for SGGZ. However, there are some departments, in particular the center for developmental disorders, which have a significantly longer waiting time.

1.1.3 Department of Mood and Anxiety

One of the outpatient departments within Mediant deals with mood and anxiety issues, and is called S&A, which is short for "stemming en angst", meaning 'mood and anxiety'. This department has around 35 employees, among which psychiatrists, psychologists with different levels of qualifications and psychiatric nurses. An overview of all types of practitioners is given in Figure 1. The type of practitioners are given in Dutch, since not all titles can be translated to international terms. In this figure, an arrow means that the type of practitioner at the origin of the arrow should be able to do any treatment the type of practitioner at the end of the arrow can do. Furthermore, the practitioners are divided into three sections, which do not have much interaction. "Sociaal psychiatrisch verpleegkundige" is shown in two sections, since they see clients from both sections. A client could have a treatment at one or more of these sections, but there are no restrictions about the order, and they could also be given in the same time period.

This paper focuses on the department S&A, since this is expected to be a more or less standard department within Mediant. A subdepartment of this department is the bipolar disorders department, but since the clientele is quite different and the subdepartment is separate, it is left out of this research. Within S&A, a lot of clients have anxiety, also for going outside or sometimes for going to the facility of Mediant. This results in relatively many clients cancelling their appointments.

1.2 Current Routing

Clients enter the S&A department in one of two possible ways: either by a direct referral to S&A or by being referred to Mediant in general, after which they first have a so-called triage and are then referred to S&A. In the latter case the patient already has a preliminary diagnosis and there is already determined whether BGGZ or SGGZ is needed. There are no walk-in clients allowed, all appointments need to be scheduled in advance.
In case the client was referred directly to S&A, a triage has to be done first. In most cases, the client is already in the right department. If the client is indeed in the right department, triage is combined with an intake. If not, the client is referred to the right department or back to the general practitioner. In case the client is referred to S&A after a triage, an intake is done first. Triages and intakes are done FIFO, with the exception that if a client does not pick up the phone for making a short term appointment, for example when someone cancelled, then the next one on the list is called.
After intake, the client is discussed in the multidisciplinary consultation, abbreviated as MDO, "multidisciplinair overleg" in Dutch. In the MDO a diagnosis is set and a treatment plan is determined, which is afterwards discussed with the client. Hereafter the client is put on the waiting list for treatment. Treatments are also started FIFO. Furthermore, the type of practitioner is determined during the MDO. A schematic overview is given in Figure 2, where the black arrows indicate that the client is the responsibility of someone within S&A, and the grey arrows indicate that the client is someone else's responsibility. If the client is your responsibility, you have to see him if he needs immediate help, even if no treatment has started yet.
Within BGGZ there are three care paths, which are standardized paths, and within SGGZ there are seven care paths. Within each care path, certain treatments are recommended, but the final decision of which treatment is recommended to the client lies with the responsible practitioner. Within BGGZ, the maximum number of appointments is 12 per year, while within SGGZ there is no maximum. This maximum is set by the health insurance companies. Furthermore, when a care path is finished, the option remains to start a new path.
On several places in this system there are queues. Waiting times exist before the triage, intake and treatment, so most clients have to wait either two or three times. There could also exist waiting times within the treatment, especially between an initial and follow-up treatment or with changes of treatment. Note that after the first contact, so after triage but also after intake, the correspond-

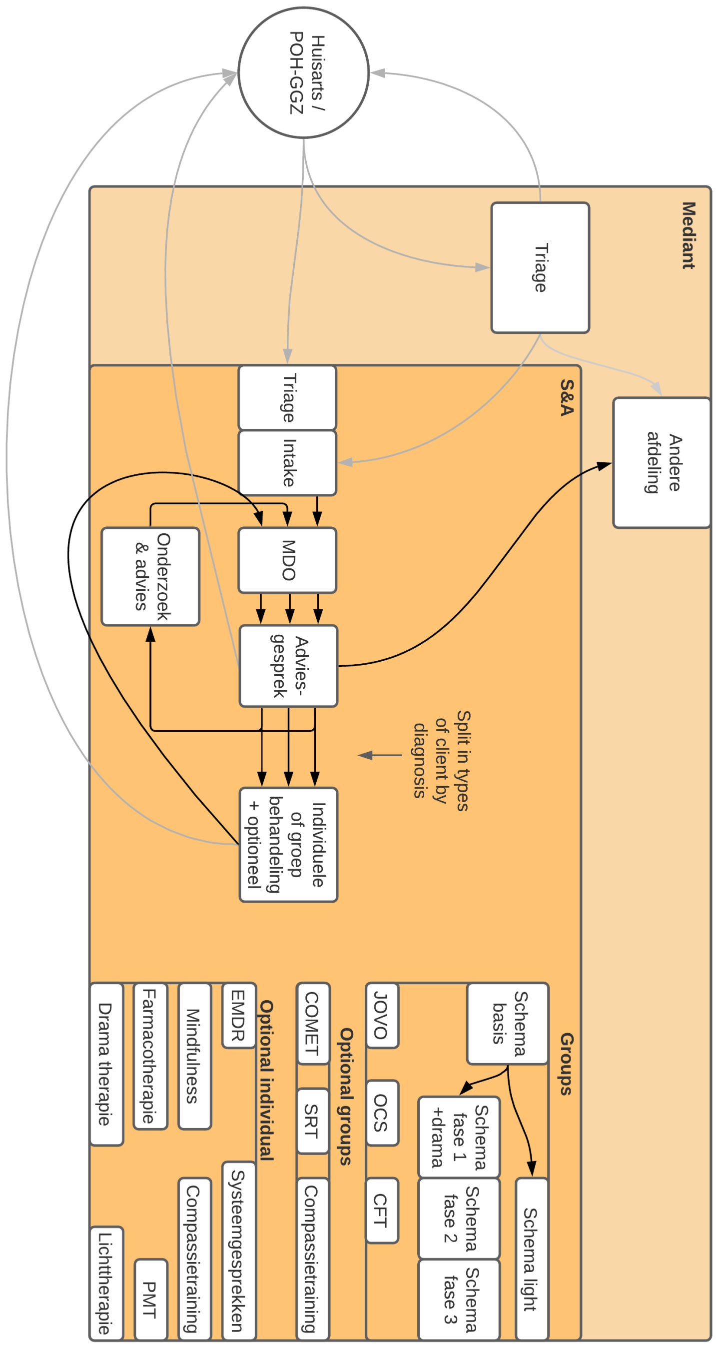

Figure 2: Schematic overview current methods.

ing practitioner is responsible for the client. This means that, even if there is a waiting list for the preferred treatment, the practitioner still has to see the client in the meantime.
There are several options for treatments, mainly individual treatments, but also group treatments are given. There are two types of group treatment: group treatment that is the main treatment and group treatment which is complementary to other treatment. In Figure 2, these are respectively called 'Groups' and 'Optional groups'. There are also individual treatments which needs to be complimentary to the main treatment, these are given in the block 'Optional individual'.
At least once in the diagnostic fase, a senior practitioner should be present, meaning either with intake or with the discussion of the treatment plan with the client. Furthermore, within SGGZ, a senior practitioner has to be present at least once a year, and can also be consulted by the practitioner if not sure about the approach.

1.3 Contributions

In this report, the current situation at the department S&A is analyzed, a scheduling method is introduced, an Markov Decision Process (MDP) is designed to assign types of clients to types of practitioners and an algorithm for an approximation of this MDP is given.

1.4 Objectives

There are three involved parties in this project, namely clients, practitioners and health insurers. From the clients as well as the health insurers perspective, reducing the total average waiting time is the goal of this project.
From the practitioners perspective, a reduced or less experienced workload is beneficial, next to reduced waiting times. Reducing the experienced workload means for example that emergency treatments should be more easy to plan, but also that there should be time to consult colleagues and reflect upon treatment.
The main goal is to reduce the average waiting time, with the restrictions that the maximum expected waiting time cannot be (too) high for all types of clients and the experienced workload should not increase. Decreasing the experienced workload is a second goal. Both goals apply specifically to the department S&A within Mediant.
Chapter 2 will, after analyzing the current situation, examine different options to obtain these goals.

1.5 Structure of the Report

Chapter 2 of this report gives an analysis of the problem, discusses possible research directions and gives the research questions. Chapter 3 discusses some basic models used in this report and Chapter 4 discusses relevant literature. A model for this problem is discussed in Chapter 5 and an approximation of this model is given. Chapter 6 compares the results from the model with the approximation, as well as the runtimes. Finally, Chapter 7 gives a discussion of the results and Chapter 8 gives the conclusions and recommendations.

2 Problem Analysis

Long waiting times are caused by several factors. For the M|G|1 queue, the waiting time is given by the Pollaczek-Kinchine formula:

$$\mathbb{E}[W_q] = \frac{1}{2} \cdot (1 + C_S^2) \frac{\rho}{1-\rho} \mathbb{E}[S], \tag{1}$$

where $\rho = \frac{\lambda}{\mu}$, λ is the expected arrival rate, μ is the expected service rate and C_s^2 is the squared coefficient of variation of the service time S. This queueing system assumes one practitioner, Poisson arrivals and an unknown service time distribution.

Before looking closer at different options to decrease waiting time, the current capacity versus workload and the sensitivity of the waiting times are examined. Finally, the research questions are stated.

2.1 Current Situation

We know from historical data how much work is arriving. This is compared with the production standard as given by the company.

Since we know on average how much work is arriving, and we know the realized queue for care paths and with the use of Little's law the waiting time between intake and treatment, we can use the Pollaczek–Khinchine formula as given in Equation (1) to find the expected service rate. Using this, we can find the amount of hours that have been available for care paths, under certain assumptions.

A sensitivity analysis is done to examine the influence of the arrival rate and capacity on the expected waiting time.

2.1.1 Data Analysis: Length of Care Paths

We use data obtained from the system used by practitioners, USER. Care paths with starting dates from 1-1-2016 till 31-12-2019 are evaluated to find the distribution and the expected length of a care path. It has been noticed that there are significant differences between BGGZ and SGGZ care paths, but the main care paths within these sections are very similar. Therefore, the data is divided into three groups: BGGZ, SGGZ and residual, where the last group contains all care paths which are not common within S&A, but nonetheless sometimes given. Of the data analyzed, 54% falls within SGGZ, 27% in BGGZ and 19% in residual. Most of the care paths in the last group are no longer used, however, these clients are still expected to arrive at S&A, only to be sorted into other care paths nowadays.

Not all care paths in the analyzed data have a (registered) end date. This might be because somehow the paths were never closed, although the client has left, but is mostly because the client has not yet left and is still in treatment. This means that we have right censored data. By using the Kaplan-Meier estimator, we find the survival function as depicted in Figure 3 with the corresponding

Type	Hours	Variance (hours^2)
BGGZ	9.5	57
SGGZ	18.7	623
Residual	13.6	140
Average	15.45	334
Modules	4.45	70

Table 1: Average treatment duration and variance per type based on Kaplan-Meier estimation.

confidence intervals. In Table 1, we find the average treatment duration based on the survival functions per type, and the variance. The latter is found by analyzing the cumulative distribution as found with the Kaplan-Meier estimator. Modules represent clients from other departments of Mediant, which receive a relatively small treatment at S&A. The average is based on the Kaplan-Meier estimator on all care paths, which is not depicted in Figure 3.

2.1.2 Capacity vs Workload

In the years 2016-2019, on average 4.2 new care paths and 1.97 modules were started each month per full-time practitioner. Of the intakes and triages, no detailed information is available, only the total amount of time spent on triage and intakes per year is known. The average amount of hours needed per month per fte is given in Table 2. Note that the number of full-time practitioners as used here is gathered from a general database, and therefore exceptions in production standard, for example for education, are not taken into account. This means that the actual number of full-time practitioners available for production is lower, and thus the arrival rate per fte is in reality somewhat higher.
The production standard of Mediant indicates every full-time practitioner should spend 1330 hours per year on client bound care, which is 110.83 hours per month. Of these hours, the goal is to spend at least 68% on direct care, which is 75.37 hours per month, leaving a goal of 35.47 hours for administration.
Together with the average amount of incoming work per full-time practitioner per month, as mentioned in Table 2, this means that to keep the system stable and the queue from overflowing, the practitioners need to spend more hours on direct care than the amount set in the production standard and any fluctuations should be absorbed using extra hours.

2.1.3 Available hours

We need to find $\mu = x \cdot \lambda$. We have an arrival rate of 4.2 new care paths per month, meaning $\lambda = 4.2$. The amount of hours that have been available per month are $x \cdot 4.2 \cdot 15.45 = x \cdot 64.89$.
From Table 1 we know that the variance of an average care path is 334 hours^2. To find the variance in month^2, we have to divide this by the hours available in a

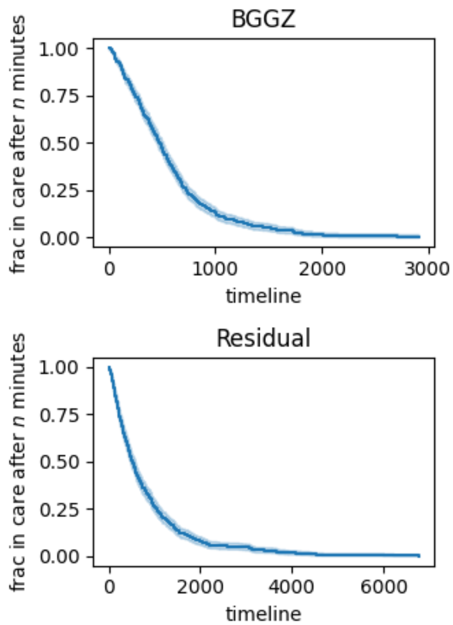

Figure 3: Kaplan-Meier estimation per type.

Type	Hours face-to-face	Hours administration
Care paths	64.89	20.96
Modules	8.77	5.25
Triages/intakes	7.88	6.69
Total	81.54	32.90

Table 2: Average workload per full-time practitioner per month

month, squared. Thus the variance is $\frac{334}{(x\cdot 64.89)^2}$. This gives a squared coefficient of variation $C_s^2 = \frac{\frac{334}{(x\cdot 64.89)^2}}{\mathbb{E}[S]^2}$.
The average queue length over the years 2016-2019 was 1.68 clients per full time practitioner. Using Little's law, we find the waiting time,

$$\mathbb{E}[W_q] = \frac{1}{\lambda}\mathbb{E}[L_q] = \frac{1}{4.2} \cdot 1.68 = 0.4 \text{ months.} \tag{2}$$

Using this knowledge and Equation 1, we find $x = 1.48$ is the only relevant solution, since we know that x is real and $x > 1$.
This means that each month, $1.48 \cdot 64.89 = 96.04$ hours need to be available for care paths.

2.1.4 Sensitivity

If we assume that we have the same hours available, but now each care path takes half an hour less, we can check what will happen to the queue. We then still have 4.2 arrivals, only each arrival takes 14.95 hours instead of 15.45. This means that we have less incoming work, namely $4.2 \cdot 14.95 = 62.79$ hours, instead of $4.2 \cdot 15.45 = 64.89$. If we incorporate this in the arrival rate, we have $\lambda = 4.2 \cdot \frac{62.79}{64.89} = 4.06$. We still have $\mu = 4.2 \cdot 1.48 = 6.22$, meaning $\rho = \frac{4.06}{6.22}$. We assume that the variance stays the same. We then see

$$\mathbb{E}[W_q] = \frac{1}{2} \cdot (1 + C_s^2) \frac{\rho}{1-\rho} \mathbb{E}[S] \tag{3}$$

$$\mathbb{E}[W_q] = 0.36 \text{ months.} \tag{4}$$

With the use of Little's law, we find the expected number of clients in the queue: $\mathbb{E}[L_q] = \lambda \cdot \mathbb{E}[W_q] = 4.06 \cdot 0.36 = 1.46$ clients per fte. An overview of the waiting time for treatment depending on the average treatment duration is given in Figure 4.

Similarly, we could find the expected waiting time depending on the number of available full-time practitioners. This is shown in Figure 5.

Each month, there are on average 3.3 intakes and triages per fte, other arrivals are re-registrations. If we assume that the fraction of re-registrations stays the same, we can also find the waiting time for a varying number of intakes and triages. This is given in Figure 6.

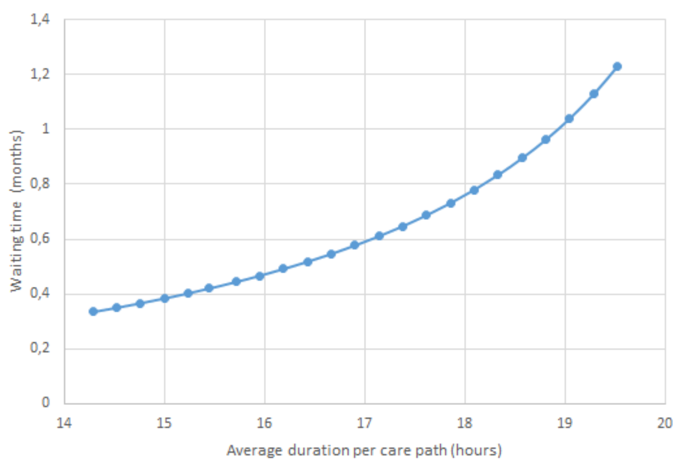

Figure 4: Waiting time as function of the average treatment duration.

2.2 Possible Research Directions

As can be seen in Equation (1), there are several options to decrease the expected waiting time: decreasing the expected arrival rate, increasing the expected ser-

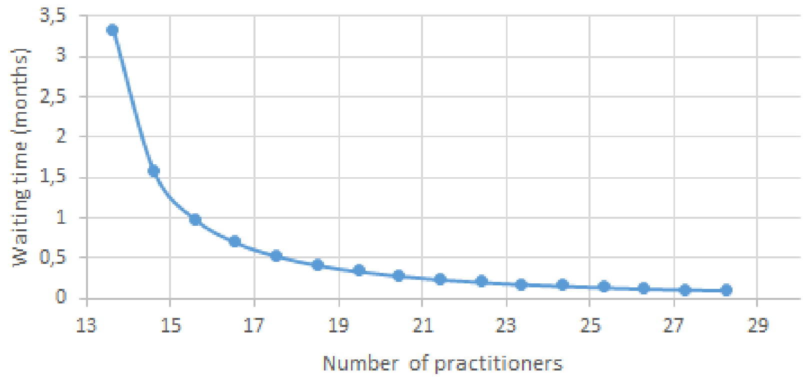

Figure 5: Waiting time as function of the number of practitioners.

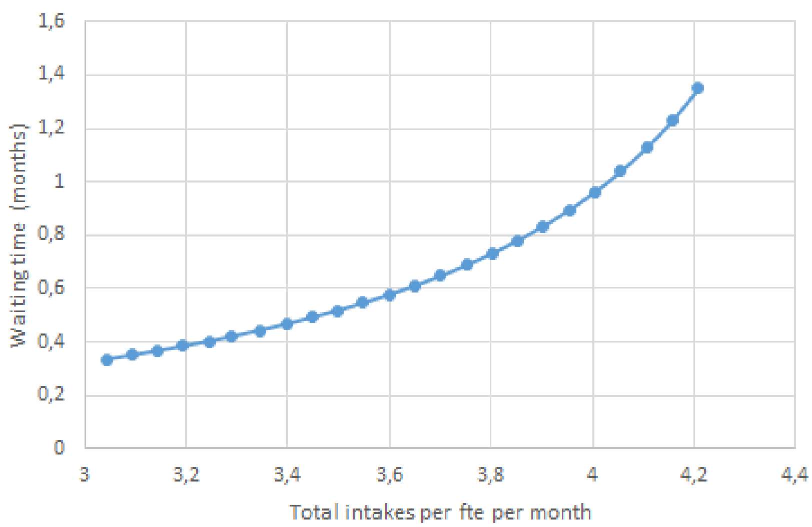

Figure 6: Waiting time as function of the number of intakes and triages.

vice rate or decreasing C_S^2. Decreasing the expected arrival rate could be either done by accepting less arrivals or by decreasing the expected amount of work, for example by examining the effectiveness and duration of treatment. This last option also involves the squared coefficient of variation of the service time. Increasing the expected service rate could be done by hiring more employees, which as stated before, is very difficult. Furthermore, the time available could be used more efficiently by looking carefully at scheduling and cancellations.

2.2.1 Arrivals

Clients arrive by referral from elsewhere. A triage is done to evaluate whether a client really needs psychiatric help. Since clients who do not need psychiatric help are already being referred elsewhere, only clients who really need psychiatric help remain. This means that there is little we can do about the arrival process. Only when really needed, this road should be further examined, by for example referring clients to other psychiatric instances or, when the need is great, by refusing clients with the mildest problems.
The arrivals to treatment, however, can be and are regulated by the number of triages and intakes planned. This can be seen from the fact that the queue length before triages and intakes increased during the COVID-19 pandemic, but the queue for treatment hardly changed.

2.2.2 Effectiveness and Duration of Treatment

Within the system, clients go all different kind of routes. If a more efficient route could be taken, this would mean that the client would spend less time in the system. When a client is waiting on triage, no treatment is given. However, when a client is waiting on an intake or a specific treatment, bridging treatment is often offered, while this bridging treatment does not necessarily provide added benefit. Another view is that clients waiting on triage might benefit from bridging treatment or earlier triage, which might even cause the actual treatment to take less time. The use of bridging treatment could therefore be evaluated.
It is remarkable that for groups the waiting time is on average longer than for individual treatment. This could avert clients from groups to other treatments with less waiting time. The question could be posed whether a group or individual treatment is more effective and efficient, even though in groups more clients are treated at once. Groups are quite difficult to plan, and if planned, not all clients are available when the group treatment takes place. While normally a group has around 6-8 clients, during the COVID-19 pandemic a maximum of 4-6 clients per group is maintained, depending on the group and the size of the room available. Groups do now have a fixed starting and endpoint, but the department is working towards more 'open' groups, meaning that clients can enter or leave at any time.
Individual treatment on the other hand is already flexible. However, the endpoint is not always clear. Practitioners are prone to treating clients longer than strictly needed, because it still has some added benefit. The consequences of these extra treatments on the waiting list are not directly clear for practitioners, it is not realised that all those extra treatments together might have a huge impact. Making this visible with the help of a model might help to motivate practitioners to stop treatment earlier. Furthermore, reminders for evaluations concerning clients who are already some time in the system will probably help stop treatment in time.

2.2.3 Scheduling Clients

The assigning of clients to practitioners influences the throughput, since one practitioner might do one type of treatment more effective than another practitioner. Furthermore, there is some overlap between competences of practitioners, but not all practitioners can do all types of treatments. For example, anyone can do a triage or intake, but only if a senior practitioner is present for part of the triage/intake or the treatment proposal. It might help to carefully consider which practitioner should be assigned to triage, intake and to which client, and how many triage/intakes to do compared to the number of treatment sessions.
Practitioners have the feeling that their agenda is full, there cannot be handled any more clients. However, their agendas are now planned ad hoc and might even change during the day. When effectively planned, in such a way that it is still manageable for practitioners, it is possible that even more clients can be treated, or at the least practitioners have more of the feeling that the schedule is doable.
Currently, whenever a new client arrives from the MDO with an appointment plan, treatments immediately are scheduled with a practitioner, wherever there is place in the practitioners schedule, and the client is notified. To make sure that there are enough sessions scheduled, more appointments are scheduled than are expected to be needed. It could be examined how the number of planned treatments influences the realised number of treatment sessions and what the effect of needing more or less treatments is on the agendas of the practitioners. Planning more treatments than needed might lead to using more treatments, since practitioners rather overtreat than undertreat. However, planning less treatments than needed creates another planning problem: Extra treatments are hard to plan, since the practitioners agenda is fully booked.

2.2.4 Cancellations

A no-show means that a client does not show up for the appointment, without notice. At S&A, no-shows occur on around 4 percent of the appointments, but are not really seen as a problem. This is mainly because all clients are adults, so no-shows can be discussed between the practitioner and the client and, if necessary, treatments can be stopped. Furthermore, since there are not a lot of no-shows and the agenda of the practitioners is quite full, time gained with no-shows can almost always be used in a useful way, for example for administration.
More often, clients cancel their appointments in advance. This gives time to schedule another appointment. However, this is mostly only possible for a cancelled triage or intake, since then a new triage or intake could be done, but for a cancelled treatment often no other treatment can be planned on such short notice, and the slot is too small for a triage or intake. The appointment status for all appointments at S&A were requested for the years 2016 till 2019. An overview is given in Figure 7. The time created by cancellations by Mediant is

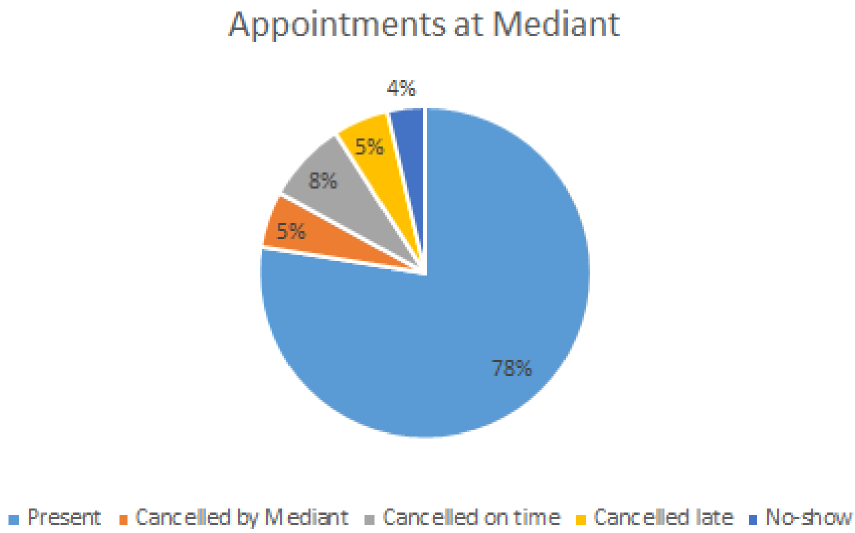

Figure 7: Appointment status.

either unavailable for other appointments, or an appointment series was discontinued. In the latter case, these slots are filled with administration, intakes or bridging therapy. If possible, the slots are used for treatments for new clients, but often there are not enough empty slots to fit all appointments of a new client, and since interruptions of treatments are highly unwanted, this does not happen a lot. Although there is no time directly wasted by cancellations by Mediant, rescheduling the cancelled appointments might mess up the rest of the schedule. In about half of the slots of timely cancelled appointments, new appointments are scheduled. The other half is mainly left empty or used for administration. Late cancellations and no-shows are almost never filled with new appointments, only administration.
The influence of alterations on the agendas should be researched. Scheduling the appointments in a clever way might reduce the cancellations by Mediant.
When clients are late for a treatment session, most practitioners still end the appointment at the agreed time, and thus are not late for the next appointment. However, for some appointments, as for example intakes, all scheduled time is needed, and the appointment takes longer than was agreed on. This can result in playing catch-up all day and feeling stressed about time. It can also lead to postpone administration, resulting in the same issue the next day.

2.3 Research Questions

As seen this chapter, there are different ways of preventing long waiting times. The main directions of interest are:

- restricting arrivals,
- the effectiveness and duration of treatment:
  - group treatment versus individual treatment,
  - bridging treatment,
- the effect of scheduling a number of appointments,
- scheduling clients and
- handling of cancellations.

In this project is chosen to focus on the scheduling of clients, mainly because it is expected that the most can be gained there. Furthermore, practitioners do know best what and how many treatments are best for the client, the only changes in approach that would be accepted are diverting more or less resource on, for example, triage and intakes and planning more or less appointments ahead, as long as there is room to add appointments if deemed needed. These two changes in approach could also be considered when considering scheduling. The main research question is:

> ”How can the stress on the mental healthcare system be reduced by better scheduling?”

Subquestions needed to answer this question are:

1. What kind of model could be used to schedule clients?
2. How can an effective and workable schedule be made?
3. How should the resources be divided between different groups of clients?

3 Basic Models

There are two main approaches to modelling a scheduling problem, namely modelling with and without uncertainty. Modelling without uncertainty is generally faster and easier and can be done by considering probabilities as fractions. However, arrivals are stochastic, it is uncertain what type of client arrives and how many treatments are needed. Since this is not modelled in a model without uncertainty, a static scheduling solution is less accurate than a dynamic scheduling solution.

We first discuss a unified framework for stochastic optimisation, then we discuss Markov Decision Processes (MDP), which are used in the model described in this report, and some general approximation methods for the MDP.

3.1 Unified Framework

Powell (Powell, 2019) has developed a unified framework for stochastic optimisation, where all sequential decision problems are modelled by five core components: State variables, decision variables, exogenous information, transition function and the objective function. We use a combination of the variables as defined by Powell and the variables as commonly used in Markov Decision Processes. All of these components are defined below.

The state variable s_t contains all information and only information needed to compute the cost/contribution function and the constraints from time t onward (Powell, 2019), S being the state space.

The action space A_s contains all possible actions a_t when in state s_t. A policy $\pi = (d_1, d_2, \dots)$, in the set of all possible policies Π gives for every state s_t a decision d_t.

The exogenous information describes how all the exogenous information, needed to go from one state to the next, is modelled. The exogenous information becomes available after there is decided on an action. The current state, the action decided on and exogenous information together determine the next state. The function that describes this relation is called the transition function. Finally, the objective function should be determined.(Powell, 2019)

3.2 Markov Decision Process

In a Markov Decision Process (MDP), a decision is only based on the current state s, and therefore we drop the time index. We then define the one-step transition matrix where $\mathbb{P}(s'|s, a)$ is the probability that we move from state s to state s' when we take action a.

Reward $r(s, a)$ for being in state s and taking action a gives a method to evaluate action a. There are two different ways to formulate an objective function based on the reward function. The first is to maximize the expected total discounted reward

$$v_\lambda^\pi(s) = \lim_{N\to\infty} \mathbb{E}_s^\pi \sum_{t=1}^{N} \lambda^{t-1} \cdot r(s_t, d_t(s_t)), \tag{5}$$

where $0 \leq \lambda < 1$ is the discount factor, over all possible policies π. In vector notation:

$$v = r_d + \lambda \mathbb{P}_d v \tag{6}$$

This means that for a policy π, the unique solution is given by:

$$v_\lambda^{d^\infty} = (I - \lambda \mathbb{P}_d)^{-1} r_d. \tag{7}$$

The second option is to maximize the average expected reward

$$v^\pi(s) = \lim_{N\to\infty} \frac{1}{N} E_s^\pi \sum_{t=1}^{N} r(s_t, d_t(s_t)) \tag{8}$$

over all possible policies π.
For both methods, there are three ways in which the MDP can be solved: value iteration, policy iteration and linear programming. These methods are described for a discounted MDP.
Using value iteration, we try to approximate the actual value $v(s)$ of each state s by repeatedly solving $v^{n+1}(s) = \max_{a\in A_s}\big(r(s,a)+\sum_{j\in S}\lambda p(j|s,a)v^n(j)\big)$ for all states $s \in S$, until $\|v^{n+1} - v^n\|$ is within an error marge. The policy is then for each state the argmax of the Bellman equation: $d(s) = \arg\max_{a\in A_s}\big(r(s,a) + \sum_{j\in S}\lambda p(j|s,a)v^{n+1}(j)\big)$. The policy as defined is ϵ-optimal.
Using policy iteration, we find the optimal policy by repeatedly obtaining v^n by solving $(I-\lambda P_{d_n})v = r_{d_n}$ and d_{n+1} by solving $d_{n+1} \in \arg\max_{d\in A_s}(r_d+\lambda P_d v^n)$, until the optimal policy is found.
To solve an MDP using linear programming, we define the primal linear program as follows:

$$\begin{aligned} \min \quad & \sum_{j\in S}\alpha(j)v(j) \\ \text{s.t.} \quad & v(s) - \sum_{j\in S}\lambda p(j|s,a)v(j) \quad \geq \quad r(s,a) \end{aligned} \tag{9}$$

where we choose $\alpha(j)$ to be positive scalars such that $\sum_{j\in S}\alpha(j) = 1$.
The dual linear program is then defined as

$$\begin{aligned} \max \quad & \sum_{s\in S}\sum_{a\in A_s} r(s,a)x(s,a) \\ \text{s.t.} \quad & \sum_{a\in A_s} x(j,a) - \sum_{s\in S}\sum_{a\in A_s}\lambda p(j|s,a)x(s,a) \quad = \quad \alpha(j) \end{aligned} \tag{10}$$

where $x(s,a) \geq 0$ for $a \in A_s$ and $s \in S$. We prefer to solve the dual program, because it has less rows. We then find the optimal policy by

$$\mathbb{P}(d(s) = a) = \frac{x(s,a)}{\sum_{a'\in A_s} x(s,a')} \tag{11}$$

where $\mathbb{P}(d(s) = a)$ is the probability of executing action a if we are in state s. (Puterman, 2014)

3.3 Possible Approximation Models

For stochastic optimization, there are two main strategies to create a policy. The first is policy search, the second is to use policies based on lookahead approximations. Both can be divided into two classes of policies.
The two approaches based on policy search are policy function approximation and cost function approximation. The first consists of analytical functions that determine an action given a state. The second uses a parameterized approximation of the cost function, under an approximation of the constraints.
The approaches based on lookahead approximations are value function approximations, where the value of being in a state is approximated, and direct lookahead policies, where is maximized over an approximation of current and future values. (Powell, 2019)

4 Related Work

This section first gives the relevant literature about scheduling in mental healthcare clinics. Then blueprint scheduling and two stationary assignment policies are discussed, which are used in Chapter 5.

4.1 Mental Healthcare

Within the mental healthcare in the Netherlands, the norms for the waiting times have not been met for some years. Therefore, the government has taken some measures to reduce the waiting times, like more collaboration between different clinics, more education spots and more transparancy about waiting times. However, the waiting times still exceed the norms as mentioned in Chapter 1. (Zorgautoriteit, 2018)
Relevant literature about managing waiting time in a mental healthcare setting is mentioned below.
Pagel et al. stated that the mental healthcare is a system running near or at capacity, claiming that it is extremely rare for a queue to fall to zero. They made a mathematical queueing model which estimates the average and variation of the number of clients treated, given a certain configuration of resources. A theoretical example is minimized over the expected increase in waiting time. (Pagel et al., 2012)
Koizumi made an open queueing model concerning interrelated mental health facilities. In this model, blocking is incorporated, such that clients might be rejected by the place they are referred to, making them remain in their present facility. It is shown that the system-wide congestion is primarily caused by shortage in one specific facility type. (Koizumi, 2002)
A multi-node, multi-server queueing system was used by Murch et al. to find the best of three different options of resource allocation after the first wave of COVID-19. (Murch et al., 2021) Carey et al. researched the effect of letting the clients schedule the appointments and therefore determine the number of appointments, rather than giving them a standard number of treatments. It appeared that the care was equivalently effective compared to the standard care, but achieved in less appointments. No clients planned appointments at regularly spaced intervals. (Carey et al., 2013) Nothing was said about the scheduling part of this experiment.
No literature about scheduling series of appointments in mental healthcare or assigning clients to practitioners has come across. These topics will be focused on in the remainder of this report.

4.2 Blueprint Scheduling

The vast majority of clients of Mediant have a fixed appointment length. Therefore, the available time can be divided into equal-length time slots. For appointments that need more time, more slots can be used. This reduces the scheduling problem to a problem of matching clients with slots, given some restrictions.

Assigning certain slots to certain appointments makes this matching problem easier. (Gupta and Denton, 2008) A blueprint schedule can be repeatedly used if there are no major changes in input, such that it does not have to be crafted from start each time.
The definition of a blueprint schedule as stated by Leeftink et al. is used: A blueprint schedule describes the capacity that can be used for a specific type of client. It can also be used to plan appointments for which more than one practitioner should be present. Different objectives could be set when designing a blueprint schedule. A blueprint schedule can be designed by mathematical programming or heuristics. (Leeftink et al., 2020)
Bikker et al. give an example of how to design a blueprint schedule using integer linear programming.(Bikker et al., 2015)

4.3 Stationary Assignment Policies

The problem of assigning clients to practitioners can be seen as assigning different types of customers to servers, with restrictions on which server is available per type. We discuss two stationary policies which are commonly used to assign clients from queues to servers: threshold policies and trunk reservation policies.

4.3.1 Threshold Policies

For $M|M|2$ queueing systems, it is shown (Lin and Kumar, 1984) that a threshold policy is optimal. Here a threshold policy is defined as a policy where the faster server always accepts customers, while the slower server only accepts customers if the queue is longer than a threshold. Viniotis and Ephremidis (Viniotis and Ephremides, 1988) extended this to a similar $GI|M|2$ model, where every set of interarrival times X_1 and X_2 satisfy $\mathbb{P}(X_1 \geq t) \geq \mathbb{P}(X_2 \geq t)$ whenever $\mathbb{E}(X_1) \geq \mathbb{E}(X_2)$, which is not a very restrictive property. Luh and Viniotis extended the results of Lin such that they proved optimality of a threshold policy in an $M|M|N$ queue, with N heterogeneous servers. (Luh and Viniotis, 2002) In this case, a server was used if and only if the queue was larger than some threshold, where each server has its own threshold.

4.3.2 Trunk Reservation Policies

We use the definition of a trunk reservation policy as introduced by Feinberg and Yang (Feinberg and Yang, 2011). They state that a trunk reservation policy is defined as a policy which has a control level M_j for each type $j \in J$, where J is the set of client types, such that a client of type j is admitted to the system if and only if the customer sees less than M_j customers in the system.
Feinberg and Yang (Feinberg and Yang, 2011) conclude that in a $M|M|k|N$ queue, the optimal policy has a trunk reservation form, under the assumptions all k servers are identical and the service times are independent of the customer type.
Maddah and El-Taha (Maddah and El-Taha, 2016) describe a Markov chain in

which two streams of customers are competing for service. They model this as a quasi-birth-death process, using selective trunk reservation. They consider two types of servers, where the protected stream is always planned at the first type of server, and the best effort stream only if there are sufficient empty servers of the first type. If the best effort stream is rejected at the first type of server, it goes to the second type of server, where again it is rejected if there are not sufficient empty servers, and accepted otherwise. If the protected stream is rejected at the first type of servers because every server is occupied, it moves to the second type of server, where it is again rejected if and only if all these servers are also occupied.

To the knowledge of the author, no literature has been published about a policy which is based on both the threshold policy and the trunk reservation policy. Such a policy is described in Section 5.5.

5 Model

We want to schedule clients that arrive at the right time at the right practitioner. To do so, we first examine the conditions for scheduling clients and see that we are able to schedule clients at a short horizon in Section 5.1. Then a blueprint schedule is designed in Section 5.2. The model for filling slots is given in Section 5.3, Section 5.4 gives a reduction of this model and Section 5.5 describes a solution method for this model.
The blueprint schedule is modelled without uncertainty, such that the options for scheduling clients are limited, and assigning clients to slots can be modelled with uncertainty. Since, given the current state, the past is irrelevant for filling slots, assigning clients to slots is modelled as a Markov Decision Process (MDP). Because of the curses of dimensionality, the MDP needs to be approximated. This gives a dynamic scheduling solution.

5.1 Schedule Short Horizon

A client needs to be notified about an appointment at least two weeks in advance. A client typically has one appointment of one hour every two weeks, preferably on the same day of the week, same time of day. When an appointment series needs to be continued, there should not be a gap of more than two weeks. Whether or not an appointment series needs to be continued can be known two appointments before the last, so four weeks before the last appointment. Because a client has to be notified at least two weeks before a new appointment, any possible new appointment needs to be made before or on the day of the last appointment scheduled.
It is possible to delay the planning of new clients until is known whether current clients need more appointments. By delaying the planning, it can be made sure that a new client is scheduled directly behind the previous client and no gaps occur. This in contrast to the current method, where gaps occur whenever a client is finished with treatments before all scheduled treatments are used. Furthermore, in the current setting it is very difficult to plan more appointments if needed, since the agenda is already fully booked. When delaying the planning, extra appointments are still possible. Therefore, short horizon scheduling prevents gaps in the schedule, while still meeting all constraints.
To easily plan clients on the same weekday and time of day, appointments slots are introduced. This means that whenever a client does not need more treatments, a new client can be planned in that slot. When these slots are known, the main problem reduces to which client to schedule when a slot opens. Therefore, a blueprint schedule is made.

5.2 Blueprint Schedule

To obtain a system in which it is easy to see whether or not there is space to schedule a client, a blueprint schedule with appointment slots as described in Section 4.2 is made.

The blueprint schedule should, next to treatment slots, also contain group treatments, intakes, meetings and time for administration/consultation. We define an integer linear program (ILP) to find the blueprint schedule for one full-time practitioner.

5.2.1 Variables and Parameters

For each time slot, the decision should be made which activity should be planned. The decision variables are:

$$x_{i,j,d} = \begin{cases} 1 & \text{if activity } i \text{ is planned at slot } j \text{ on day } d, \\ 0 & \text{else.} \end{cases}$$

Where

- $i \in I = \big(1$ (treatment), 2 (group treatment), 3 (intake), 4 (meeting), 5 (administration)$\big)$,
- $j \in J = (1, 2, \ldots, 18)$,
- $d \in D = (1, 2, \ldots, 10)$,

since for a full-time practitioner, there are 9 working hours a day, so 18 slots of half an hour. Most appointments and meetings are biweekly, so a blueprint for two weeks is designed.
Furthermore, we define the binary parameters:

$$z_{j,d} = \begin{cases} 1 & \text{if practitioner is working at time slot } j, \text{ day } d, \\ 0 & \text{else,} \end{cases}$$

and the integer parameters:

$$\begin{aligned} c_{prod} &= \text{Amount of treatment hours,} \\ c_{intake} &= \text{Amount of intakes.} \end{aligned}$$

The amount of treatment hours, meaning individual treatment slots, groups and intakes, should exceed the production standard which is set by Mediant. In contrast to the current situation, where practitioners make more hours than is asked of them, in the new blueprint we just want to obtain the production standard as mentioned in Section 2.1.2. The number of intakes is set at the same number as the average in the years 2016-2019, since in these years the production standard was obtained, and the queue not too long. When the production standard is no longer obtained, the number of intakes should be increased.

5.2.2 Objective

The objective is to maximize the amount of administration hours, such that the stress on the schedule is minimal. The goal is then

$$\max \sum_{j \in J} \sum_{d \in D} x_{5,j,d}. \tag{12}$$

5.2.3 Constraints

For each slot the the amount of planned activities should equal one if the practitioner is working and zero otherwise.

$$\sum_{i \in I} x_{i,j,d} = z_{j,d} \qquad \forall j \in J, d \in D \tag{13}$$

Since cancellations and no-shows are not counted as production hours, we have to add some extra slots to still obtain the production standard. Since the policy on cancellations is intended to be more strict, it is not known how many extra hours are needed, but we rather have too many, to prevent having to add extra hours later on. Therefore, an extra twenty percent is added to the production standard. We have to match or exceed this number in the blueprint schedule, and we have to match the number of intakes.

$$\sum_{i \in (1,2,3)} \sum_{j \in J} \sum_{d \in D} x_{i,j,d} \geq 1.2 \cdot c_{prod} \tag{14}$$

$$\sum_{j \in J} \sum_{d \in D} x_{3,j,d} = c_{intake} \tag{15}$$

Since the slots are half an hour, and treatments and intakes are one hour, we have to make sure that there are always two subsequent slots reserved for treatment:

$$x_{i,j-1,d} + x_{i,j+1,d} \geq x_{i,j,d} \qquad \forall i \in (1,3), j \in J, d \in D \tag{16}$$

After an intake, an hour of administration is needed.

$$x_{5,j+2,d} \geq x_{3,j,d} \qquad \forall j \in J, d \in D \tag{17}$$

Furthermore, meetings and group treatments are fixed, so these are manually set to one. Half an hour before lunch is fixed to administration, such that practitioners have time to consult each other and to have last-minute meetings or deal with emergencies. This at the request of the management.

Since emergencies do not happen often and are very unpredictable, it is chosen to not compensate for these hours in the production standard. Emergency care should preferably be done outside of treatment slots, so in time reserved for administration. To make sure that this is doable, it is preferred to have some time for administration at the end of each day.
Since there are many solutions giving the same, optimal objective, solving this program with a solver will not be very quick. However, since the constraints are very natural, it is easy to make a schedule by hand. An example of such a blueprint schedule is given in Figure 8. Of course, practitioners are free to make a personal blueprint, as long as the constraints are met.

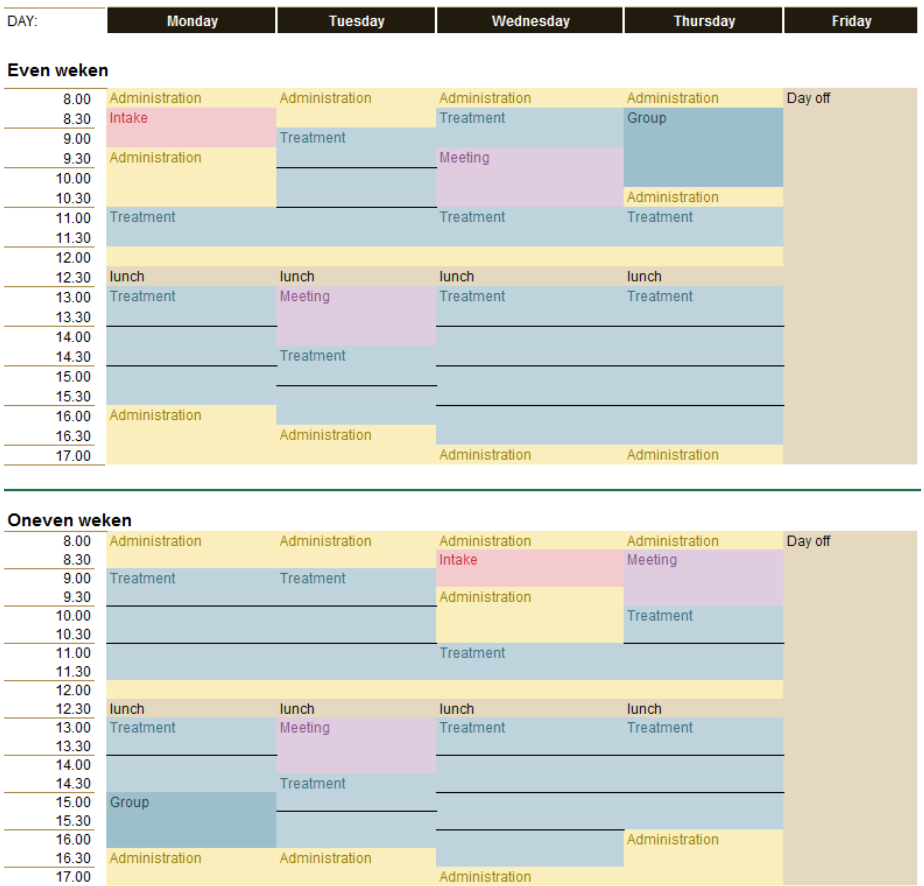

Figure 8: Example of a blueprint schedule.

5.3 Dynamic Stochastic Model for filling Slots

The sections below discuss the components of the unified framework as defined by Powell (Powell, 2019) and mentioned in Section 3.1. Since we have a stationary process, we model this as a Markov Decision Process (MDP), as described in Section 3.2.
There are no restrictions on the order when a client needs treatment from more than one of the subdivisions as given in Figure 1. Therefore, the planning of each subdivision can be modelled separately. We describe the model for the subdivision "Psychische zorg", since this is the most complicated one. Subdivision "Somatische zorg" could be modelled similarly and the subdivision "Ambulant en vaktherapie" consists of practitioners without overlap in expertise, so these practitioners should plan their own clients as described in Section 5.1 and 5.2.

5.3.1 State Variables

The state variable is defined as $s_t = (\mathbf{p}_t, \mathbf{c}_t)$, containing a vector with information about the slots of practitioners $\mathbf{p}_t$ and a vector with information about clients in the queue $\mathbf{c}_t$. Since we model this as an MDP, we drop the time indication.
The state variables are defined as follows:

- $\mathbf{p} = (\mathbf{p_1}, \mathbf{p_2}, \ldots, \mathbf{p_n})$, where n is the number of practitioners,
- $\mathbf{p_i} = ((d_{i1}, p_{i1}), (d_{i2}, p_{i2}), \ldots, (d_{ik}, p_{ik}))$, where k is the number of slots in the blueprint schedule of practitioner i, d_{ij} is the complexity of the client scheduled at slot j with practitioner i and p_{ij} is the number of appointments this client already has had. If no client is scheduled at slot j with practitioner i, $d_{ij} = p_{ij} = 0$.
- $\mathbf{c} = (c_{w,d,z}, \ldots)$, indicates for every possible combination of waiting time w, complexity d and possible practitioners z, the number c of clients in the waiting list.

5.3.2 Decision Variables

The decision variable $\mathbf{a} \in A_s$ indicates for every slot whether no one should be planned (0), or which combination of waiting time w, complexity d and possible practitioners z should be planned. Of course, in non-empty slots, the only option to plan should be 0. Furthermore, there should not be more clients planned than there are clients in the list and the allowable types of practitioners for these clients should be honored.

5.3.3 Exogenous Information

The exogenous information arriving after making a decision is the amount of clients arriving of each type and the number of clients finished with treatment.

The probability distributions of the number of treatments that are needed depend on the complexity of the client. Arrivals are assumed to be independent of the state. The probability distributions of the service times are assumed to be known from historical data, while the arrival rates are assumed to be Poisson, with an average obtained from historical data. All arrivals for treatment are only known after the MDO, so once a week. Therefore scheduling decisions should also be made once a week, after the MDO.

5.3.4 Transition Function

The state variable is updated such that for clients finished with treatment, d_{ij} and p_{ij} are set to zero. For clients not finished with treatment, p_{ij} is set to $p_{ij}+1$. When a new client is scheduled, d_{ij} is set accordingly and p_{ij} is set zero. New arrived clients are added to the queue with their complexity d and possible practitioners z and $w=0$. Clients scheduled are subtracted from the corresponding queue. All clients remaining in the queue transfer to a queue with one more week waiting time.

5.3.5 Objective Function

Since we care more about the queues now than later, we decide to maximize the total expected discounted reward. We want to minimize average waiting time, but we also want to avoid long waiting times. Therefore, we define the expected total discounted reward as

$$v_\delta^\pi(s) = \lim_{N\to\infty} \mathrm{E}^\pi \sum_{t=0}^{N} \delta^{t-1} \cdot r(s_t, d_t(s_t)) \tag{18}$$

for policy π, where $0 \geq \delta < 1$ and

$$r(s_t, a_t) = -(a_t \cdot \sum_{a_i \in a_t} w_i + \mathrm{P}_l \cdot l) \tag{19}$$

where w_i is the waiting time of client i, who is leaving the queue and l is the number of clients on the waiting list with a waiting time longer than x weeks. P_l is the weight factor determining how much penalty is given when the waiting time is too long.

5.4 Reducing the State Space

To solve this model we need to keep track of the number of clients in the queue per type of client, allowable practitioners and waiting time. Furthermore, of every client in service we need to know the type and number of appointments he has had. As a result the problem becomes intractable very fast. For example, when having four types of clients, each with three possible practitioner combinations, a maximum waiting time of 10 weeks, a maximum of 10 clients in each queue, 10 slots and a maximum of 10 treatments per client, the number of states already exceeds 10^{136}.

5.4.1 Geometric distribution

To reduce the state space, one could assume that the distribution of the number of treatment sessions needed is geometric. Since treatment sessions are almost always one hour, we assume that the total amount of treatment is a multiple of 60 minutes. However, when there are multiple clients and/or practitioners, the registered treatment time is divided pro rata in minutes, making it almost continuous. The probability plot of the data against the exponential distribution, the continuous counterpart of the geometric distribution, is shown in Figure 9. In the upper left corner the BGGZ clients are shown, the upper right corner shows the SGGZ clients and the lower left corner shows the residual care paths. The coefficients of determination are respectively 0.98, 0.98 and 0.97 for BGGZ, SGGZ and residual care paths, meaning that the model explains a lot of variation within the data.

The p-values obtained by the Kolmogorov-Smirnov test are respectively $9.656 \cdot 10^{-13}, 0.077$ and 0.177, meaning that the null-hypothesis that the distribution is exponential should be rejected for BGGZ, but not for SGGZ and the residual care paths on a 5% confidence level. The fact that BGGZ care paths are not exponentially distributed is probably because the amount of treatment sessions in BGGZ care paths is limited to a certain amount per year by health insurers, meaning that there are no long care paths. Clients that start with a BGGZ care path and have a long treatment time do exist, but after the limited number of treatments within BGGZ, an SGGZ care path is started. The fact that adding large numbers to the test set increases the p-value supports the hypothesis that treatment time of BGGZ care paths is also exponentially distributed.

The distributions of SGGZ and residual care paths thus resemble exponential distributions, and we assume that the total treatment time of clients who start in an BGGZ care path is exponentially distributed. We only model sessions of one hour, therefore we assume that we can use the geometric distribution in our model. This implies that for the slots, instead of both the type and the number of sessions completed, only the type has to be registered. The implication for the state variable is described in Section 5.4.3.

5.4.2 Objective

If the objective is adjusted, it is no longer necessary to record for each client in the queue their waiting time. Instead of minimizing the average waiting time, the spilled capacity could be minimized by giving a reward for each scheduled client. Instead of giving a penalty for each client that waits too long, a penalty could be given when a queue contains more clients than a threshold. This way, it is still avoided for the queue to become unreasonable long for some type of clients, without having to register the waiting time of each individual client. The implication for the state variable and the formulation of the new objective can be found in Section 5.4.3.

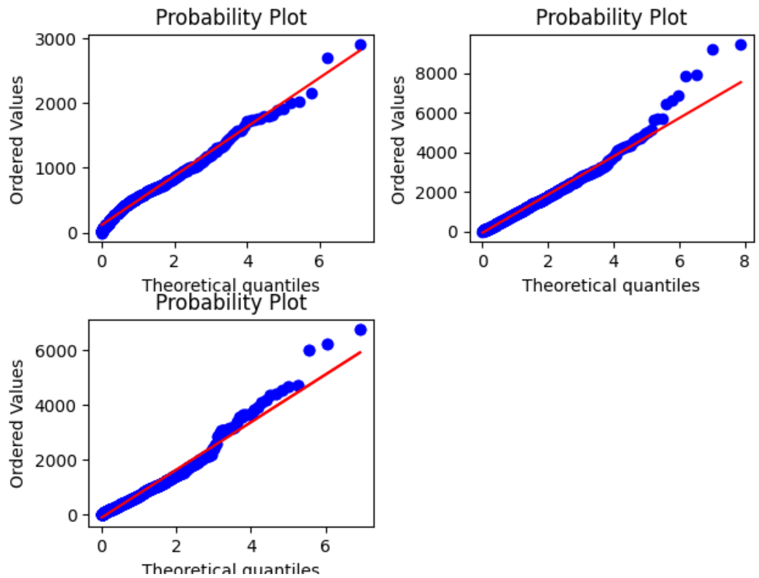

Figure 9: Probability plot per type of treatment. Upper left is BGGZ, upper right is SGGZ and lower left are residual care paths.

5.4.3 Formulation

The state variable remains a vector $(\mathbf{p}, \mathbf{c})$, where

- $\mathbf{p} = (\mathbf{p_1}, \mathbf{p_2}, \ldots, \mathbf{p_n})$, where n is the number of practitioners,
- $\mathbf{p_i} = (p_{i1}, p_{i2}, \ldots, p_{ik})$, where k is the number of slots with practitioner i and p_{ij} is the complexity of the client scheduled in slot j with practitioner i. If no client is scheduled at slot j with practitioner i, $p_{ij} = 0$.
- $\mathbf{c} = (c_{d,t}, \ldots)$ indicating for every possible combination of complexity d and possible practitioners t, the number of clients c in the waiting list.

The objective function $r(s_t, a_t)$ is updated to

$$r(s_t, a_t) = |a_t| - f_p(\mathbf{c}) \tag{20}$$

where $|a_t|$ is the number of clients scheduled from the queue and

$$f_p(\mathbf{c}) = \sum_{j \in \mathbf{c}} \begin{cases} j - N_{i,penalty} & \text{if } j > N_{i,penalty} \\ 0 & \text{else,} \end{cases}$$

indicating a penalty for every queue of type i with more clients than a threshold $N_{i,penalty}$. Since all clients are valued equally, we want the maximum waiting time for each type to be the same. Therefore, we define $N_{i,penalty}$ to be proportional to the arrival rate of type i.

With these adjustments, we have reduced the number of states in the example as mentioned before to around 10^{18}. Since this is still way too large to solve exactly and only for 10 slots while we have over 700 slots in reality, an approximation method should be used.

5.5 Policy approximation

There are several methods to approximate the exact solution of this model, which are mentioned in Section 3.3. Since it is important for Mediant that a policy is understandable and easy applicable in practice, we choose to use policy function approximation, where we can optimise over a simple policy.
We use respectively KP, GZ, BP and SPV as abbreviations for the types of practitioners as given in Figure 1. The service rate of practitioner type i is given by μ_i and is defined as the total number of slots available for type i practitioners, divided by the average number of appointments needed per client.
To keep the policy simple, no distinction is made between different types of clients other than the possible practitioners. This reduces the state space in the small example mentioned before to about 10^6 states.
We call clients which can only be treated by a KP type A, clients who can also be treated by a GZ type B, then type C can be treated by a BP and higher and type D can be treated by all of them. Furthermore, we have clients who can only be treated by SPV, type E. This is made visible in Figure 10, where the red lines indicate the assignment to the lowest ranked practitioner possible, and therefore the default practitioner. The arrival rate of a client of type i is defined as λ_i.

In a high-load system like the department S&A at Mediant, it is reasonable to assume that filling as much slots as possible is a good approximation. Since we do not model different service times for different types, we can reason that, since $r(s,a) = |a_t| - \mathrm{P}_q \cdot q$ and $|a_t|$ is always the minimum between the number of open slots and the number of clients in the queue, the main goal is to minimize q. The main trade-off is then between clients who have a high expected arrival and clients who are difficult to plan.

We assume that KP always plans clients of type A or B, and GZ plans clients of type B or C. We could then decompose the system into four subsystems, AB, BC, CD and DE. The first subsystem AB consists of the blue box in Figure 11, subsystem BC is the green box, CD is red and DE purple. Note that subsystems AB, BC and CD are similar, but subsystem DE is slightly different. The capacity of SPV is virtually split to be able to model this subsystem similar to the other subsystems. Section 5.5.1 and 5.5.2 describe which clients to plan in respectively subsystem AB (subsystems BC and CD are similar) and subsystem DE, when only considering this single subsystem. Section 5.5.3 gives a description of how these subsystems can be combined, by in subsystem AB taking the amount of clients of type C that are treated by GZ as found in subsystem BC into consideration. In subsystem BC, both the amount of clients of type B treated by KP and the amount of clients of type D treated by BP need to be taken into consideration. The other subsystems can be combined

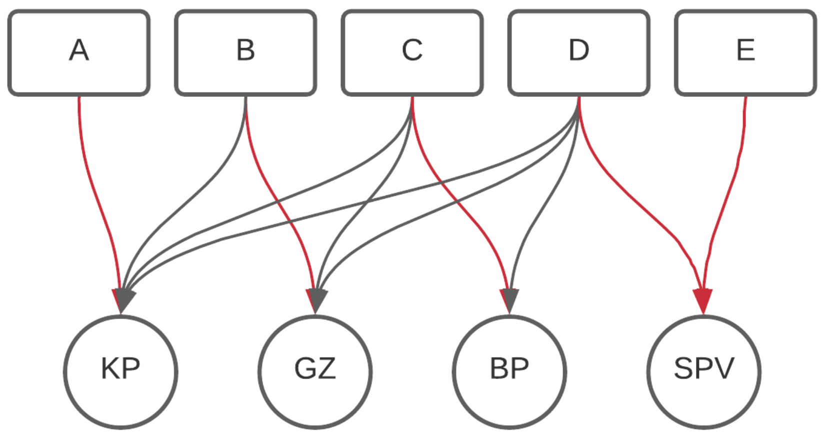

Figure 10: Possible assignments of different types of clients.

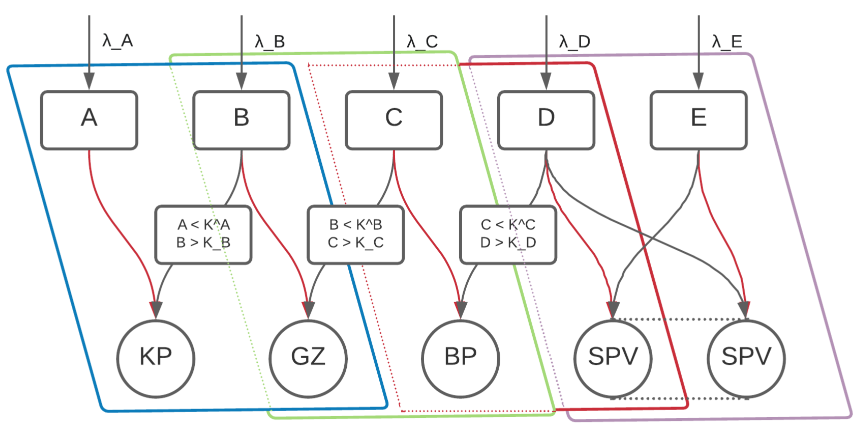

Figure 11: Subsystems.

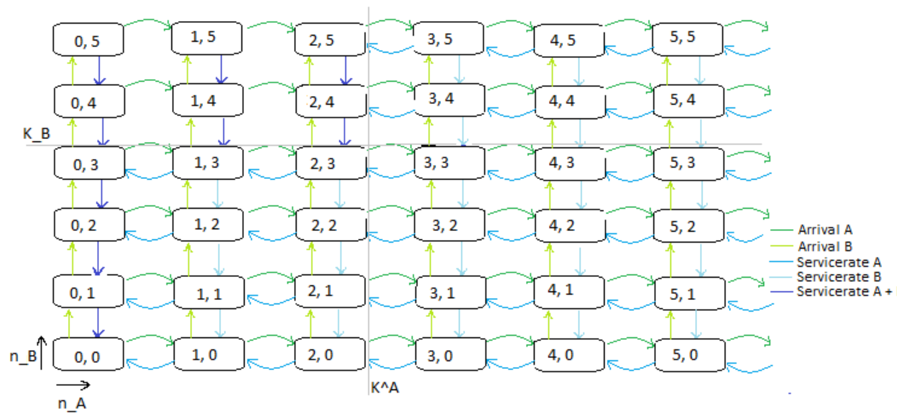

Figure 12: Quasi-birth-death process of subsystem AB.

similarly.

5.5.1 Analysis subsystem AB

First subsystem AB is considered. Let n_A denote the number of type A clients in queue and n_B the number of type B clients in queue. Normally, clients of type A are treated by KP, while clients of type B are treated by GZ. However, KP could also treat clients of type B. This might be useful when n_B is quite large, while n_A is relatively small. Furthermore, when $n_A = 0$ and $n_B \geq 1$, it is also reasonable that KP treats a client of type B. To ensure this type of behaviour, a combination of a threshold policy and adapted trunk reservation control policy is used. These policies are described in more detail in Section 4.3. We define a threshold K_B such that clients of type B are sent to KP if $n_B > K_B$. If clients of type B are sent to KP, these clients are to be accepted or not. If they are, they have priority on clients of type A. To prevent the queue of clients of type A from becoming too long, there should be clients rejected. The classic trunk reservation control policy is to reject a client if the capacity is below some point. Translating to this situation, we reject clients if n_A is larger than a threshold K^A. Using this policy and assuming exponential arrival and service rates, we have a Markov chain, which can be analyzed using a quasi-birth-death process as described by Van Leeuwaarden et al. (Van Leeuwaarden and Winands, 2006). This quasi-birth-death process is given in Figure 12 for a fixed K^A and K_B. The states are defined as (n_A, n_B) and are ordered lexicographically, that is ((0,0), (0, 1), (0, 2), $\ldots$, (0, n), (1, 0), (2, 0), $\ldots$, (2, n), $\ldots$). The set of states where $n_A \leq K^A$ is called *level* 0, and the set of states $((i + K^A, 0), (i + K^A, 1), \ldots)$ $i = 1, 2, \ldots$ is called *level* i.
Note that in this model n_B is cut-off at n, while n_A is not. It makes sense to cut-off a different type of client in every subsystem, so either n_A in AB, n_B in BC, n_C in CD and n_D in DE or n_B in AB, n_C in BC, n_D in CD and n_E in

DE. If n_A is cut-off, there is a chance that the optimal solution is to deviate a lot from the default practitioner, and to just 'give up' on clients of type A, since the queue cannot grow. Vice versa is not possible, so we choose to not cut-off n_A.
The generator Q is of the form

$$Q^{AB} = \begin{pmatrix} B_{00}^{AB} & B_{01}^{AB} & & & & 0 \\ B_{10}^{AB} & A_1^{AB} & A_0^{AB} & & & \\ & A_2^{AB} & A_1^{AB} & A_0^{AB} & & \\ & & A_2^{AB} & A_1^{AB} & A_0^{AB} & \\ 0 & & & . & . & . \end{pmatrix}$$

For ease of notation, let $2\lambda = \lambda_A + \lambda_B$ and $2\mu = \mu_{KP} + \mu_{GZ}$. The matrices $B_{00}^{AB}, B_{01}^{AB}, B_{10}^{AB}, A_0^{AB}, A_1^{AB}$ and A_2^{AB} are for the example in Figure 12 defined as follows:

$$B_{00}^{AB} = \begin{pmatrix}
-2\lambda & \lambda_B & 0 & 0 & 0 & 0 & \lambda_A & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2\mu & -2(\lambda+\mu) & \lambda_B & 0 & 0 & 0 & 0 & \lambda_A & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 2\mu & -2(\lambda+\mu) & \lambda_B & 0 & 0 & 0 & 0 & \lambda_A & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2\mu & -2(\lambda+\mu) & \lambda_B & 0 & 0 & 0 & 0 & \lambda_A & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 2\mu & -2(\lambda+\mu) & \lambda_B & 0 & 0 & 0 & 0 & \lambda_A & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 2\mu & -(\lambda_A+2\mu) & 0 & 0 & 0 & 0 & 0 & \lambda_A & 0 & 0 & 0 & 0 & 0 & 0 \\
\mu_{KP} & 0 & 0 & 0 & 0 & 0 & -(2\lambda+\mu_{KP}) & \lambda_B & 0 & 0 & 0 & 0 & \lambda_A & 0 & 0 & 0 & 0 & 0 \\
0 & \mu_{KP} & 0 & 0 & 0 & 0 & \mu_{GZ} & -2(\lambda+\mu) & \lambda_B & 0 & 0 & 0 & 0 & \lambda_A & 0 & 0 & 0 & 0 \\
0 & 0 & \mu_{KP} & 0 & 0 & 0 & 0 & \mu_{GZ} & -2(\lambda+\mu) & \lambda_B & 0 & 0 & 0 & 0 & \lambda_A & 0 & 0 & 0 \\
0 & 0 & 0 & \mu_{KP} & 0 & 0 & 0 & 0 & \mu_{GZ} & -2(\lambda+\mu) & \lambda_B & 0 & 0 & 0 & 0 & \lambda_A & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2\mu & -2(\lambda+\mu) & \lambda_B & 0 & 0 & 0 & 0 & \lambda_A & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2\mu & -(\lambda_A+2\mu) & 0 & 0 & 0 & 0 & 0 & \lambda_A \\
0 & 0 & 0 & 0 & 0 & 0 & \mu_{KP} & 0 & 0 & 0 & 0 & 0 & -(2\lambda+\mu_{KP}) & \lambda_B & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_{KP} & 0 & 0 & 0 & 0 & \mu_{GZ} & -2(\lambda+\mu) & \lambda_B & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_{KP} & 0 & 0 & 0 & 0 & \mu_{GZ} & -2(\lambda+\mu) & \lambda_B & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_{KP} & 0 & 0 & 0 & 0 & \mu_{GZ} & -2(\lambda+\mu) & \lambda_B & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2\mu & -2(\lambda+\mu) & \lambda_B \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2\mu & -(\lambda_A+2\mu)
\end{pmatrix}$$

$$B_{10}^{AB} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_{KP} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_{KP} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_{KP} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_{KP} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_{KP} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_{KP}
\end{pmatrix}$$

$$B_{01}^{AB} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\lambda_A & 0 & 0 & 0 & 0 & 0 \\
0 & \lambda_A & 0 & 0 & 0 & 0 \\
0 & 0 & \lambda_A & 0 & 0 & 0 \\
0 & 0 & 0 & \lambda_A & 0 & 0 \\
0 & 0 & 0 & 0 & \lambda_A & 0 \\
0 & 0 & 0 & 0 & 0 & \lambda_A
\end{pmatrix}$$

$$A_0^{AB} = \begin{pmatrix} \lambda_A \cdot I_6 \end{pmatrix}$$
$$A_1^{AB} = \begin{pmatrix} -(2\lambda+\mu_{KP}) & \lambda_B & 0 & 0 & 0 & 0 \\ \mu_{GZ} & -2(\lambda+\mu) & \lambda_B & 0 & 0 & 0 \\ 0 & \mu_{GZ} & -2(\lambda+\mu) & \lambda_B & 0 & 0 \\ 0 & 0 & \mu_{GZ} & -2(\lambda+\mu) & \lambda_B & 0 \\ 0 & 0 & 0 & \mu_{GZ} & -2(\lambda+\mu) & \lambda_B \\ 0 & 0 & 0 & 0 & \mu_{GZ} & -(\lambda_A+2\mu) \end{pmatrix}$$
$$A_2^{AB} = \begin{pmatrix} \mu_{KP} \cdot I_6 \end{pmatrix},$$
where I_6 is the 6-by-6 identity matrix. If Q is ergodic, meaning that $\lambda_A < \mu_{KP}$ we find the equilibrium probability vectors π_i^{AB}. For ease of notation, the superscript AB is dropped with matrices $R, B_{00}, B_{10}, B_{01}, A_0, A_1$ and A_2.

$$\pi_i^{AB} = (\pi_{i,0}^{AB}, \pi_{i,1}^{AB}, \ldots, \pi_{i,n}^{AB}) = \pi_1^{AB} \cdot R^{i-1} \qquad i = 1, 2, \ldots \tag{21}$$

where the rate matrix R is found by calculating

$$R_{k+1} = -(A_0 + R_k^2 A_2) A_1^{-1} \tag{22}$$

until $|R_{k+1} - R_k| < \epsilon$, with ϵ being a small number.
To find the value of π_1, we need to solve the system of equations:

$$\begin{aligned} \pi_0^{AB} B_{00} + \pi_1^{AB} B_{10} \quad &= 0 \\ \pi_0^{AB} B_{01} + \pi_1^{AB} A_1 + \pi_1^{AB} R A_2 \quad &= 0 \end{aligned} \tag{23}$$

and the normalization equation

$$1 = \sum_{i=0}^{\infty} \pi_i^{AB} e, \tag{24}$$

where e is an all-ones vector.
$\pi_{i,j}^{AB}$ is then the proportion of time the Markov chain is in state $(n_A = i + K^A, n_B = j)$ for $i \geq 1$. For $i = 0$, $\pi_{0,j}$ is the proportion of time that the Markov chain is in the state at position j of the lexicograpically ordered vector ((0, 0), (0, 1), …, (0, n), (1, 0), (1, 1), …, (1, n), …, (K^A, 0), (K^A, 1), …, (K^A, n)). We define

$$\pi^{AB}(n_A, n_B) = \begin{cases} \pi_{n_A - K^A, n_B}^{AB} & \text{if } n_A > K^A, \\ \pi_{0, n_A \cdot n + n_B}^{AB} & \text{else} \end{cases} \tag{25}$$

giving the proportion of time that the Markov chain is in state (n_A, n_B). When we know the proportion of time the Markov chain is in each state, we can calculate the expected penalty by

$$\mathbb{E}[\text{penalty}^{AB}] = \sum_{n_A=0}^{\infty} \sum_{n_B=0}^{n} \pi^{AB}(n_A, n_B) \cdot f_p((n_A, n_B)). \tag{26}$$

We then find K^A and K_B by

$$\underset{0 \leq K^A \leq n}{\arg\min} \; \underset{0 \leq K_B \leq n}{\arg\min} \; \mathbb{E}[\text{penalty}^{AB}]. \tag{27}$$

To make this solvable in practice, the value of K^A is maximal n. Similarly, subsystems BC and CD can be solved.

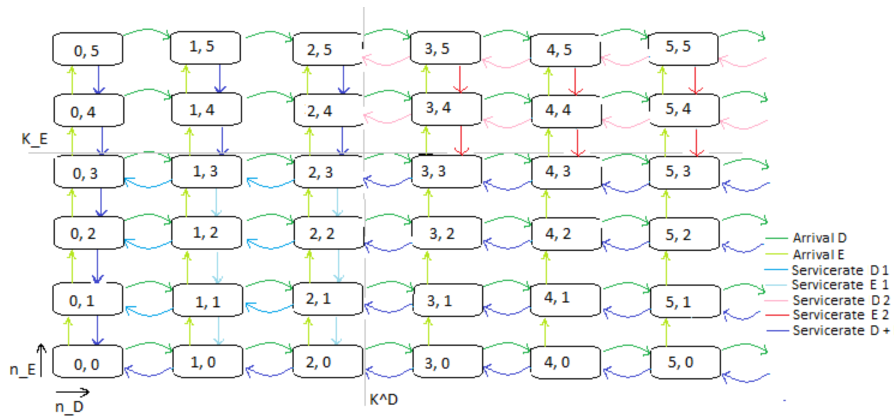

Figure 13: Quasi-birth-death process of subsystem DE.

5.5.2 Analysis of subsystem DE

Because there is no separate server for clients of type E, subsystem DE has to be modelled slightly different than the other subsystems. In contrast to the other subsystems, when $n_D > K^D$ and $n_E \leq K_E$, all capacity is used for clients of type D. This was not possible in the other subsystems, but is here. Because clients of type D and E are valued equally, the choice is made to make this subsystem as symmetric as possible.
This subsystem is also modelled as a quasi-birth-death process, which is given in Figure 13 for fixed values of K^D and K_E. The states are defined (n_D, n_E) and ordered and leveled as in Section 5.5.1. Here we have split the capacity of SPV practitioners into a part for D and a part for E, such that $\mu_D + \mu_E = \mu_{SPV}$. For states with $n_D \leq K^D$ and $n_E \leq K_E$ we define the service rates μ_{D1} and μ_{E1} and for states with $n_D > K^D$ and $n_E > K_E$ we define the service rates μ_{D2} and μ_{E2}, where $\mu_{D1} + \mu_{E1} = \mu_{D2} + \mu_{E2} = \mu_{SPV}$. Since there are no requirements about the partition other than the aforementioned and $\mu_{D1} > 0, \mu_{D2} > 0, \mu_{E1} > 0$ and $\mu_{E2} > 0$, there is optimised over the partitions. The generator Q is then of the form

$$Q^{DE} = \begin{pmatrix} B_{00}^{DE} & B_{01}^{DE} & & & 0 \\ B_{10}^{DE} & A_1^{DE} & A_0^{DE} & & \\ & A_2^{DE} & A_1^{DE} & A_0^{DE} & \\ & & A_2^{DE} & A_1^{DE} & A_0^{DE} \\ 0 & & & . & . & . \end{pmatrix}.$$

Let $2\lambda = \lambda_D + \lambda_E$ and $\mu_{SPV} = 2\mu$. The matrices $B_{00}^{DE}, B_{01}^{DE}, B_{10}^{DE}, A_0^{DE}, A_1^{DE}$ and A_2^{DE} are for the example in Figure 13 defined as follows:

$$
B_{00}^{DE} = \begin{pmatrix}
-2\lambda & \lambda_E & 0 & 0 & 0 & 0 & \lambda_D & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2\mu & -2(\lambda+\mu) & \lambda_E & 0 & 0 & 0 & 0 & \lambda_D & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 2\mu & -2(\lambda+\mu) & \lambda_E & 0 & 0 & 0 & 0 & \lambda_D & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2\mu & -2(\lambda+\mu) & \lambda_E & 0 & 0 & 0 & 0 & \lambda_D & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 2\mu & -2(\lambda+\mu) & \lambda_E & 0 & 0 & 0 & 0 & \lambda_D & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 2\mu & -(\lambda_D+2\mu) & 0 & 0 & 0 & 0 & 0 & \lambda_D & 0 & 0 & 0 & 0 & 0 & 0 \\
2\mu & 0 & 0 & 0 & 0 & 0 & -2(\lambda+\mu) & \lambda_E & 0 & 0 & 0 & 0 & \lambda_D & 0 & 0 & 0 & 0 & 0 \\
0 & \mu_{D1} & 0 & 0 & 0 & 0 & \mu_{E1} & -2(\lambda+\mu) & \lambda_E & 0 & 0 & 0 & 0 & \lambda_D & 0 & 0 & 0 & 0 \\
0 & 0 & \mu_{D1} & 0 & 0 & 0 & 0 & \mu_{E1} & -2(\lambda+\mu) & \lambda_E & 0 & 0 & 0 & 0 & \lambda_D & 0 & 0 & 0 \\
0 & 0 & 0 & \mu_{D1} & 0 & 0 & 0 & 0 & \mu_{E1} & -2(\lambda+\mu) & \lambda_E & 0 & 0 & 0 & 0 & \lambda_D & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2\mu & -2(\lambda+\mu) & \lambda_E & 0 & 0 & 0 & 0 & \lambda_D & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2\mu & -(\lambda_D+2\mu) & 0 & 0 & 0 & 0 & 0 & \lambda_D \\
0 & 0 & 0 & 0 & 0 & 0 & 2\mu & 0 & 0 & 0 & 0 & 0 & -2(\lambda+\mu) & \lambda_E & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_{D1} & 0 & 0 & 0 & 0 & \mu_{E1} & -2(\lambda+\mu) & \lambda_E & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_{D1} & 0 & 0 & 0 & 0 & \mu_{E1} & -2(\lambda+\mu) & \lambda_E & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_{D1} & 0 & 0 & 0 & 0 & \mu_{E1} & -2(\lambda+\mu) & \lambda_E & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2\mu & -2(\lambda+\mu) & \lambda_E \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2\mu & -(\lambda_D+2\mu)
\end{pmatrix}
$$

$$
B_{10}^{DE} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2\mu & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2\mu & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2\mu & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2\mu & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_{D2} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_{D2}
\end{pmatrix}
$$

$$
B_{01}^{DE} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\lambda_D & 0 & 0 & 0 & 0 & 0 \\
0 & \lambda_D & 0 & 0 & 0 & 0 \\
0 & 0 & \lambda_D & 0 & 0 & 0 \\
0 & 0 & 0 & \lambda_D & 0 & 0 \\
0 & 0 & 0 & 0 & \lambda_D & 0 \\
0 & 0 & 0 & 0 & 0 & \lambda_D
\end{pmatrix}
$$

$A_0^{DE} = \begin{pmatrix}\lambda_D \cdot I_6\end{pmatrix}$

$A_1^{DE} = \begin{pmatrix} -2(\lambda+\mu) & \lambda_E & 0 & 0 & 0 & 0 \\ 0 & -2(\lambda+\mu) & \lambda_E & 0 & 0 & 0 \\ 0 & 0 & -2(\lambda+\mu) & \lambda_E & 0 & 0 \\ 0 & 0 & 0 & -2(\lambda+\mu) & \lambda_E & 0 \\ 0 & 0 & 0 & \mu_{E2} & -2(\lambda+\mu) & \lambda_E \\ 0 & 0 & 0 & 0 & \mu_{E2} & -(\lambda_D+2\mu) \end{pmatrix}$

$A_2^{DE} = \begin{pmatrix} 2\mu & 0 & 0 & 0 & 0 & 0 \\ 0 & 2\mu & 0 & 0 & 0 & 0 \\ 0 & 0 & 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & 2\mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu_{D2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu_{D2} \end{pmatrix}$

If Q^{DE} is ergodic, meaning that $\lambda_D + \lambda_E < \mu_{SPV}$ we find the equilibrium probability vectors π_i^{DE}. For ease of notation, the superscript DE is dropped with matrices $R, B_{00}, B_{10}, B_{01}, A_0, A_1$ and A_2.

$$\pi_i^{DE} = (\pi_{i,0}^{DE}, \pi_{i,1}^{DE}, \ldots, \pi_{i,n}^{DE}) = \pi_1^{DE} \cdot R^{i-1} \qquad i = 1, 2, \ldots, \tag{28}$$

where we find R using Equations (22), (23) and (24). We define

$$\pi^{DE}(n_D, n_E) = \begin{cases} \pi_{n_D - K^D, n_E}^{DE} & \text{if } n_D > K^D, \\ \pi_{0, n_D \cdot n + n_E}^{DE} & \text{else} \end{cases} \tag{29}$$

giving the proportion of time that the Markov chain is in state (n_D, n_E). Note that in this model n_E is cut-off at n, while n_D is not.
When we know the proportion of time the Markov chain is in each state, we can calculate the expected penalty by

$$\mathbb{E}[\text{penalty}^{DE}] = \sum_{n_D=0}^{\infty} \sum_{n_E=0}^{n} \pi^{DE}(n_D, n_E) \cdot f_p((n_D, n_E)). \tag{30}$$

We first find the optimal values of μ_{D1} and μ_{D2} by

$$\underset{0 \le \mu_{D1} \le \mu_{SPV}}{\arg\min} \; \underset{0 \le \mu_{D2} \le \mu_{SPV}}{\arg\min} \; \mathbb{E}[\text{penalty}^{DE}], \tag{31}$$

and then find the optimal values of K^D and K_E by

$$\underset{0 \le K^D \le n}{\arg\min} \; \underset{0 \le K_E \le n}{\arg\min} \; \mathbb{E}[\text{penalty}^{DE}], \tag{32}$$

using μ_{D1} and μ_{D2}. To make this solvable in practice, the value of K^D is maximal n.

5.5.3 Combining subsystems

This section characterizes the relation between the subsystems. We first examine the transition from subsystem AB to subsystem BC.
From subsystem AB we know the probability that KP accepts clients of type B, which is given by

$$\mathbb{P}(KP, B) = \sum_{i=0}^{K^A} \sum_{j=K_B+1}^{n} \pi^{AB}(i, j). \tag{33}$$

This is equal to the fraction of time that KP accepts clients of type B. There are two ways in which we can use this knowledge in subsystem BC to approximate the values of $\pi^{BC}(n_B, n_C)$: The first one being to add $d_{1,f} = \mathbb{P}(KP, B) \cdot \mu_{KP}$ to the service rate of GZ:

$$\hat{\mu}_{GZ,f} = \mu_{GZ} + d_{1,f}; \tag{34}$$

The second possibility is to remove the number of clients treated by KP from the arrival rate

$$\hat{\lambda}_{B,f} = \lambda_B - d_{1,f}. \tag{35}$$

So either we add the capacity of KP that is used for clients of type B, meaning that we see them this fraction of time as GZ and they could also treat clients of type C. This is in line with Figure 10. Or we remove the clients treated by KP from the subsystem BC altogether. Whenever KP is treating clients of type B, $n_B \geq 1$ and the throughput rate is equal to the service rate μ_{KP}. Thus the number of clients treated by KP is $\mathbb{P}(KP, B) \cdot \mu_{KP} = d_{1,f}$.
If we analyze just the arrivals of type B and the service of GZ as an $M|M|1$ queue, we find that adjusting the service rate always gives an equal or higher expected queue length, denoted by $\mathbb{E}[L_{f,\hat{\mu}}]$, compared to the expected queue length obtained by adjusting the arrival rate, denoted $\mathbb{E}[L_{f,\hat{\lambda}}]$. This is true because

$$\begin{array}{rrcl} & \mathbb{E}[L_{f,\hat{\mu}}] & \geq & \mathbb{E}[L_{f,\hat{\lambda}}] \\ \iff & \frac{\rho_{\hat{\mu}}}{1-\rho_{\hat{\mu}}} & \geq & \frac{\rho_{\hat{\lambda}}}{1-\rho_{\hat{\lambda}}} \\ \iff & \frac{\lambda_B}{\hat{\mu}_{GZ,f}} & \geq & \frac{\hat{\lambda}_{B,f}}{\mu_{GZ}} \\ \iff & \frac{\lambda_B}{\mu_{GZ}+d_{1,f}} & \geq & \frac{\lambda_B - d_{1,f}}{\mu_{GZ}} \\ \iff & \lambda_B \cdot \mu_{GZ} & \geq & \lambda_B \cdot \mu_{GZ} - \mu_{GZ} \cdot d_{1,f} + \lambda_B \cdot d_{1,f} - d_{1,f}^2 \\ \iff & \lambda_B \cdot \mu_{GZ} & \geq & \lambda_B \cdot \mu_{GZ} + \underbrace{(\lambda_B - \mu_{KP})}_{<0} \cdot \underbrace{d_{1,f}}_{\geq 0} - \underbrace{d_{1,f}^2}_{\geq 0} . \Box \end{array} \tag{36}$$

Transitions from subsystem BC to CD and from CD to DE are similar.

We now examine the transition from subsystem BC to subsystem AB. When we solved subsystem BC, we know the fraction of time that clients of type C are send to GZ, given by

$$\mathbb{P}(GZ, C) = \sum_{i=0}^{K^B} \sum_{j=K_C+1}^{n} \pi^{BC}(i, j). \tag{37}$$

With similar reasoning as before, we find $d_{2,b} = \mathbb{P}(GZ, C) \cdot \mu_{GZ}$ clients are sent to GZ. Again, there are two ways to use this knowledge in subsystem AB: The first one is to remove the capacity used by these clients from the total capacity of GZ:

$$\hat{\mu}_{GZ,b} = \mu_{GZ} - d_{2,b}; \tag{38}$$

The second option is to add these clients to the expected arrival of clients of type B:

$$\hat{\lambda}_{B,b} = \lambda_B + d_{2,b}. \tag{39}$$

It is important to note that with adjusting the arrival rate, the clients originally from type C are now seen as type B can then also be send to KP if the system is really busy, which is in line with Figure 10.

If we analyze just the arrivals of type B and the service of GZ as an $M|M|1$ queue, we find the following, as proven in (40):

Adjusting the service rate gives an equal or higher expected queue length, denoted by $\mathbb{E}[L_{b,\hat{\mu}}]$ compared to to the expected queue length obtained by adjusting the arrival rate, denoted $\mathbb{E}[L_{b,\hat{\lambda}}]$, if and only if $\mathbb{P}(GZ, C) \geq 1 - \frac{\lambda_B}{\mu_{GZ}}$.

$$\begin{aligned}
& & \mathbb{E}[L_{b,\hat{\mu}}] &\geq \mathbb{E}[L_{b,\hat{\lambda}}] \\
&\iff & \tfrac{\rho_{\hat{\mu}}}{1-\rho_{\hat{\mu}}} &\geq \tfrac{\rho_{\hat{\lambda}}}{1-\rho_{\hat{\lambda}}} \\
&\iff & \tfrac{\lambda_B}{\hat{\mu}_{GZ,b}} &\geq \tfrac{\hat{\lambda}_{B,b}}{\mu_{GZ}} \\
&\iff & \tfrac{\lambda_B}{\mu_{GZ}-d_{2,b}} &\geq \tfrac{\lambda_B+d_{2,b}}{\mu_{GZ}} \\
&\iff & \lambda_B \cdot \mu_{GZ} &\geq \lambda_B \cdot \mu_{GZ} + \mu_{GZ} \cdot d_{2,b} - \lambda_B \cdot d_{2,b} - d_{2,b}^2 \\
&\iff & \lambda_B \cdot \mu_{GZ} &\geq \lambda_B \cdot \mu_{GZ} + (\mu_{GZ} - \lambda_B - d_{2,b}) \cdot d_{2,b} \\
&\iff & \lambda_B \cdot \mu_{GZ} &\geq \lambda_B \cdot \mu_{GZ} + \underbrace{\Big(\big(1-\mathbb{P}(GZ,C)\big)\mu_{GZ} - \lambda_B \Big)}_{\leq 0 \iff \mathbb{P}(GZ,C)\cdot\mu_{GZ} \geq \mu_{GZ}-\lambda_B} \cdot \underbrace{\mathbb{P}(GZ,C) \cdot \mu_{GZ}}_{\geq 0}
\end{aligned} \tag{40}$$

Transitions from subsystem CD to BC are similar.

The transition from subsystem DE to subsystem CD, however, is slightly different because clients of type E cannot be treated by BP or higher. Therefore, we do not add clients of type E to λ_D, but we find the total capacity of SPV that is used for type D in subsystem DE and use that as the service capacity in subsystem CD, keeping the original arrival rate λ_D. We find the capacity of SPV used for clients of type D by

$$\begin{aligned}
\hat{\mu}_{D,b} = &\sum_{n_D=0}^{K^D} \pi^{DE}(n_D, 0) \cdot \mu_{SPV} + \sum_{n_D=0}^{K^D} \sum_{n_E=1}^{K_E} \pi^{DE}(n_D, n_E) \cdot \mu_{D1} + \\
&\sum_{n_D=K^D+1}^{\infty} \sum_{n_E=0}^{K_E} \pi^{DE}(n_D, n_E) \cdot \mu_{SPV} + \sum_{n_D=K^D+1}^{\infty} \sum_{n_E=K_E}^{n} \pi^{DE}(n_D, n_E) \cdot \mu_{D2}.
\end{aligned} \tag{41}$$

5.5.4 Algorithms to obtain Thresholds

We want to iteratively solve subsystems AB, BC, CD and DE, each time using the knowledge of the previous subsystem. For ease of notation, we denote $A = 1, B = 2, C = 3, D = 4$ and $E = 5$, and let the subsystems be defined by

the number corresponding to the first letter, such that $AB = 1$. Furthermore, we define KP=1, GZ=2, BP=3 and SPV=4.
We define the function $f_{p,i}(K^i, K_{i+1}, \lambda_i, \lambda_{i+1}, \mu_i, \mu_{i+1})$, to have as output the expected penalty of subsystem $i \in (1, 2, 3)$.
Furthermore, we define the function $f_{p,4}(K^4, K_5, \lambda_4, \lambda_5, \mu_4, \mu_{D1}, \mu_{D2})$, which has the expected penalty of subsystem 4 as output.
We find the expected penalty as described in Section 5.5.1 respectively 5.5.2.

An initial approximation for K^1 and K_2 is set by finding the argmin of $f_{p,i}$ for $i = 1$. We then set K^i and K_{i+1} to the argmin of $f_{p,i}$ for $i = 2, 3$ using either $\hat{\lambda}_{i,f}$ or $\hat{\mu}_{i,f}$ as adjusted input. For subsystem 4, first the optimal values of μ_{D1} and μ_{D2} are calculated and then, using these values, the optimal values of K^4 and K_5 are calculated. These are found by finding the argmin of $f_{p,4}$ based on either $\hat{\lambda}_{i,f}$ or $\hat{\mu}_{i,f}$. We solve subsystems 3 and 2, using $\hat{\mu}_{4,b}$ for subsytem 3, either $\hat{\mu}_{i+1,b}$ or $\hat{\lambda}_{i+1,b}$ for subsystem 2 and either $\hat{\lambda}_{i,f}$ or $\hat{\mu}_{i,f}$ for both. We then solve subsystem 1 using either $\hat{\mu}_{2,b}$ or $\hat{\lambda}_{2,b}$. We iterate this until all thresholds are stable. We have then found the fixed point of this system.
We have to set an initial value of K^4 and K_5 to obtain the first values of μ_{D1} and μ_{D2}. If $K^4 = n$ and/or $K_5 = n$ or $K^4 = 0$ and/or $K_5 = 0$, then either μ_{D1} or μ_{D2} is unused in subsystem 4. To find a reasonable value for both μ_{D1} and μ_{D2}, we set both K^4 and K_5 halfway 0 and n.
We define Algorithm 1 to obtain all thresholds using Equation (34) in subsystem $i \in (2, 3, 4)$ and Equation (38) in subsystem $i \in (1, 2)$ if $\mathbb{P}(i+1, i+2) \geq 1 - \frac{\lambda_{i+1}}{\mu_{i+1}}$ and Equation (39) else. Algorithm 2 obtains the thresholds using Equation (35) in subsystem $i \in (2, 3, 4)$ and Equation (38) in subsystem $i \in (1, 2)$ if $\mathbb{P}(i + 1, i + 2) < 1 - \frac{\lambda_{i+1}}{\mu_{i+1}}$ and Equation (39) else.
Due to the results in Equations (36) and (40) we expect Algorithm 1 to overestimate the expected queue length and Algorithm 2 to underestimate the expected queue length.

The policy is then defined as follows:
For KP plan type A unless $n_A \leq K^A$ and $n_B > K_B$, in that case plan type B.
For GZ plan type B unless $n_B \leq K^B$ and $n_C > K_C$, in that case plan type C.
For BP plan type C unless $n_C \leq K^C$ and $n_D > K_D$, in that case plan type D.
For SPV:

$$\begin{array}{ll}
\text{if } n_D > K^D \text{ and } n_E \leq K_E & \text{plan type } D \\
\text{if } n_D > K^D \text{ and } n_E > K_E & \begin{cases} \text{w.p. } \frac{\mu_{D2}}{\mu_{SPV}} & \text{plan type } D \\ \text{w.p. } \frac{\mu_{E2}}{\mu_{SPV}} & \text{plan type } E \end{cases} \\
\text{if } n_D \leq K^D \text{ and } n_E > K_E & \text{plan type } E \\
\text{if } n_D \leq K^D \text{ and } n_E \leq K_E & \begin{cases} \text{w.p. } \frac{\mu_{D1}}{\mu_{SPV}} & \text{plan type } D \\ \text{w.p. } \frac{\mu_{E1}}{\mu_{SPV}} & \text{plan type } E \end{cases}
\end{array}$$

The remaining empty slots are filled with a client in the highest possible category, type A being the highest.

Algorithm 1: Combining the subsystems with maximal expected queue length

Result: $K_{total} = (K^1, K_2, K^2, K_3, K^3, K_4, K^4, K_5)$, μ_{D1}, μ_{D2}

Initialization:

$K^4, K_5 = \lfloor 0.5n \rfloor, \lfloor 0.5n \rfloor$

$K^1, K_2 := \arg\min_{0 \geq K^1 \geq n} \arg\min_{0 \geq K_2 \geq n} f_{p,1}(K^1, K_2, \lambda_1, \lambda_2, \mu_1, \mu_2)$

$d_1 := \mu_1 \cdot \sum_{j=0}^{K^1} \sum_{k=K_2+1}^{n} \pi^1(j,k)$

$c := 1$

for $i \in (2,3,4)$ **do**

  $\hat{\lambda}_{i,b} := \hat{\lambda}_{i,f} := \lambda_i$

  $\hat{\mu}_{i,b} := \hat{\mu}_{i,f} := \mu_i$

end

while $K^{c-1}_{total} \neq K^{c-2}_{total}$ *and* $c \leq 10$ **do**

  for $i \in (2,3,4)$ **do**

    $\hat{\mu}_{i,f} := \mu_i + d_{i-1}$

    if $i \neq 4$ **then**

      $K^i, K_{i+1} :=$
      $\arg\min_{0 \leq K^i \leq n} \arg\min_{0 \leq K_{i+1} \leq n} f_{p,i}(K^i, K_{i+1}, \lambda_i, \hat{\lambda}_{i+1,b}, \hat{\mu}_{i,f}, \hat{\mu}_{i+1,b})$

      $d_i := \mu_i \cdot \sum_{j=0}^{K^i} \sum_{k=K_{i+1}+1}^{n} \pi^i(j,k)$

    else

      $\mu_{D1}, \mu_{D2} :=$
      $\arg\min_{0 \leq \mu_{D1} \leq \hat{\mu}_{4,f}} \arg\min_{0 \leq \mu_{D2} \leq \hat{\mu}_4} f_{p,4}(K^4, K_5, \lambda_4, \lambda_5, \hat{\mu}_{4,f}, \mu_{D1}, \mu_{D2})$

      $K^4, K_5 := \arg\min_{0 \leq K^4 \leq n} \arg\min_{0 \leq K_5 \leq n} f_{p,4}(K^4, K_5, \lambda_4, \lambda_5, \hat{\mu}_{4,f}, \mu_{D1}, \mu_{D2})$

    end

  end

  for $i \in (3,2,1)$ **do**

    if *i=3* **then**

      $\hat{\mu}_{i+1,b} :=$ Equation 41

      $\hat{\lambda}_{i+1,b} := \lambda_{i+1}$

    else if $\mathbb{P}(i+1, i+2) \geq 1 - \frac{\lambda_{i+1}}{\mu_{i+1}}$ **then**

      $\hat{\mu}_{i+1,b} := \mu_{i+1} - d_{i+1}$

      $\hat{\lambda}_{i+1,b} := \lambda_{i+1}$

    else

      $\hat{\lambda}_{i+1,b} := \lambda_{i+1} + d_{i+1}$

      $\hat{\mu}_{i+1,b} := \mu_{i+1}$

    end

    $K^i, K_{i+1} :=$
    $\arg\min_{0 \leq K^i \leq n} \arg\min_{0 \leq K_{i+1} \leq n} f_{p,i}(K^i, K_{i+1}, \lambda_i, \hat{\lambda}_{i+1,b}, \hat{\mu}_{i,f}, \hat{\mu}_{i+1,b})$

    $d_i := \mu_i \cdot \sum_{j=0}^{K^i} \sum_{k=K_{i+1}+1}^{n} \pi^i(j,k)$

    $\mathbb{P}(i, i+1) := \sum_{j=0}^{K^i} \sum_{k=K_{i+1}+1}^{n} \pi^i(j,k)$

  end

  $K^c_{tot} := (K^1, K_2, K^2, K_3, K^3, K_4, K^4, K_5)$

  $c := c + 1$

end

Algorithm 2: Combining the subsystems with maximal expected queue length

Result: $\mathrm{K}_{total} = (\mathrm{K}^1, K_2, K^2, K_3, K^3, K_4, K^4, K_5)$, μ_{D1}, μ_{D2}

Initialization:

$K^4, K_5 = \lfloor 0.5n \rfloor, \lfloor 0.5n \rfloor$

$K^1, K_2 := \underset{0 \geq K^1 \geq n}{\arg\min} \underset{0 \geq K_2 \geq n}{\arg\min} f_{p,1}(K^1, K_2, \lambda_1, \lambda_2, \mu_1, \mu_2)$

$d_1 := \mu_1 \cdot \sum_{j=0}^{K^1} \sum_{k=K_2+1}^{n} \pi^1(j,k)$

$c := 1$

for $i \in (2,3,4)$ **do**

  $\hat{\lambda}_{i,b} := \hat{\lambda}_{i,f} := \lambda_i$

  $\hat{\mu}_{i,b} := \hat{\mu}_{i,f} := \mu_i$

end

while $K_{total}^{c-1} \neq K_{total}^{c-2}$ *and* $c \leq 10$ **do**

  for $i \in (2,3,4)$ **do**

    $\hat{\lambda}_{i,f} := \lambda_i - d_{i-1}$

    if $i \neq 4$ **then**

      $K^i, K_{i+1} :=$
      $\underset{0 \leq K^i \leq n}{\arg\min} \underset{0 \leq K_{i+1} \leq n}{\arg\min} f_{p,i}(K^i, K_{i+1}, \hat{\lambda}_{i,f}, \hat{\lambda}_{i+1,b}, \mu_i, \hat{\mu}_{i+1,b})$

      $\mathrm{d}_i := \mu_i \cdot \sum_{j=0}^{K^i} \sum_{k=K_{i+1}+1}^{n} \pi^i(j,k)$

    else

      $\mu_{D1}, \mu_{D2} :=$
      $\underset{0 \leq \mu_{D1} \leq \mu_4}{\arg\min} \underset{0 \leq \mu_{D2} \leq \mu_4}{\arg\min} f_{p,4}(K^4, K_5, \hat{\lambda}_{4,f}, \lambda_5, \mu_4, \mu_{D1}, \mu_{D2})$

      $\mathrm{K}^4, K_5 := \underset{0 \leq K^4 \leq n}{\arg\min} \underset{0 \leq K_5 \leq n}{\arg\min} f_{p,4}(K^4, K_5, \hat{\lambda}_{4,f}, \lambda_5, \mu_4, \mu_{D1}, \mu_{D2})$

    end

  end

  for $i \in (3,2,1)$ **do**

    if *i=3* **then**

      $\hat{\mu}_{i+1,b} :=$ Equation 41

      $\hat{\lambda}_{i+1,b} := \lambda_{i+1}$

    else if $\mathbb{P}(i+1, i+2) < 1 - \frac{\lambda_{i+1}}{\mu_{i+1}}$ **then**

      $\hat{\mu}_{i+1,b} := \mu_{i+1} - d_{i+1}$

      $\hat{\lambda}_{i+1,b} := \lambda_{i+1}$

    else

      $\hat{\lambda}_{i+1,b} := \lambda_{i+1} + d_{i+1}$

      $\hat{\mu}_{i+1,b} := \mu_{i+1}$

    end

    $K^i, K_{i+1} :=$
    $\underset{0 \leq K^i \leq n}{\arg\min} \underset{0 \leq K_{i+1} \leq n}{\arg\min} f_{p,i}(K^i, K_{i+1}, \hat{\lambda}_{i,f}, \hat{\lambda}_{i+1,b}, \mu_i, \hat{\mu}_{i+1,b})$

    $d_i := \mu_i \cdot \sum_{j=0}^{K^i} \sum_{k=K_{i+1}+1}^{n} \pi^i(j,k)$

    $\mathbb{P}(i, i+1) := \sum_{j=0}^{K^i} \sum_{k=K_{i+1}+1}^{n} \pi^i(j,k)$

  end

  $K_{tot}^c := (K^1, K_2, K^2, K_3, K^3, K_4, K^4, K_5)$

  $c := c + 1$

end

λ_i	μ_i	λ_{i+1}	μ_{i+1}	$\mathbb{P}(i, i+1)$	$\mathbb{P}(i, i+1) \cdot \mu_i$
2	7	5	6	0.081	0.566
2	8	5	6	0.065	0.519
3	7	5	6	0.026	0.185
3	8	5	6	0.037	0.293
5	9	5	6	0.0135	0.1216
6	9	5	6	0.0118	0.1058
7	9	5	6	0.0120	0.1080

Table 3: Values of $\mathbb{P}(i, i+1)$ and $\mathbb{P}(i, i+1) \cdot \mu_i$ for various inputs.

5.5.5 Convergence of Algorithms

To prove convergence of the algorithms, we need to prove that either $\mathbb{P}(i, i+1)$ or $\mathbb{P}(i, i+1) \cdot \mu_i$ converges, since each iteration these values are updated. Based on these values the value of either $\lambda_i, \mu_i, \lambda_{i+1}$ or μ_{i+1} is updated, which is then again used to update the value of $\mathbb{P}(i, i+1)$.

According to a numerical analysis, which is given in Table 3, for a fixed n and $N_{i,threshold}$ $\forall$ i there is no correlation between either μ_i or λ_i in system i and either $\mathbb{P}(i, i+1)$ or $\mathbb{P}(i, i+1) \cdot \mu_i$. Therefore, convergence of either of those is impossible to prove.

In practice, Algorithm 2 did not converge for one of the instances, but kept going back and forth between two solutions. For that instance, Algorithm 1 did converge. Since the two solutions obtained with Algorithm 2 gave similar expected penalties and these two solutions were found in the first three iterations, we have chosen to define a maximum number of iterations $c = 10$, to ensure termination of the algorithms. As both solutions found in this instance are similar, we expect that this will still give a reasonable good solution.

6 Results

This chapter firstly discusses how well the policy approximation behaves in comparison to the MDP approach. Then the the results of a simulation of a policy similar to the current method versus the policy approximation method are given.

6.1 Small-scale: MDP vs Policy Approximation

We cannot solve the MDP in realistic settings, but we do want to give an indication of how well the approximation is doing. Therefore, we compare them in a small-scale example.

In this small scale example, we still have all five types of clients and four types of practitioners. Each type of practitioner has one slot. The maximum queue length is set to three for each type. Because the maximum queue length is quite small and expected service time cannot be less than 1 using the geometric distribution, the arrival rate is also set small, $\lambda_i = 0.5$ for $i = 1, 2, 3$ and $\lambda_i = 0.25$ for $i = 4, 5$. We do keep a similar ρ to real life, by setting $\mu_i = 0.575$ $\forall i$. This means that the average treatment duration is $\frac{1}{0.575} = 1.74$ treatments, with a probability of $p = 0.575$ of needing no more treatments. All values for $N_{i,threshold}$ are set to zero, so any client in queue will give a penalty. $\delta = 0.99$, is set quite high, since we do value clients in the future quite high.

The MDP as described in Section 5.3 and 5.4 is a discounted MDP. We maximize the expected total discounted reward, as defined in Section 5.3.5. However, since we do not care in which state we start, we take the average:

$$\mathbb{E}[v_\lambda] = \frac{1}{|S|} \sum_{s \in S} v_\lambda^\pi(s). \tag{42}$$

We solve the MDP using linear programming, as described in Section 3.2 and obtain $\mathbb{E}[v_\lambda^{MDP}]$. We then compare this with $\mathbb{E}[v_\lambda^{alg,i}]$, the value corresponding to the policy found with Algorithm $i = 1, 2$ in Table 4. Furthermore, we compare the total expected queue length $\mathbb{E}[L_{q,total}]$ and the expected queue length of each type of client $\mathbb{E}[L_{q,i}]$, using the steady state distribution. We also compare the runtime. We define the runtime to be the time between the initialisation of all states and the basic solution, meaning that for each state an action can be determined within seconds.

Since the algorithms unexpectedly assign almost all capacity (99%) of SPV to either clients of type D or type E, these solutions are compared with the solutions in which the capacity is divided in proportion to the arrival rates, which is also given in Table 4.

Note that for both approximations, both algorithms gave the same thresholds, the differences in expected penalty and queue length are due to randomly choosing clients of type D or E for SPV, with probabilities in proportion with the service rate, making a slightly different transition matrix P each time.

		Optimising μ_{D1} and μ_{D2}		Capacity divided pro rata	
	MDP	Algorithm 1	Algorithm 2	Algorithm 1	Algorithm 2
$\mathbb{E}[v_\lambda]$	690.1183	873.6636	873.6914	904.4529	903.8921
$\mathbb{E}[L_{q,total}]$	6.8451	8.8537	8.8540	9.2200	9.2134
$\mathbb{E}[L_{q,A}]$	2.9251	2.0476	2.0476	2.0477	2.0478
$\mathbb{E}[L_{q,B}]$	1.4889	1.9625	1.9625	1.9638	1.9636
$\mathbb{E}[L_{q,C}]$	1.2221	1.9039	1.9039	1.9108	1.9105
$\mathbb{E}[L_{q,D}]$	0.3796	0.8322	0.8325	1.2138	1.2080
$\mathbb{E}[L_{q,E}]$	0.8294	2.1076	2.1076	2.0838	2.0834
Runtime	131280 s	167 s	167 s	1.7 s	1.7 s

Table 4: Small-scale comparison between the runtime and solutions given by the MDP and Algorithm 1 and 2.

Max queue length $n =$		10	15	20	25
Optimising μ_{D1} and μ_{D2}	Algorithm 1	600 s	1259 s	6216 s	16973 s
	Algorithm 2	600 s	1255 s	6171 s	17000 s
Capacity divided pro rata	Algorithm 1	47.8 s	207 s	922 s	3470 s
	Algorithm 2	44.5	207 s	914 s	3277 s

Table 5: Comparison of the average runtime of Algorithm 1 and 2.

6.2 Large Scale: Approximation vs Current Policy

For realistically sized problems, we compare the runtime for varying maximal queue lengths n for Algorithm 1 and 2 in Table 5, both in- and excluding the optimisation of μ_{D1} and μ_{D2}. As inputs, we use realistic values. Since the average care path takes 15.45 hours, as seen in Table 2, we set $\mu_i = \frac{1}{15.45} \cdot y_i$, where y_i is the number of slots all practitioners of type i currently have together per week according to the production standard. We do not know the arrival rate per type of client, but we do know the total arrival rate, which is 20 new care paths per week. We divide this over the types such that $\lambda_i = \frac{20 \cdot y_i}{\sum_{j=1}^4 y_j}$ for $i = 1, 2, 3$. For type D and E, we divide the arrivals equally over both types, such that $\lambda_D = \lambda_E = 0.5 \frac{20 \cdot y_4}{\sum_{j=1}^4 y_j}$. Furthermore, we set $N_{i,penalty}$ to 1.5 times λ_i for all i. It is remarkable that for all different maximum queue lengths n, the same thresholds are found using both Algorithm 1 and Algorithm 2, and therefore the expected queue lengths are also the same.

In these cases, the state space has grown too large to obtain the steady state analytically, therefore a simulation is done with the found thresholds and the policy as defined in Section 5.5.4.
Using the above mentioned settings and the thresholds found with $n = 25$, a simulation is done over 100,000 weeks, of which 10,000 are used as a warm-up period. In this simulation, the arrivals are generated by a Poisson distribution.

	$L_{q,A}$	$L_{q,B}$	$L_{q,C}$	$L_{q,D}$	$L_{q,E}$	$L_{q,total}$
Approximation	3.22	6.59	5.52	3.10	3.14	21.57
Approximation pro rata	3.22	6.59	5.52	3.10	3.14	21.57
Current policy	3.08	6.71	5.87	3.10	3.09	21.86

Table 6: Comparison of the average queue lengths of the approximation and a policy similar to the current policy.

	$L_{q,A}$	$L_{q,B}$	$L_{q,C}$	$L_{q,D}$	$L_{q,E}$	$L_{q,total}$
Approximation	34.10	111.81	45.72	4.65	38.74	235.04
Approximation pro rata	34.11	112.61	48.65	15.52	23.87	234.75
Current policy	28.72	209.34	50.53	3.67	3.66	295.92

Table 7: Comparison of the average queue lengths of the approximation and a policy similar to the current policy under high load.

Each week, for each client in a slot the probability of leaving is given by $\frac{1}{15.45}$, creating a geometric distribution. The average queue lengths found by this simulation are given in Table 6, both for an division of SPV capacity pro rata and for the division as given by the optimisation algorithm.
Furthermore, the average queue lengths of the same simulation with a similar policy to the current policy are given. The policy resembling the current policy is defined as KP only treating clients of type A, GZ only type B, BP only type C and SPV treating clients of type D or E proportional to their arrival rate if both are in queue. This is assumed that clients are planned back to back.

Since the queue is at the moment very long, we assess the approximation also on a higher load, by increasing the arrival rates to 99% of the service rates. The same method as before is used. The values of the average queue lengths can be found in Table 7.

7 Discussion

The model relies on the blueprint schedule and short-term planning, which is new for this department of Mediant. Changes to the blueprint, for example when a practitioner has different working hours or different groups, still need a lot of manual scheduling, since clients receiving treatment from that practitioner need to be rescheduled into the new blueprint. The blueprint is expected to take some getting used to, but also to give the practitioners more peace of mind, since every two weeks are similar and less changes are made last-minute. Furthermore, the short-term planning is expected to prevent a lot of gaps in the schedule, thus improving efficiency.
The policy found by the approximation assigns types of clients to types of practitioners, but does not yet assign a specific client to a specific practitioner. There are many subtle differences between practitioners, making this a challenge to adequately assign clients to practitioners.

The MDP seems to be a good model, but is still an approximation, as geometric service times and exponential inter-arrival times are assumed. Therefore the policies as found by the algorithms are an approximation of an approximation. In the small instance, we see that using the MDP the expected queue length for clients of type A, as depicted in Table 4, is very near the maximal allowed number of clients in the queue, keeping the penalty near the maximum for that queue. If the maximum allowed number of clients in the queue would be larger, the penalty would grow larger and other choices might be made.

Since there is found an instance for which Algorithm 2 does not converge, it is likely that Algorithm 1 does not always converge. Therefore the number of iterations have to be limited and we might not see the best solution. However, the solutions found are deemed good enough approximations. If converged, both algorithms give the same values for the thresholds, indicating that this is quite a robust solution, but making it impossible to determine which of the algorithms perform better.
The algorithms both assign almost all capacity of SPV to either clients of type D or clients of type E when $n_D \leq K^D$ and $n_E \leq K_E$ or $n_D > K^D$ and $n_E > K_E$. However, the differences between the penalties and the expected queue lengths of the optimal solution and of the solution in which the capacity is divided proportional to the arrival rate are small. Therefore it would be better to divide the capacity proportional to the arrival rate, since in the case of long queues, still both types of clients are treated and the queue lengths are more balanced.
The approximation performs better under high load compared to a similar policy to the current policy than under light load. This is reasonable, since under high load it is more often profitable to assign clients to another practitioner than the default practitioner. Furthermore, in the policy the approximation is compared with, short term planning is assumed, which is not the case. Else the expected queues for the current policy would be even larger.

8 Conclusion and Recommendations

Clients can be scheduled back to back, avoiding gaps between different clients and difficulties to plan extra appointments when needed. This is expected to increase the throughput. A blueprint schedule is expected to give some regularity and with that some peace of mind for practitioners, since it is less changeable than the current schedules. Furthermore, it makes it easier to plan clients back to back and to see how much of the capacity is used.
An MDP is developed to schedule types of clients to types of practitioners. Since this is not solveable on large scale, an approximation based on both a threshold and a trunk reservation policy is developed. The approximation gives an expected discounted penalty of 27% higher than the penalty found with MDP, and an expected total queue length of 29% higher than the expected total queue length found with MDP. It is striking that using the approximation the expected queue length of type A is significantly shorter than with the MDP solution, and for type E vice versa.
The approximation gives a better expected queue length than planning via the default assignment of clients to types of practitioners, assuming back-to-back planning, especially under high load. This means that, with both introducing back-to-back planning and using the approximation for assigning types of clients to types of practitioners, the queues are expected to decrease significantly.

Further research could be directed to making a model for assigning a specific client to a specific practitioner. Another interesting research direction is to incorporate the difference in service time between a BGGZ client and an SGGZ client in the model. Furthermore, the current model could be extended such that there are clients that are to be seen weekly and clients that are to be seen biweekly.
The results of these models should be such that they can be applied by the planners in a structured manner.

References

Bikker, I. A., Kortbeek, N., van Os, R. M., & Boucherie, R. J. (2015). Reducing access times for radiation treatment by aligning the doctor's schemes. *Operations research for health care*, *7*, 111–121.

Carey, T. A., Tai, S. J., & Stiles, W. B. (2013). Effective and efficient: Using patient-led appointment scheduling in routine mental health practice in remote australia. *Professional Psychology: Research and Practice*, *44*(6), 405.

Feinberg, E. A., & Yang, F. (2011). Optimality of trunk reservation for an m/m/k/n queue with several customer types and holding costs. *Probability in the Engineering and Informational Sciences*, *25*(4), 537.

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

Koizumi, N. (2002). *A queueing network model with blocking: Analysis of congested patients flows in mental health systems*. University of Pennsylvania.

Leeftink, A., Bikker, I., Vliegen, I., & Boucherie, R. (2020). Multi-disciplinary planning in health care: A review. *Health Systems*, *9*(2), 95–118.

Lin, W., & Kumar, P. (1984). Optimal control of a queueing system with two heterogeneous servers. *IEEE Transactions on Automatic control*, *29*(8), 696–703.

Luh, H. P., & Viniotis, I. (2002). Threshold control policies for heterogeneous server systems. *Mathematical Methods of Operations Research*, *55*(1), 121–142.

Maddah, B., & El-Taha, M. (2016). Selective trunk with multiserver reservation. *Advances in Operations Research*.

Murch, B., Cooper, J., Hodgett, T., Gara, E., Walker, J., & Wood, R. (2021). Modelling the effect of first-wave covid-19 on mental health services. *Operations Research for Health Care*, 100311.

Pagel, C., Richards, D. A., & Utley, M. (2012). A mathematical modelling approach for systems where the servers are almost always busy. *Computational and mathematical methods in medicine*, *2012*.

Powell, W. B. (2019). *Reinforcement learning and stochastic optimization and learning: A unified framework*. Department of Operations Research; Financial Engineering, Princeton University.

Puterman, M. L. (2014). *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons.

Redko, C., Rapp, R. C., & Carlson, R. G. (2006). Waiting time as a barrier to treatment entry: Perceptions of substance users. *Journal of Drug Issues*, *36*(4), 831–852.

Reichert, A., & Jacobs, R. (2018). The impact of waiting time on patient outcomes: Evidence from early intervention in psychosis services in england. *Health Economics*, *27*(11), 1772–1787. https://doi.org/https://doi.org/10.1002/hec.3800

Van Leeuwaarden, J., & Winands, E. (2006). Quasi-birth-and-death processes with an explicit rate matrix. *Stochastic models*, *22*(1), 77–98.

Viniotis, I., & Ephremides, A. (1988). Extension of the optimality of the threshold policy in heterogeneous multiserver queueing systems. *IEEE Transactions on Automatic Control*, *33*(1), 104–109. https://doi.org/10.1109/9.371

Westin, A. M., Barksdale, C. L., & Stephan, S. H. (2014). The effect of waiting time on youth engagement to evidence based treatments. *Community mental health journal*, *50*(2), 221–228.

Zorgautoriteit, N. (2017). Voortgangsrapportage: Wachttijden in de ggz. *NVGzP*.

Zorgautoriteit, N. (2018). Eindrapportage landelijke afspraken: Wachttijden in de ggz. *NVGzP*.