BACHELOR

Appointment scheduling

van Egmond, Hannah

Award date:
2023

Appointment scheduling

H.M. van Egmond
1473190

Supervisor:
M.A.A. Boon.
Committee member:
C.A.J. Hurkens.

Department of Mathematics and Computer Science
Eindhoven University of Technology
The Netherlands
February 6, 2023

Abstract

In this Bachelor Final Project we study appointment scheduling both in the single server and the multiserver setting. A static, a dynamic and an adaptive approach for the single server setting are discussed and results of some experiments are presented. In the chapter about the static case it is shown how appointment scheduling can be used in practice. For the dynamic approach the same results as in [1] are found. For the adaptive approach we managed to derive a general expression for the expected sojourn time of the first client conditional on the already elapsed service time. Moreover a multiserver setting is discussed. The implementation of the discussed method seems to be extremely slow, therefore the corresponding experiments are only executed for a few clients. Lastly no-shows and walk-ins are included in multiple models, such as the static phase-type, static homogeneous exponential and the multiserver case. Furthermore no-shows have been included in the model for dynamic scheduling for homogeneous exponentially distributed service times, which has not been done before. There seem to be two general methods for including no-shows. Either a probability q of having a no-show is included in the model such that a client has, with probability q, the service time that he would have if no-shows were not included and with probability $1-q$ a service time equal to 0. Or, what only works in the phase-type case, an adapted squared coefficient of variation is calculated and used to schedule all clients.

Contents

1 Introduction

There are many service systems where appointment scheduling plays a very important role, such as patients visiting the doctor, patients visiting the dentist or people at home waiting for their package to be delivered. Another example is the arrival of ships to the harbor in a seaport [2], due to limited space, ships are scheduled to arrive at the seaport at specific times. Appointment scheduling is a subject that is widely studied in mathematics. When the service times of the patient or ship at the harbor are known, it is a problem that is solved by using combinatorial optimization [3]. In the case that the service times are not known, the case which will be explored in this Bachelor Final Project, the problem is solved using the theory of queuing systems [4].

When searching for an optimal schedule, the main objective is to properly weigh the interests of the service provider and its clients. The server wants the service system to be efficient and the clients want a sufficiently high level of service. Mostly, these two quantities are phrased in terms of the server's idle time and the individual client's waiting time respectively [1]. In order to generate an optimal schedule, one has the task to define the client's arrival times that minimize the cost function that comprises the weighted expected idle time of the server and the expected waiting times of the clients.

Most studies in the literature focus primarily on static schedules. This means that the client's arrival times are determined a priori and are not updated during the makespan. However, as proven in multiple papers, scheduling static is not the best method. Since for example gains can be achieved if the schedule is updated [5]. If the server is ahead of schedule, one could let the next clients arrive earlier, such that the increase of the cost function due to idle time of the server can be limited. In the situation that one is behind schedule, one could let the next clients arrive later, such that the increase of the cost function due to waiting time of the clients can be limited.

Moreover, most studies in literature consider the single server setting, despite the fact that this is not always the most realistic one, as there are settings where a client can be served by different doctors, for example in first aid. Additionally, most studies assume that all patients arrive, however, also this seems to be unrealistic for many situations [6].

As discussed in [7] the problem why most of the literature about the appointment scheduling is not yet applied in practice is that it discusses a too simplified setting of the reality. Therefore we consider in this Bachelor Final Project not only different scheduling methods for the single server setting, but also a multiserver setting and we study how no-shows and walk-ins can be dealt with. Descriptions of the mathematical models and the results of experiments performed with those are presented. All experiments are written in *Wolfram Mathematica 13.1* on a Lenovo Thinkpad P1.

With this Bachelor Final Project we contribute to the already existing literature since we discuss a lot of different models in one report. We have not found literature that, as we do, discusses both single server and multiserver models. Moreover we show in Subsection 5.1.3, by means of some experiments, how appointment scheduling can be used in practice. Furthermore we have derived a general expression for the expected remaining service time in the adaptive single server model, provided a worked out example of the multiserver procedure and discussed multiple methods of including no-shows in one report.

This Bachelor Final Project is organized as follows. First in Section 2 some applicable theory is discussed. Next, in Section 3 the relevant results from existing literature are discussed. In Section 4 the mathematical model is described. Section 5 discusses different single server scheduling methods and per method the results of the experiments belonging to it. Moreover, in Section 6 a model for the multiserver schedule is discussed. Different methods of including no-shows and walk-ins are described in Section 7. We conclude this Bachelor Final Project in Section 8, in which some conclusions from the results and ideas for further research are described.

Since most of the literature studied for this Bachelor Final Project is about the healthcare setting, also this terminology will be used. In other words, the server can be referred to as the doctor, the clients can be referred to as patients and the service times can be referred to as their treatment durations.

2 Applicable theory

To make sure that the reader has a basic understanding of the concepts that are used in this Bachelor Final Project some applicable theory is described in this section. In particular, the matrix exponential, the continuous phase-type distribution, the concept of fitting a phase-type distribution and the Kronecker product and sum are described.

2.1 Matrix exponential

The scalar exponential function $\exp(x)$ can be represented by the power series

$$e^x = 1 + x + \frac{x^2}{2!} + ... = \sum_{n=0}^{\infty} \frac{x^n}{n!},$$

in which x is a number. This definition can be extended for the exponential function for $n \times n$ matrices. So the matrix exponential denoted by $\exp(\boldsymbol{A})$, with $\boldsymbol{A}$ being a $n \times n$ constant matrix equals

$$e^{\boldsymbol{A}} = \boldsymbol{I} + \boldsymbol{A} + \frac{\boldsymbol{A}^2}{2!} + ... = \sum_{k=0}^{\infty} \frac{\boldsymbol{A}^k}{k!},$$

in which $\boldsymbol{I}$ is the $n \times n$ identity matrix [8]. As can be observed, we indicate vectors and matrices in **bold**.

2.2 Continuous phase-type distribution

A phase-type distribution is the distribution of the time to absorption in a finite Markov chain of dimension $m+1$. Let $\{X(t)\}_{t\geqslant 0}$ be a Markov jump process on the finite state space $E = \{1, 2, ..., m, m+1\}$, where the states $1, ..., m$ are transient states and state $m+1$ is absorbing. Then $\{X(t)\}_{t\geqslant 0}$ has an intensity matrix of the form

$$Q = \begin{pmatrix} \boldsymbol{T} & \boldsymbol{t} \\ \boldsymbol{0}_{1\times m} & 0 \end{pmatrix}.$$

Here $\boldsymbol{T}$ is a $m \times m$ dimensional matrix, $\boldsymbol{t}$ is a m dimensional column vector and $\boldsymbol{0}$ is the m dimensional row vector of only zeros. Since the intensities of rows must sum to zero, we notice that $\boldsymbol{t} = -\boldsymbol{T}\boldsymbol{1}$. The intensities $\boldsymbol{t}_i$ are the intensities by which the process jumps to the absorbing state and are referred to as exit rates. The $\boldsymbol{T}_{i,j}$ denote the intensities by which the process jumps from transient state i to transient state j. Lastly, let us define $\boldsymbol{\alpha}$ as the initial probability which is m dimensional. Since it is a probability vector, all its entries are non-negative and they sum up to 1. Now the pair $(\boldsymbol{\alpha}, \boldsymbol{T})$ is called the representation for the phase-type distribution [1, 9].

Next, consider the probabilities $p_i(t)$, with $i = 1, ..., m$, denoting the probability of the Markov jump process being in transient state i at time t. These probabilities are collected in the vector $\boldsymbol{p}(t)$, solving the system of differential equations

$$\boldsymbol{p'}(t) = \boldsymbol{p}(t)\boldsymbol{T},$$

which satisfies the initial condition $\boldsymbol{p}(0) = \boldsymbol{\alpha}$. This system has the solution $\boldsymbol{p}(t) = \boldsymbol{\alpha} e^{(t\boldsymbol{T})}$. From this it follows that the probability that the jump process is not yet absorbed at time t is $p(t)\boldsymbol{1} = \alpha e^{(t\boldsymbol{T})}\boldsymbol{1} = P(X > t)$. From this the cumulative distribution function and probability density function can be deduced. The cumulative distribution function is:

$$P(X \leqslant t) = F(t) = 1 - \boldsymbol{\alpha} e^{(t\boldsymbol{T})}\boldsymbol{1}.$$

By using $e^{t\boldsymbol{T}} = \sum_{i=0}^{\infty} \frac{(t\boldsymbol{T})^i}{i!}$ and $\boldsymbol{T}\boldsymbol{1} + \boldsymbol{t} = \boldsymbol{0}$ we find the probability density function:

$$\begin{aligned} f(t) &= F'(t) \\ &= -\boldsymbol{\alpha} e^{(t\boldsymbol{T})}\boldsymbol{T}\boldsymbol{1} \\ &= \boldsymbol{\alpha} e^{(t\boldsymbol{T})}\boldsymbol{t}. \end{aligned}$$

Lastly the expected value of a phase-type distributed random variable is given by $\boldsymbol{\alpha}(-\boldsymbol{T})^{-1}\boldsymbol{1}$ [9].

2.3 Fitting a phase-type distribution

Fitting a phase-type distribution will be used in multiple models that we study. Phase-type distributions are useful since their densities and distribution function can be calculated easily, as shown in Subsection 2.2. Moreover, any positive distribution can be approximated closely by a phase-type distribution [5]. As is common in literature, the squared coefficient of variation is used for phase-type fitting. The SCV, the squared coefficient of variation, is defined as the variance divided by the square of the mean,

$$\mathrm{SCV}(B_i) = \frac{\sigma(B_i)^2}{\mu(B_i)^2} = \frac{\mathrm{Var}(B_i)}{\mu(B_i)^2}.$$

Here B_i is a non-negative random variable. We continue by making a case distinction for the SCV being smaller than, equal to or larger than 1. We briefly discuss how to map the mean and SCV of a random variable B on the corresponding parameters. In Appendix A a more in-depth description of how these parameters are determined is provided. If the SCV equals one, both fits that you would get for SCV larger than 1 or smaller than 1 would also work, however, by doing so the parameters become unnecessarily difficult [1, 5, 10].

Case 1: SCV equals 1. This is the easiest case. If the SCV equals 1, the standard deviation and the mean are equal. This means that an exponential distributions fits well.

Case 2: SCV smaller than 1 If the SCV is smaller than 1, we approximate B by a mixture of two Erlang distributions, denoted by

$$B \sim E(K, \mu)\mathbb{1}_{\{U<p\}} + E(K+1, \mu)\mathbb{1}_{\{U>p\}}.$$

for some $K \in \mathbb{N}$, $\mu > 0$ and $p \in [0, 1]$. Here $E(K, \mu)$ represents an Erlang distributed random variable with parameters and μ, and U denotes an independent uniform random variable on [0,1]. So with probability p the random variable B equals an Erlang-distributed random variable with K phases, and with probability $1 - p$ an Erlang-distributed random variable with $K + 1$ phases. The parameters are determined as:

$$K = \left\lfloor \frac{1}{S(B)} \right\rfloor, \; p = \frac{(K+1)S(B) - \sqrt{(K+1)(1 - K \cdot S(B))}}{S(B) + 1}, \; \mu = \frac{K + 1 - p}{\mathbb{E}[B]}.$$

The corresponding transition matrix $\boldsymbol{T} \in \mathbb{R}^{(K+1)\times(K+1)}$ is

$$\begin{pmatrix} -\mu & \mu & 0 & \cdot & \cdot & 0 \\ 0 & -\mu & \mu & & & \cdot \\ \cdot & & \cdot & \cdot & & \cdot \\ \cdot & & & \cdot & \mu & 0 \\ \cdot & & & & -\mu & \mu q \\ 0 & \cdot & \cdot & \cdot & 0 & -\mu \end{pmatrix},$$

with $q = 1-p$ and initial probability distribution $\boldsymbol{\gamma} = (1, 0, ..., 0)$. The corresponding phase diagram can be found in Figure 1

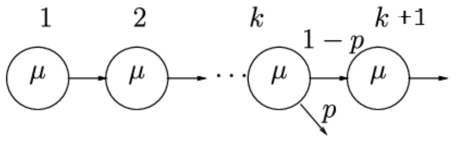


Figure 1: Phase diagram mixed Erlang. Figure courtesy of the authors of [11].

Case 3: SCV larger than 1. If the SCV is larger than 1, we approximate B by a hyperexponential distribution. For some $\mu_1, \mu_2 > 0$ and $p \in [0, 1]$,

$$B \sim \exp(\mu_1)\mathbb{1}_{\{U<p\}} + \exp(\mu_2)\mathbb{1}_{\{U>p\}},$$

So B equals with probability p an exponentially distributed random variable with mean μ_1^{-1} and with probability $1-p$ an exponentially distributed random variable with mean μ_2^{-1}. We use the, so called, *balanced means* condition to get μ_1 and μ_2, i.e. $\mu_1 = 2p\mu$ and $\mu_2 = 2(1-p)\mu$ for some $\mu > 0$. Using this, we find

$$p = \frac{1}{2}\left(1 + \sqrt{\frac{\text{SCV}-1}{\text{SCV}+1}}\right), \ \mu_1 = \frac{2p}{\mathbb{E}[B]}, \ \mu_2 = \frac{2(1-p)}{\mathbb{E}[B]}.$$

The corresponding transition matrix $\boldsymbol{T} \in \mathbb{R}^{2\times 2}$ is $\begin{pmatrix} -\mu_1 & 0 \\ 0 & -\mu_2 \end{pmatrix}$ and $\boldsymbol{\gamma} = \begin{pmatrix} p, & 1-p \end{pmatrix}$. The corresponding phase diagram can be found in Figure 2

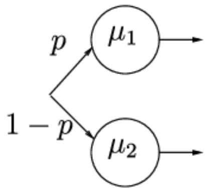


Figure 2: Phase diagram hyperexponential. Figure courtesy of the authors of [11].

2.4 Kronecker product and sum

For two matrices $\mathbf{A}$ with dimension $l \times k$ and $\mathbf{B}$ with dimension $n \times m$ the Kronecker product $\otimes$ is defined as

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \dots & a_{1k}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \dots & a_{2k}\mathbf{B} \\ . & . & \dots & . \\ . & . & \dots & . \\ . & . & \dots & . \\ a_{l1}\mathbf{B} & a_{l2}\mathbf{B} & \dots & a_{lk}\mathbf{B} \end{pmatrix}$$

and the Kronecker sum is defined as $\mathbf{A} \oplus \mathbf{B} = \mathbf{A} \otimes I_n + I_l \otimes \mathbf{B}$ [12]. For the matrices $\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$ and $\begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$ the Kronecker sum equals

$$\begin{pmatrix} a_{11}+b_{12} & b_{11} & a_{12} & 0 \\ b_{21} & a_{11}+b_{22} & 0 & a_{12} \\ a_{21} & 0 & a_{22}+b_{11} & b_{12} \\ 0 & a_{21} & b_{21} & a_{22}+b_{22} \end{pmatrix}.$$

3 Literature overview

In this section a brief literature overview will be provided. A more extensive literature overview can be found in *Outpatient scheduling in health care: A review of literature* and *Outpatient appointment systems in healthcare: A review of optimization studies* [7, 13]. Many papers have appeared in the literature on appointment scheduling, most of them are motivated by healthcare applications.

The vast majority of the literature considers the single server setting. Moreover, the dominant approach in literature is the one that considers static scheduling. This means that clients' arrival times are determined before the start of the makespan and are not updated meanwhile the server is serving clients [14, 15, 16, 17, 18]. [16] is known as the first formulation of the single server environment, by Bailey and Welch. Those authors are known by the Bailey-Welch appointment rule. They introduced the method in which the makespan is divided in blocks with lengths equal to the average service time. The first block is booked by at least two patients and the subsequent blocks are assigned to only one patient, to prevent possible idle times in early stages of the schedule. By simulation it is found that this method works surprisingly well. In [19] Jo and Lau study variants on this appointment rule and it is concluded that the three most important factors on the performance of an appointment schedule are the number of clients to be scheduled, the service-time variability and no-shows. More recent literature has considered the problem, without the constraint that appointed arrivals need to be equidistant, see for example [15, 5, 1, 10]. Moreover, the problem is made tractable by applying approximations or simulations [15, 18]. In 2015 in [20] the so called *Lag order approximation method* is presented, that optimizes the arrival times of clients based on only the last K of his predecessors. Since this method takes less variables into account, the computation times are lower. In most papers the arrival times are chosen such that the sum of the waiting time of the clients and the idle time of the server are minimized, this is called the simultaneous approach. Therefore we want to mention [15], in which both this and the sequential approach are studied, the approach in which the arrival times of the next client is scheduled given the arrival epochs of all previous clients.

At first exponential service times were assumed. For example Pedgen and Rosenshine describe in [14] a method to determine an optimal schedule for n customers, assuming that the customers have homogeneous exponentially distributed service times. Later on, in multiple papers it is concluded that in practice the service times are not likely to be exponentially distributed. Therefore phase-type distributions are often used, as it is known that they approximate positive distributions closely [14, 15, 18]. Another reason for using the phase-type distribution is that by doing so, tractability is obtained. This phase-type distribution is fitted based on the SCV. In [7] realistic values for the SCV are discussed. Generally this value seems to be in the range of 0.35 up to 0.85.

Other approaches than static scheduling for the single server case are also discussed in literature. Firstly, in [5] an approach, called adaptive scheduling, is discussed that does update the schedule after the server has already started serving the customers. It does so each $\tau_m := m\Delta$ time units, for some predefined interval length $\Delta > 0$. It is concluded that the costs of scheduling adaptive are lower than when scheduling staticly. Secondly, [1] discusses an approach, called dynamic scheduling, for which no such a priori schedule is made at all. The idea of the approach is that the servers get jobs one by one, in the sense that at the moment client i enters the system, the arrival time of client $i + 1$ is scheduled. This is an example of sequential scheduling, the patients are scheduled one by one instead of simultaneously. As for adaptive scheduling it is concluded that scheduling dynamic results in lower costs than scheduling static.

In the field of appointment scheduling that we consider in this Bachelor Final Project, the order of the clients is given, however, there are papers, such as [21], that also discuss the sequencing problem. Moreover, some papers that did not consider this problem mainly, still mentioned it briefly, when considering heterogeneous distributed service times, such as [15].

Less research has been done regarding the multiserver setting. Research about this setting has mostly been restricted to multistage settings, in which people for example first have a CT-scan and then have an appointment with the doctor. In *Appointment scheduling in healthcare* the setting of two servers in tandem is discussed by Kuiper [22]. Some research about the multiserver setting considers the single-stage multiserver setting, so the setting of multiple parallel servers. In [23] a multiple server variant of the phase-type model is discussed.

Other papers discuss further extensions to make the setting more useful in practice by, for example, including no-shows and walk-ins. The literature on including no-shows can basically be split in two categories. One part uses adapted values for the quantities, such as the mean value and variance, that are needed to make a schedule. [10] fits in this category, since they perform a phase-type fit with an adapted mean value, variance and SCV of the expected service time of a client. Literature from the second category considers including a factor p, and/or q for the probability that there is a no-show or a walk-in. In [24] the case with homogeneous exponentially distributed service times including no-shows is discussed using this method. This is done for the phase-type single server case in [22]. Even in the multiserver setting no-shows are included [23]. The method applied in this paper also falls in the latter category.

All of the literature works with a certain objective function for the costs that has to be minimized. The exact function that is used differs a lot. Mostly it is a combination of the expected idle times of the server and the expected waiting times of the clients. Sometimes also the overtime, the extra time that is needed to serve all patients compared to the scheduled endtime, is included. However, by [15], this is not needed, since it has a similar effect as assigning a higher weight to the idle times in the objective function.

Since the goal is to optimize the objective function it is important to know whether this function is convex[1]. For the single server appointment scheduling problem in continuous time, strong arguments have been giving that it is convex [10]. A proof has been added as an appendix, see Appendix C. For the multiserver case such a proof has not been written yet. This is due to the fact that most proofs for the single server case rely on keeping track of the work load per slot by the Lindley recursion, but this does not work since the variables for clients waiting times and servers' workload do not coincide in a multiserver setting [23]. Since in [23] they have strong reason to believe that their solutions are global optima, they believe their multiserver setting is convex.

One of the earlier mentioned reviews of existing literature has the interesting conclusion that despite the large amount of published theoretical work, the use in practice has been very limited. According to this paper the main goal of further research should be to close this gap between theory and practice [7].

[1] In optimization a convex function is known as a function with the property that every optimum is a global optimum.

4 Mathematical model

In this section the modelling framework will be described that is used in this Bachelor Final Project. This setting is intensively used in literature about appointment scheduling. Everything that is stated in this section is applicable for the single server setting, however, this does not hold for the multiserver setting. If something is not applicable it is explicitly mentioned.

As stated in the introduction, in order to generated an optimal schedule, one has the task to define the clients' arrival times that minimize the cost function. We consider a sequence of $n \in \mathbb{N}$ clients with service times that are represented by the independent and non-negative random variables $B_1, B_2, ..., B_n$. A schedule is then defined as an increasing sequence of arrival times $t_1, ..., t_n$ at which the n clients are supposed to arrive at the server. The times are typically chosen such that there is a balance between the interests of the server and the clients. For both parties the goal is to have the least amount of costs, defining the costs for the service provider to be the sum of the idle times and the costs for the clients to be the sum of their waiting times. This results in the following cost function, from now on referred to as the objective function,

$$C[x_1, x_2, ..., x_n] = \omega \sum_{i=1}^{n} \mathbb{E}[I_i] + (1 - \omega) \sum_{i=1}^{n} \mathbb{E}[W_i], \tag{4.1}$$

with I_i the idle time of client i, W_i the waiting time associated with client i and ω a variable factor to define which of the latter two quantities is the most important in the weighted sum. There are a lot of variations possible for such an objective function. For example the objective function could be quadratic or *overtime* could be included. However, the variant above, Equation (4.1), will be used in this Bachelor Final Project. Therefore a more detailed explanation of the different possibilities for the objective function can be found in Appendix B. For this objective function it holds that when ω approaches 1, i.e. the situation in which the value of the objective function is essentially determined by the idle times only, $\sum_{i=1}^{n} \mathbb{E}[W_i]$ explodes. This rule is also known as the utilization law of Hopp and Spearman [22]. On the other hand, when ω approaches 0, i.e. the situation in which the value of the objective function is essentially determined by the waiting times only, $\sum_{i=1}^{n} \mathbb{E}[I_i]$ increases extremely.

The arrival epoch of client i will be denoted by t_i. Clearly t_1 is always set equal to 0 and so $\mathbb{E}[I_i] = 0$, in order to minimize the objective function. The interarrival times $t_i - t_{i-1}$ are denoted by x_i, with $x_1 = 0$. For an illustration of the introduced quantities see Figure 3.

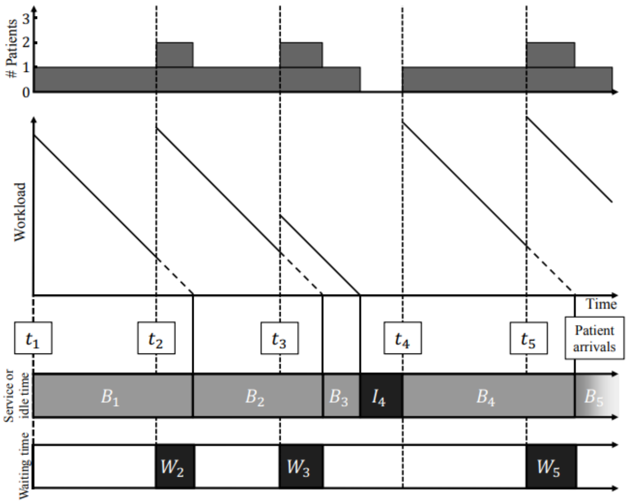


Figure 3: Illustration of the quantities in a single server setting for appointment scheduling. Figure courtesy of A.Kuiper [22].

Since we aim to minimize our objective function, it is important to know whether the function is convex in its arguments, because in that case we are sure that such a (unique) minimum can be found. Since a proof for the single server case is written in [10], we do not discuss that proof in this Bachelor Final Project, but it can be found in Appendix C. For the multiserver case such a proof is not written yet. However, due to some assumptions that are made, the model that we describe in Subsection 6.1 is assumed to be convex [23].

An interesting quantity for appointment scheduling is the time needed until the last client has left the system, called the schedule's makespan. This can be expressed in multiple ways:

$$\sum_{i=1}^{n} I_i + \sum_{i=1}^{n} B_i = t_n + W_n + B_n = \sum_{i=1}^{n} x_i + W_n + B_n; \tag{4.2}$$

here the lefthand side follows from the observation that during the whole makespan, the server is either busy serving a client or not. Therefore the makespan is the sum of the service times (the time that the server is busy) and the idle times (the time that the server is not busy). The other two expressions follow from the realization that the makespan equals the arrival time of the last client increased by their service time and possible waiting time. The arrival time of the last client can either be expressed as t_n or the sum of all interarrival times, $\sum_{i=1}^{n} x_i$. In the single server case we have, due to *Lindley's recursion*, the following equalities for the idle and waiting times

$$I_i = \max\{t_i - t_{i-1} - W_{i-1} - B_{i-1}, 0\}$$

and

$$W_i = \max\{W_{i-1} + B_{i-1} - t_i + t_{i-1}, 0\},$$

or, when we use x_i for the interarrival times,

$$I_i = \max\{x_i - W_{i-1} - B_{i-1}, 0\} \tag{4.3}$$

and

$$W_i = \max\{W_{i-1} + B_{i-1} - x_i, 0\}. \tag{4.4}$$

In the multiserver setting those equalities don't hold as the variables for the waiting time of the clients and the servers' workload do not coincide.

Now we will show how the objective function can be rewritten such that the problem is reformulated in the expected sojourn times only. This version of the objective function is important for the phase-type case of the single server model and the same method is used for other models to rewrite the objective function.

The sojourn time is the total time that a client is in the system. This can be denoted as $S_i = W_i + B_i$. This and the equations for I_i, W_i and S_i are very useful when evaluating the objective function in the phase-type case, since we can rewrite the objective function in terms of the expected sojourn times of the clients, which will be done as follows. Let us have a look at the time at which client i leaves the system. This time is equal to the sum of all service and idle times corresponding to all clients up to and including this client. Hence, for $j = 1, ..., n$ we have

$$\sum_{i=1}^{j} B_i + \sum_{i=1}^{j} I_i,$$

which can be rewritten as $t_j + S_j$, so as the sum of the time that client j arrives and the sojourn time of client j (This fact can also be observed from Figure 3). From this equality it can be concluded that we can express the (expected) waiting and idle times in terms of the (expected) sojourn times. Obviously $\mathbb{E}[W_1] = \mathbb{E}[I_1] = 0$ and for $i = 2, ..., n$,

$$\mathbb{E}[W_i] = \mathbb{E}[S_i] - \mathbb{E}[B_i], \; \mathbb{E}[I_i] = t_i + \mathbb{E}[W_i] - (t_{i-1} + \mathbb{E}[S_{i-1}]). \tag{4.5}$$

The expression for the expected idle time can be interpreted as the expected difference between the service completion of client $i-1$ and the start of service of client i. As can be observed, we have now rewritten the quantities $\mathbb{E}[W_i]$ and $\mathbb{E}[I_i]$ in terms of only the arrival times (t_i and t_{i-1}), which are the values we will optimize, the expected service times $\mathbb{E}[B_i]$, which are given and the expected sojourn times $\mathbb{E}[S_i]$, which we can calculate. So it becomes clear that this reformulation makes it easier to evaluate the objective function. Using those equalities, our objective function becomes:

$$\begin{aligned}
C[x_1, x_2, ..., x_i] &= \omega \sum_{i=1}^{n} \mathbb{E}[I_i] + (1-\omega) \sum_{i=1}^{n} \mathbb{E}[W_i] \\
&= \omega \sum_{i=1}^{n} (t_i + \mathbb{E}[W_i] - (t_{i-1} + \mathbb{E}[S_{i-1}])) + (1-\omega) \sum_{i=1}^{n} (\mathbb{E}[S_i] - \mathbb{E}[B_i]) \\
&= \omega \sum_{i=1}^{n} (t_i + \mathbb{E}[S_i] - \mathbb{E}[B_i] - (t_{i-1} + \mathbb{E}[S_{i-1}])) + (1-\omega) \sum_{i=1}^{n} (\mathbb{E}[S_i] - \mathbb{E}[B_i]) \\
&= \omega \sum_{i=1}^{n} (t_i - t_{i-1} + \mathbb{E}[S_i] - \mathbb{E}[S_{i-1}] - \mathbb{E}[B_i]) + (1-\omega) \sum_{i=1}^{n} (\mathbb{E}[S_i] - \mathbb{E}[B_i]) \\
&= \omega \left(t_n + \mathbb{E}[S_n] + \sum_{i=1}^{n} (-\mathbb{E}[B_i]) \right) + (1-\omega) \sum_{i=1}^{n} (\mathbb{E}[S_i] - \mathbb{E}[B_i]) \\
&= \omega \left(\sum_{i=1}^{n} x_i + \mathbb{E}[S_n] - \sum_{i=1}^{n} \mathbb{E}[B_i] \right) + (1-\omega) \sum_{i=1}^{n} (\mathbb{E}[S_i] - \mathbb{E}[B_i]). \qquad (4.6)
\end{aligned}$$

There are more options to rewrite our objective function. If another rewritten version is used for a model, this is stated in the corresponding section. Lastly we will define the assumptions under which we investigate the problem, they are listed below:

1. The number of servers, s, equals one.

2. The order of the served clients is fixed, i.e. The service order of the clients is first appointment, first serve.

3. The clients are punctual, i.e. client i arrives at its assigned time t_i.

4. The server is punctual, i.e., if the server is idle and a client arrives, the server will immediately start serving the client.

5. There is no additional stream of urgent arrivals, i.e. walk-in clients.

6. All clients show up for their appointment, i.e., the model does not include no-shows.

7. The service times of the customers are modelled by a probability distribution. Unless stated otherwise, the service times for all customers are i.i.d.

From those assumptions it becomes clear that we study a queue with fixed arrival times, random general distributed service times and one server. Further on in this Bachelor Final Project some of those restriction will be relaxed, in particular the first, fifth and sixth, such that our results will become more applicable to practical situations.

5 Single server models

In this section a static, a dynamic and an adaptive model for the single server case will be described. Next to the mathematical descriptions also some experiments executed for those models will be discussed.

5.1 Static model

The static model is also called the 'a priori' model. Using this model, the optimal arrival times are determined before the server has started serving the customers and they can not be changed after the process of serving has started. In this subsection a static model for the exponential homogeneous and phase-type case will be discussed. After a description of the mathematical model of those cases, also the results of some experiments are presented in Subsection 5.1.3. First some experiments are performed that provide the reader with some general knowledge on appointment scheduling. After those, we also describe, by means of some experiments, how appointment scheduling can be used in practice.

5.1.1 Exponential homogeneous

For the exponential homogeneous case we studied the model described in the paper *Scheduling arrivals* to queues written by Claude Dennis Pegden and Matthew Rosenshine [14]. This paper considers the problem of finding an optimal schedule for n arriving customers to a server. The goal is finding the schedule for the n customers which minimizes the total system cost which comprises the customer waiting time and the server's availability. Their objective function differs from ours, see Equation (4.1), since they take into account the server's availability instead of the servers idle time, but of course the described algorithm for finding the schedule is still applicable. First we look at how we have rewritten our objective function to make the model applicable and then we look at the algorithm for calculating the schedule.

We have rewritten our objective function as follows:

$$\begin{aligned}
C[x_1, x_2, ..., x_i] &= \omega \sum_{i=1}^{n} E[I_i] + (1-\omega) \sum_{i=1}^{n} E[W_i] \\
&= \omega \sum_{i=1}^{n} (t_i + E[W_i] - (t_{i-1} + E[S_{i-1}])) + (1-\omega) \sum_{i=1}^{n} E[W_i] \\
&= \omega \sum_{i=1}^{n} (t_i + E[W_i] - (t_{i-1} + E[W_{i-1}] + E[B_{i-1}])) + (1-\omega) \sum_{i=1}^{n} E[W_i] \\
&= \omega \sum_{i=1}^{n} \left(t_i + E[W_i] - \left(t_{i-1} + E[W_{i-1}] + \frac{1}{\mu}\right)\right) + (1-\omega) \sum_{i=1}^{n} E[W_i] \\
&= \omega \sum_{i=1}^{n} \left(t_i + E[W_i] - t_{i-1} - E[W_{i-1}] - \frac{1}{\mu}\right) + (1-\omega) \sum_{i=1}^{n} E[W_i] \\
&= \omega \sum_{i=1}^{n} \left(x[i-1] + E[W_i] - \left(E[W_{i-1}] + \frac{1}{\mu}\right)\right) + (1-\omega) \sum_{i=1}^{n} E[W_i]. \qquad (5.1)
\end{aligned}$$

We have done this such that the objective function is rewritten as a function of only the waiting times, since the $\mathbb{E}[B_{i-1}]$'s are known and the arrival times, the $t_i's$, are the values that we are optimizing.

To be able to calculate the value of the objective function, the expected waiting times should be computed first. For this, we have used the algorithm of Pegden and Rosenshine. Before diving into the mathematics it is important to mention that the definition of x_i in this paper is different from most literature. In the paper of Pegden and Rosenshine x_i is the time interval between the scheduled arrival times of the i-th and the $(i+1)$-th customer, instead of between the $(i-1)$-th and i-th customer.

Now we will describe how the expressions for the waiting times as functions of the interarrival times can be derived. An important quantity for this is $N(t_i)$, which is defined as the number of customers in the system just prior to the time of the i-th arrival. Since the expected service times of all clients are the same, the expected waiting time for customer i arriving at time t_i depends only upon the number of customers in the system at time t_i. This gives the following expression for the waiting times:

$$w_i = \sum_{j=1}^{i-1} \left(\frac{j}{\mu}\right) \mathbb{P}(N(t_i) = j), \text{ with } j \in \{0, ..., n-1\}. \tag{5.2}$$

From now on a case distinction is made for $j = 0$ and for $j > 0$. For both cases it will be shown how the values of $\mathbb{P}(N(t_i) = j)$ are derived and the probability will be computed from the state probabilities at time t_{i-1}.

Case $j > 0$ In this case the probability can be written as

$$\begin{aligned} \mathbb{P}(N(t_i) = j) = &\sum_{k=0}^{i-j-1} \mathbb{P}(j+k-1 \text{ customers in the system just prior to } (i-1)\text{-st arrival}) \\ &\times \mathbb{P}(k \text{ departures between the } (i-1)\text{-st and the } i\text{-th arrival}). \\ = &\sum_{k=0}^{i-j-1} \frac{(\mu x_{i-1})^k}{k!} e^{-\mu x_{i-1}} \mathbb{P}(N(t_{i-1}) = j+k-1) \text{ with } j > 0,\ i \geqslant 2. \end{aligned} \tag{5.3}$$

Here the probability that there are k departures between the $(i-1)$-st and the i-th arrival equals the probability that exactly k events in a Poisson process with rate μ happen.

Case $j = 0$ Observe that $N(t_i) = 0$ if there are $k-1$ customers in the system just prior to the $(i-1)$-st arrival and the service times between the $(i-1)$-st and the i-th arrival are such that the sum of k service times is less than x_{i-1}. So if $j = 0$, the probability $\mathbb{P}(N(t_i) = j)$ can be obtained as follows:

$$\begin{aligned} \mathbb{P}(N(t_i) = 0) = &\sum_{k=1}^{i-1} \mathbb{P}(k-1 \text{ customers in the system just prior to the } (i-1)\text{-st arrival}) \\ &\times \mathbb{P}(\text{ time between } (i-1)\text{-st and the } i\text{-th arrivals is sufficient for } k \text{ or more departures}) \\ = &\sum_{k=1}^{i-1} \mathbb{P}(N(t_{i-1}) = k-1) \sum_{l=k}^{\infty} \frac{(\mu x_{i-1})^l e^{-\mu x_{i-1}}}{l!} \\ = &\sum_{k=1}^{i-1} \mathbb{P}(N(t_{i-1}) = k-1) \left(1 - \sum_{l=0}^{k} \frac{(\mu x_{i-1})^l e^{-\mu x_{i-1}}}{l!}\right) \text{ with } i \geqslant 2\ . \end{aligned} \tag{5.4}$$

This all results in the following algorithm to derive the expression for the waiting times of all clients.

Algorithm 1 Expected waiting times for exponential homogeneous static case

```
Set $w_1 = 0$.
Set $\mathbb{P}\{N(t_1) = 0\} = 1$
for $i = 2, 3, ..., n$ do
    for $j = 0, ..., i-1$ do
        $\mathbb{P}(N(t_i) = j) = \sum_{k=0}^{i-j-1} \frac{(\mu_{i-1})^k}{k!} e^{-\mu x_{i-1}} \cdot \mathbb{P}(N(t_{i-1}) = j+k-1)$
    end for
end for
for $i = 2, 3, ..., n$ do
    $w_i = \sum_{j=1}^{i-1} (\frac{j}{\mu}) \cdot \mathbb{P}(N(t_i) = j)$
end for
Return $w_1, ...., w_n$.
```

Now the expressions for the waiting times are derived and the optimal schedule is determined by first inserting the expressions of the waiting times in the objective function and then optimizing this function with respect to t_i with $i \in \{1, ..., n\}$ as variables.

5.1.2 Phase-type distributions

As stated in the literature overview it is not realistic to assume that the service times are exponentially distributed, therefore we also discuss the phase-type case. In this subsection a model that can be used to derive a static schedule when the service times are phase-type distributed is described and in an example it is shown what some of the matrices that are used look like in the hyperexponential case. In Section 2 it is already described how a phase-type distribution can be fitted when the mean and variance of all service times are known. In this subsection we will present a method to derive the expected sojourn times of all clients. As described in Section 4 the value of the (rewritten) objective function can then be calculated and thus the optimal schedule can be obtained. The procedure presented in this section is the one described in [5].

For all clients that are to be scheduled their phase-type fit is derived as described in Section 2 and therefore $B_i \sim \mathrm{PH}_{d_i}(\boldsymbol{\gamma}_i, \boldsymbol{T_i})$ for $i \in \{1, ..., n\}$. Given the schedule $t_1, ...t_n$, let $x_i = t_i - t_{i-1}$ be the i-th interarrival time. Furthermore let $N_i(t)$ denote the number of clients in the system at the shifted time $t \in [0, x_{i+1})$, so t time units after the arrival of the i-th client and let $Z_i(t)$ denote the phase the client in service is in at the same time t. When the system is idle, so when no client is in service, we set $Z_i(t) = 0$. Now denote the probability that the system is in state (k, z) at shifted time t by

$$p_{kz}^{(i)}(t) := \mathbb{P}\left(N_i(t) = k, Z_i(t) = z\right), \text{ with } i = 1, ..., n,\ k = 1, ..., i \text{ and } z = 1, ...d_{i-k+1}.$$

Observe that if $N_i(t) = k$, so if there are k clients in the system at shifted time t, then the index of the client is service is $i - k + 1$. Furthermore, observe that this probability is only defined when the system is busy, so when the server is serving a job, since the only case for which the probability is not defined is $(0, 0)$. Now the sojourn time distribution $F_i(t)$ of the i-th client equals

$$\begin{aligned} F_i(t) :&= \mathbb{P}(S_i \leqslant t) \\ &= 1 - \mathbb{P}(S_i > t) \\ &= 1 - \sum_{k=1}^{i} \sum_{z=1}^{d_{i-k+1}} p_{kz}^{(i)}(t) \\ &= 1 - \boldsymbol{P}_i(t) \mathbf{1}_{\sum_{k=1}^{i} d_k} \\ &= 1 - \boldsymbol{P}_i(t) \mathbf{1}_{D_i}, \end{aligned} \tag{5.5}$$

here $\mathbf{P}_i(t)$ is defined as

$$\mathbf{P}_i(t) := \lim_{s \uparrow t} \left(p_{i,1}^{(i)}(s), \ldots, p_{i,d_1}^{(i)}(s), p_{i-1,1}^{(i)}(s), \ldots, p_{i-1,d_2}^{(i)}(s), \ldots, p_{1,1}^{(i)}(s), \ldots, p_{1,d_i}^{(i)}(s)\right),$$

$D_i := \sum_{k=1}^{i} d_i$ and $\mathbf{1}_d$ denotes the all-ones vector of dimension $d \in \mathbb{N}$. Recall that d_i denotes the dimension of $\boldsymbol{\gamma}_i$. Now, the expected sojourn times of all clients $i = 1, ..., n$ at time $t \geqslant 0$ are calculated using Algorithm 2 [5].

Algorithm 2 Expected sojourn times in phase-type single server case

```
 1: $\boldsymbol{G}_1 = \gamma_1$
 2: Set $\boldsymbol{V}_1 = \boldsymbol{T}_1$
 3: for $i = 2, 3, ..., n$ do
 4:     $\boldsymbol{G}_i = \left[\boldsymbol{P}_{i-1}(x_i), \gamma_i F_{i-1}(x_i)\right];$
 5:     $\boldsymbol{V}_i = \left[\begin{array}{c|c} \boldsymbol{V}_{i-1} & \begin{array}{c} \mathbf{0}_{D_{i-2}\times d_i} \\ \left(-\boldsymbol{T}_{i-1}\mathbf{1}_{d_{i-1}}\right)\gamma_i \end{array} \\ \hline \mathbf{0}_{d_i\times D_{i-1}} & \boldsymbol{T}_i \end{array}\right];$
 6:     $\boldsymbol{P}_i(t) = \boldsymbol{G}_i \exp\left(\boldsymbol{V}_i t\right);$
 7:     $F_i(t) = 1 - \boldsymbol{P}_i(t)\mathbf{1}_{D_i}$
 8: end for
 9: for $i = 1, ..., n$ do
10:     $\mathbb{E}[S_i] = -\boldsymbol{G}_i\boldsymbol{V}_i^{-1}\mathbf{1}_{D_i}$
11: end for
12: Return $\mathbb{E}[S_1], ..., \mathbb{E}[S_n]$.
```

The theory about continuous phase-type distributions from Subsection 2.2 is extremely useful for understanding how the expressions in line 4-7 and 10 are derived. The recursion for $\boldsymbol{G}_i$ can be explained by the fact that when client i arrives at time x_i there are two possible scenarios. Either a previous client is still in service and thus client i needs to wait or the service of all previous clients has been completed, with probability $F_{i-1}(x_i)$, and thus client i immediately goes into service. The recursion for $\boldsymbol{V}_i$ can be explained by the same argument. To get a better feeling for how the recursion works and what the matrices look like, an example is worked out in Example 1.

Since the expressions for the expected sojourn times are derived, the optimal schedule can now be derived by first inserting those expressions in the objective function and then optimizing this function with respect to t_i with $i \in \{1, ..., n\}$ as variables. For this, the rewritten version of the objective function as described in Section 4 is used.

Example 1. Algorithm 2 for fitted hyperexponential
Assume that client $i = 1$ and $i = 2$ both have a SCV such that a hyperexponential distribution is fitted. Then we get, by Algorithm 2, the following matrices. Since both clients get a hyperexponential fit $\boldsymbol{G}_i = \gamma_1 = \begin{pmatrix} p_1 & (1-p_1) \end{pmatrix}$, $\gamma_2 = \begin{pmatrix} p_2 & (1-p_2) \end{pmatrix}$,

$$\boldsymbol{V}_1 = \boldsymbol{T}_1 = \begin{pmatrix} -\mu_{1,1} & 0 \\ 0 & -\mu_{1,2} \end{pmatrix} \text{ and } \boldsymbol{T}_2 = \begin{pmatrix} -\mu_{2,1} & 0 \\ 0 & -\mu_{2,2} \end{pmatrix}.$$

For client $i = 1$ we then get:

$$\boldsymbol{P}_1 = \gamma_1 \exp(\boldsymbol{V}_1 t) = \begin{pmatrix} p_1 & (1-p_1) \end{pmatrix} \cdot \begin{pmatrix} e^{-\mu_{1,1}} & 0 \\ 0 & e^{-\mu_{1,2}} \end{pmatrix} = \begin{pmatrix} p_1 \cdot e^{-\mu_{1,1}} & (1-p_1)\cdot e^{-\mu_{1,2}} \end{pmatrix} \text{ and}$$

$$F_i(t) = 1 - \boldsymbol{P}_i(t)\mathbf{1}_{D_i} = 1 - p_1 \cdot e^{-\mu_{1,1}} - (1-p_1)\cdot e^{-\mu_{1,2}}.$$

For client $i = 2$ we then get:

$$G_2 = \left[\boldsymbol{P}_1(x_2), \gamma_2 F_1(x_2)\right] = \begin{pmatrix} pe^{-\mu_{1,1}t} & (1-p)e^{-\mu_{1,2}t} & p_2F_1(x_2) & (1-p_2)F_1(x_2) \end{pmatrix},$$

$$\boldsymbol{V}_2 = \begin{pmatrix} -\mu_{1,1} & 0 & \mu_{2,1}\gamma_{2,1} & \mu_{2,1}\gamma_{2,2} \\ 0 & -\mu_{1,2} & \mu_{2,2}\gamma_{2,1} & \mu_{2,2}\gamma_{2,2} \\ 0 & 0 & -\mu_{2,1} & 0 \\ 0 & 0 & 0 & -\mu_{2,2} \end{pmatrix} \text{ and}$$

$$\boldsymbol{P}_2(t) = \boldsymbol{G}_2 \cdot \exp(\boldsymbol{V}_2 t).$$

.

5.1.3 Experiments single server static scheduling

In this subsection the results of some experiments performed for the static single server case are discussed. The first two provide the reader some general results of appointment scheduling in the single server case, the next three experiments provide an example of how appointment scheduling can be used in healthcare. As for all experiments in this Bachelor Final Project it holds that, when not stated otherwise, $n = 5$, $s = 1$, $\omega = 0.5$.

Experiment 5.1 Dome-shape
In this first experiment we have studied the interarrival times of the optimal schedule when scheduling static using the model for homogeneous exponential distributed service times described in Subsection 5.1.1 with as parameter $\mathbb{E}[B_i] = 1$ for $i \in \{1, ..., 5\}$. When applying this method we yielded the following plot.

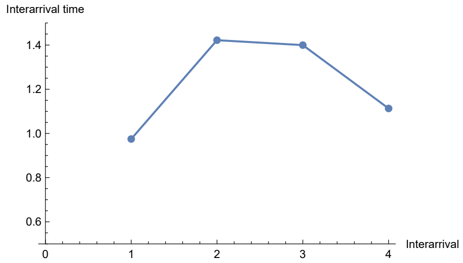
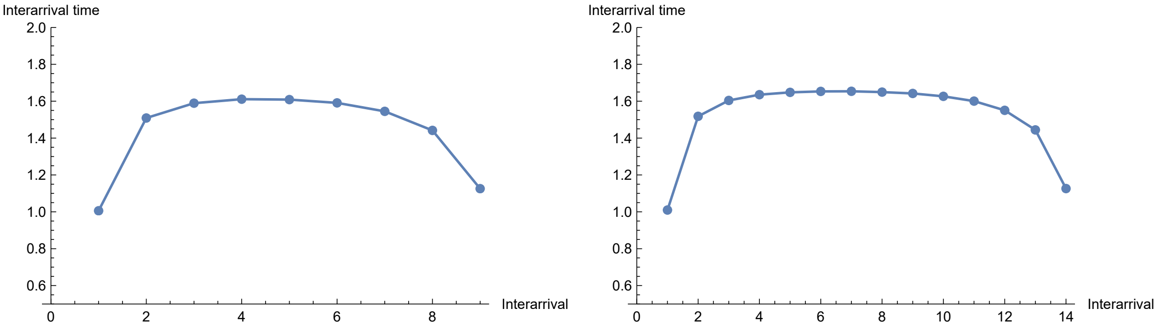


Figure 4: Interarrival times static scheduling Experiment 5.1.

As can be seen in Figure 4, the interarrival times have a so-called *dome-shape* [10]. The optimal interarrival times increase in the beginning and decrease towards the end of the schedule. The short interarrival times in the beginning can be explained by the fact that the risk of waiting is relatively low, since there are less clients scheduled before you. The short interarrival times at the end can be explained by the fact that there are less clients after you, so only a few clients are suffering from the possibly high waiting times. In the middle the interarrival times are nearly constant. This result becomes more clear when looking at the interarrival times for higher values of n, see Figure 5. This phenomenon is studied more in depth by Kuiper, Kemper and Mandjes in [15], that discusses *stationary schedules.*

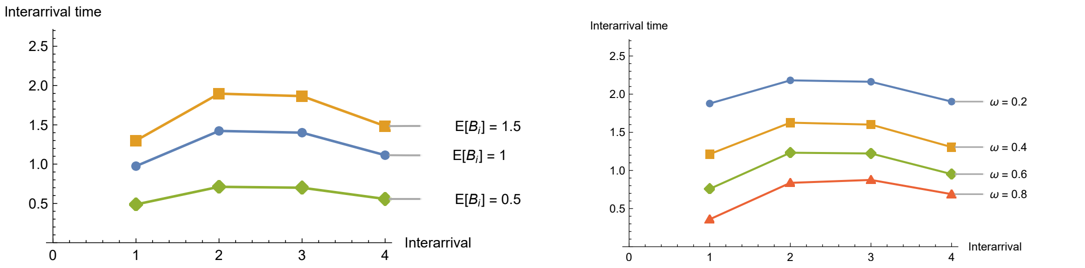


(a) Case n = 10.

(b) Case n = 15.

Figure 5: Interarrival times exponential homogeneous static scheduling.

Experiment 5.2 Influence ω, variance and mean
In this experiment we study the influence of the expected value of the service time, ω and variance of the service time on the interarrival times and costs. We have used the exponential homogeneous model for checking the influence of the first two quantities and the phase-type model is used for checking the influence of the latter one. If not stated otherwise $E[B_i] = 1$ for all i. The interarrival times for different expected values and ω's can be found in Figure 6.



(a) Interarrival times for different $\mathbb{E}[B_i]$'s.

(b) Interarrival times for different ω's.

Figure 6: Interarrival times for varied $\mathbb{E}[B_i]$ or ω.

As is generally known, it can be observed that when the expected value gets higher or the value of ω gets lower, the interarrival times increase. It is intuitive that when the patients have a higher expected service time, their interarrival times increase. The second observation can also be explained easily. As ω increases in our objective function, see Equation (4.1), it means that the waiting times get less important and the idle times get more important for the costs and thus you want to prevent having those. Therefore the interarrival times will decrease.

$\mathbf{E[B}_i\mathbf{]}$	**0.5**	**1.0**	**1.5**
cost	0.94	1.88	2.51

Table 1: Costs for $\mathbb{E}[B] \in \{0.5, 1.0, 1.5\}$.

$\boldsymbol{\omega}$	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
costs	0.98	1.46	1.74	1.87	1.88	1.78	1.56	1.21	0.71

Table 2: costs for $\omega \in \{0.1, ..., 0.9\}$.

From Table 1 it can be observed that costs become higher when the expected service time per client increases. This seems intuitive, since for example waiting times can be much larger. From Table 2 it can be observed that when $\omega = 0.5$ the costs are higher than for any other value of ω, when this value increases or decreases the costs will get lower. This has also been observed when n or $\mathbb{E}[B]$ had different values. It can be explained by the fact that when $\omega = 0.5$ there is no preference between idle times and waiting times. Lastly we look at the influence of the variance. As in the literature, it is found that the variance has a high influence on the costs, as can be seen in Table 3, but not that much on the interarrival times, as can be seen in Table 7.

Variance	**costs**
0.5	1.337
1	1.881
1.5	2.181
2	2.398

Table 3: Costs when variance $\in \{0.5, 1, 1.5, 2\}$.

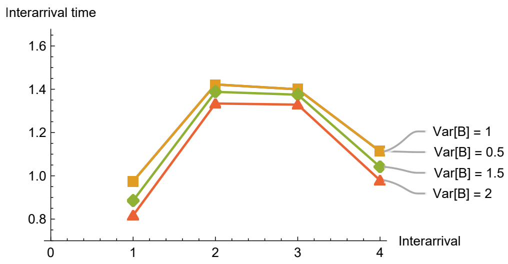


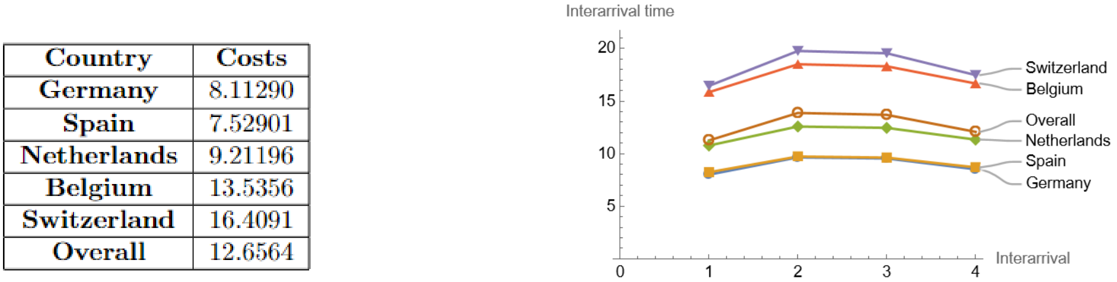
Figure 7: Interarrival times when variance $\in \{0.5, 1, 1.5, 2\}$.

For the following experiments information found about the mean service times for appointments with the general practitioner in different countries is used. In [25] a cross sectional study in six European countries of the consultation length in general practice is described. For this research consultations at a general practice are filmed, which is according to [26] the best method of researching such consultations. For us the main interest is Table 4, that provides the mean and standard deviation of the consultation length of six different countries, since we can use these values to perform a phase-type fit and therefore find an appropriate distribution. Those values result in SCV's in the range from 0.20 up to 0.40, consistent with [7], where it is stated that CV values of the consultation length are typically in the range from 0.35 up to 0.85 and thus the SCV's are typically in between 0.1225 and 0.7225.

Country	**Mean**	**Standard deviation**
Germany	7.6	4.3
Spain	7.8	4.0
Netherlands	10.2	4.9
Belgium	15.0	7.2
Switzerland	15.6	8.7
Overall	10.7	6.7

Table 4: Mean and standard deviation of consultation length in general practice of six countries [25].

Experiment 5.3 Costs and interarrival times of different countries
In this experiment we want to calculate the cost and interarrival times, when scheduling statically, when patients are from different countries using the phase-type approach. It is expected that those values are different, since this is basically just static scheduling with different values for the mean and variance and thus for the SCV. Observe that in Table 4 the standard deviation is given, but the variance is needed so therefore we need to square this value. In Figure 8 the costs and the interarrival times are presented when a schedule is made for the different countries with $n = 5$ and $\omega = 0.5$. From Figure 8 it can be observed that the optimal schedules for the different countries indeed differ. Moreover, it can be observed that the minimal costs increase when the variance increases and the interarrival times increase when the mean consultation length increases.



Country	**Costs**
Germany	8.11290
Spain	7.52901
Netherlands	9.21196
Belgium	13.5356
Switzerland	16.4091
Overall	12.6564

(a) Costs of Experiment 5.3.

(b) Interarrival times of Experiment 5.3.

Figure 8: Costs and interarrival times of Experiment 5.3.

Experiment 5.4. Expected sojourn time distribution and expected makespan
In Subsection 5.1.2 we described a method to calculate the expected sojourn time. When adding this quantity from the last client and all the planned interarrival times, the expected total time that the server needs to serve all clients, the makespan, can be calculated. In the same section, an expression for the sojourn time distribution is given in Equation (5.5). This function can be used to calculate the distribution of the makespan. In this experiment we show the calculations for both quantities for the Netherlands, $n = 5$ and $\omega = 0.5$. For this case the costs and interarrival times are presented in Figure 8. Using those interarrival times, mean and variance of the consultation length for all 5 clients, the expected sojourn time of the last, the fifth, client is calculated, resulting in $\mathbb{E}[S_5] = 13.2501$. This means that we expect the doctor to be done with serving those five clients at $t = \sum_{i=1}^{4} x_i + \mathbb{E}[S_5] = 60.39$ minutes. The distribution of the sojourn time of client i can be

calculated as follows:

$$\begin{aligned} F_i(t) :&= \mathbb{P}(S_i \leqslant t) \\ &= 1 - \mathbb{P}(S_i > t) \\ &= 1 - \sum_{k=1}^{i} \sum_{z=1}^{d_{i-k+1}} p_{kz}^{(i)}(t) \\ &= 1 - \mathbf{P}_i(t)\mathbf{1}_{\sum_{k=1}^{i} d_k}. \end{aligned}$$

The sojourn time distribution of our last, fifth, client is plotted in Figure 9.

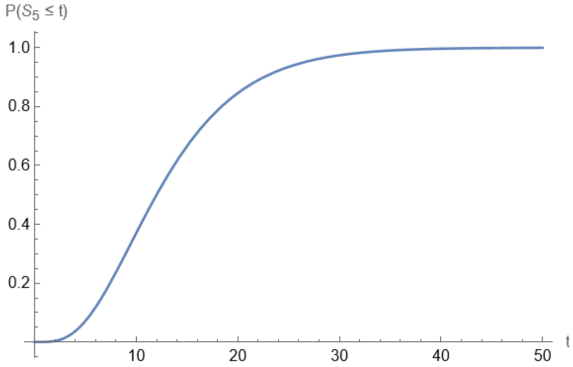


Figure 9: Sojourn time distribution of client $i = 5$ in case $n = 5$ and $\omega = 0.5$ in the Netherlands.

If the doctor wants to be 95 percent sure about his end time, he wants $\mathbb{P}(S_i \leqslant t)$ to be at least 0.95. It is found that from $t = 27$ min onwards $\mathbb{P}(S_5 \leqslant t) > 0.95$, so the doctor should use this value to calculate the length of the makespan such that he is 95 percent sure about being finished at that time. Thus the doctor should take as planned makespan $t = \sum_{i=1}^{4} x_i + 27 = 74.14$ minutes.

Experiment 5.5. Advised amount of clients to be scheduled

This last experiment is thought of because in a real life situation it would be likely that a general practitioner is interested in knowing how many patients he should schedule to be 95 percent sure that he is done serving all clients at the end of his workday. Most general practitioners work from around 8:00 until 17:00 with a one hour break, which would result in a workday of 8 hours. In the Netherlands it is then expected, using the values from Table 4, that around 45 patients can be scheduled using the information from [25]. Since our code would run for too long, due to possibly extremely large matrices and the fact that efficiency of the code did not have priority, the experiment is executed for 1 hour instead of 8 hours. Hence the goal is to calculate how many patients can be scheduled such that we can be 95 percent sure that the doctor is done serving patients at $t = 60$ min, for the case that we look at the Netherlands and take $\omega = 0.5$. Therefore for $n = 1, ..., 5$ the value of $\mathbb{P}(\sum_{i=1}^{n-1} x_i + S_n \leqslant 60)$ is calculated. A plot has been made from the probability that the last client is served at time t, see Figure 10. Here $prob_{Mn}$ is defined as $\mathbb{P}(\sum_{i=1}^{n-1}(x_i + S_n) \leqslant t)$. Since from $n = 4$ onwards the probability that the last client is served at $t = 60$ is less than 0.95, the doctor will be advised to schedule 3 patients.

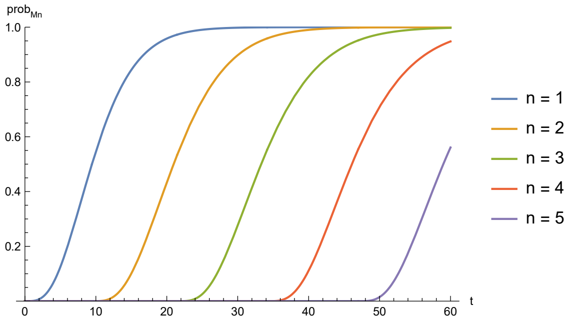


Figure 10: Makespan distribution for $n = 1, ..., 5$ in case $n = 5$ with $\omega = 0.5$ in the Netherlands.

5.2 Dynamic model

In this section we describe the dynamic method for the case of homogeneous and heterogeneous exponentially distributed service times as in [1], in which a technique is set up to determine the optimal arrival time of the next client. The dynamic approach is a sequential approach as the n-th client is scheduled after the $(n-1)$-st client has arrived. You are thus scheduling knowing that you will schedule again, when the next client arrives. Since the n-th client is scheduled after the $(n-1)$-st client has arrived, there is less uncertainty than in the static case, so it is expected that the costs will be lower, which is the main advantage of scheduling dynamic instead of static. The main disadvantage is that, since the n-th client is scheduled after the $(n-1)$-st client, the clients have to be free all day, since they do not know when they are scheduled. First we will look at the exponential homogeneous case, then at the exponential heterogeneous case and lastly we discuss some experiments performed with those models.

5.2.1 Exponential homogeneous

In the case of exponentially distributed service times, the state of the system is just the number of clients that are waiting, since the elapsed service time of the client in service is irrelevant due to the memorylessness property. Recall that the memorylessness property states that if we define X as an exponential random variable, then $\mathbb{P}(X > x + a | X > a) = \mathbb{P}(X > x)$, for $a, x \geqslant 0$. We want to evaluate the costs between the arrival of the i-th client and the $(i+1)$-st client. We identify time 0 with the arrival of client i and we assume that immediately after this arrival there are k clients in the system. Time t is then defined as the time at which $(i+1)$-st client is scheduled to arrive. Moreover, we define N_s as the number of clients in the system at time $s \in [0,t]$ including the client that is in service. The contribution of the idle time to the cost function due to the interval $[0,t]$ is $\omega f_k(t)$ with

$$\begin{aligned} f_k(t) &:= \int_0^t \mathbb{E}(\mathbb{1}_{\{N_s=0\}} | N_{0+} = k)\mathrm{d}s \\ &= \int_0^t \mathbb{P}(N_s = 0 | N_{0+} = k)\mathrm{d}s \\ &= t \cdot (1 - F_{\mu t}(k-1)) - \frac{k}{\mu} \cdot (1 - F_{\mu t}(k)), \end{aligned} \tag{5.6}$$

for $k = 1, ..., i$ and $t \geqslant 0$. Here $F_\mu(k) := \mathbb{P}(\text{Pois}(\mu) \leqslant k)$ denotes the distribution function of a Poisson random variable with mean μ. The contribution of the waiting time to the cost function due to the interval $[0,t]$ is $(1-\omega)g_k(t)$ with

$$\begin{aligned} g_k(t) :&= \int_0^t \sum_{l=0}^{k-1} (k-l-1)\mathbb{E}(\mathbb{1}_{\{N_s=k-l\}} | N_{0+} = k)\mathrm{d}s \\ &= \int_0^t \sum_{l=0}^{k-1} (k-l-1)\mathbb{E}(\mathbb{1}_{\{N_s=k-l\}} | N_{0+} = k)\mathrm{d}s \\ &= (k-1)t \cdot F_{\mu t}(k-1) - \frac{\mu t^2}{2} \cdot F_{\mu t}(k-2) + \frac{k(k-1)}{2\mu} \cdot (1 - F_{\mu t}(k)), \end{aligned} \tag{5.7}$$

for $k = 1, ..., i$ and $t \geqslant 0$ (see [1]). The last important quantity are the transition probabilities

$$p_{kl}(t) := \mathbb{P}(N_{t+} = l | N_{0+} = k) \text{ for } k = 1, ..., i \text{ and } l = 1, ..., k+1.$$

Their explicit form is described as:

$$p_{k1}(t) = \sum_{m=k}^{\infty} e^{-\mu t} \frac{(\mu t)^m}{m!}, p_{kl}(t) = e^{-\mu t} \frac{(\mu t)^{k-l+1}}{(k-l+1)!}, \tag{5.8}$$

for $k = 1, ..., i$ and $l = 2, ..., k+1$ and $t \geqslant 0$. These can be derived easily. The first quantity in Equation (5.8) can be explained by $p_{k1}(t)$ being a Poisson process, where at least k events must occur in the time interval $[0, t]$. The second quantity in Equation (5.8) one can be explained by $p_{kl}(t)$ being a Poisson process, where exactly $k - l + 1$ events must occur in the time interval $[0, t]$. The dynamic programming algorithm to find the optimal arrival time of the next client is then defined as described in Algorithm 3. Here $C_i(k)$ with $i = 1, ..., n$ and $k = 1, ..., i$ defines the cost incurred from the arrival of the i-th client, given that there are k clients in the system immediately after the arrival of this i-th client.

Algorithm 3 Dynamic program exponential homogeneous case

Let $f_k(t)$, $g_k(t)$ and $p_{kl}(t)$ be given as described above. We can determine the $C_i(k)$ recursively for $i = 1, ..., n - 1$ and $k = 1, ..., i$

$$C_i(k) = \inf_{t \geqslant 0} (\omega f_k(t) + (1 - \omega) g_k(t) + \sum_{l=1}^{k+1} p_{kl}(t) C_{i+1}(l)))$$

whereas, for $k = 1, ..., n$,

$$C_n(k) = (1 - \omega) g_k(\infty) = (1 - \omega) \frac{k(k-1)}{2\mu}.$$

Instead of using $g_k(t)$ also h_k, defined as $h_k = \frac{k-1}{\mu}$, can be used. This denotes the expected waiting time of a client if the number of clients immediately after his arrival is k instead of the expected waiting time by all customers in the system between two subsequent arrivals, given that there are k clients present at the beginning of the slot.

5.2.2 Exponential heterogeneous

In this subsection the dynamic programming approach is discussed in the case of heterogeneous exponentially distributed service times, as presented in [1]. The same approach as for homogeneous exponentially distributed service times, as in Subsection 5.2.1, will be used. We will define $f_{ki}(t)$ and $g_{ki}(t)$, the counterparts of $f_k(t)$ and $g_k(t)$ and the transition probabilities. But before doing so we will first have a look at how the density of the sum of independent exponential random variables can be written as a mixture of exponential terms, a fact that we will extensively use for the expressions of the latter quantities.

As equivalent to notation in previous sections, the means of the $B_i's$ are denoted by $\frac{1}{\mu_i} \in (0, \infty)$ for client $i = 1, ..., n$. Let

$$\varphi_{kl}(s) := \frac{d}{ds} \mathbb{P} \left(\sum_{j=k}^{k+l} E_j \leqslant s \right)$$

denote the density of the sum of independent exponentially distributed random variables, with E_j denoting an exponentially distributed random variable with mean $\frac{1}{\mu_j}$. This density can also be written as a mixture of exponential terms:

$$\varphi_{k\ell}(s) = \sum_{j=k}^{k+\ell} c_{k\ell j} e^{-\mu_j s},$$

for $k = 1, 2, \ldots$ and $\ell = 0, 1, \ldots$ and $s \geqslant 0$, there are constants $c_{k\ell j} \in \mathbb{R}$. The coefficients $c_{k\ell j}$ are given recursively through $c_{k0k} = \mu_k$ and

$$c_{k,\ell+1,j} = c_{k\ell j} \frac{\mu_{k+\ell+1}}{\mu_{k+\ell+1} - \mu_j} \text{ for } j = k, \ldots, k + \ell, \quad c_{k,\ell+1,k+\ell+1} = \sum_{j=k}^{k+\ell} c_{k\ell j} \frac{\mu_{k+\ell+1}}{\mu_j - \mu_{k+\ell+1}}.$$

Now we will define the counterpart of the quantities $f_k(t)$ and $g_k(t)$ of Subsection 5.1.1. For $f_{ki}(t)$

we have

$$\begin{aligned}
f_{ki}(t) &:= \int_0^t \mathbb{P}_i(N_s = 0|N_{0+} = k)\mathrm{d}s \\
&= \int_0^t \mathbb{P}\left(\sum_{j=i-k+1}^{i} \mathrm{E}_j \leqslant s\right)\mathrm{d}s \\
&= \int_0^t \sum_{j=i-k+1}^{i} c_{i-k+1,k-1,j} \int_0^s e^{-\mu_j u}\mathrm{d}u \ \mathrm{d}s \\
&= \sum_{j=i-k+1}^{i} c_{i-k+1,k-1,j} \int_0^t \psi_j(s)\mathrm{d}s \\
&= t - \sum_{j=i-k+1}^{i} \frac{c_{i-k+1,k-1,j}}{\mu_j}\psi_j(t) \qquad (5.9)
\end{aligned}$$

with

$$\psi_j(t) := \int_0^t e^{-\mu_j s}\mathrm{d}s = \frac{1-e^{-\mu_j t}}{\mu_j}.$$

And for $g_{ki}(t)$ we get

$$\begin{aligned}
g_{ki}(t) &:= \int_0^t \sum_{l=0}^{k-1}(k-l-1)\mathbb{P}_i(N_s = k-l|N_{0+} = k)\mathrm{d}s \\
&= \int_0^t \sum_{l=0}^{k-1}(k-l-1)\int_0^s \mathbb{P}\left(\sum_{j=i-k+1}^{i-k+\ell} E_j \in \mathrm{d}u\right)\mathbb{P}\left(E_{i-k+\ell+1} > s-u\right)\mathrm{d}u\mathrm{d}s \\
&= \int_0^t \sum_{l=0}^{k-1}(k-l-1)\int_0^s \varphi_{i-k+1,\ell-1}(u)e^{-\mu_{i-k+\ell+1}(s-u)}\mathrm{d}u\mathrm{d}s \\
&= \int_0^t \sum_{l=0}^{k-1}(k-l-1)\frac{\varphi_{i-k+1,\ell}(s)}{\mu_{i-k+\ell+1}}\mathrm{d}s \\
&= \sum_{\ell=0}^{k-1}(k-\ell-1)\sum_{j=i-k+1}^{i-k+\ell+1}\frac{c_{i-k+1,\ell,j}}{\mu_{i-k+\ell+1}}\psi_j(t) \text{ for } l = 0,...,k-1. \qquad (5.10)
\end{aligned}$$

Moreover, we define the transition probabilities, $p_{kl,i}(t) := \mathbb{P}(N_{t+} = l|N_{0+} = k)$, which can be calculated as follows

$$p_{k1,i}(t) = 1 - \sum_{l=2}^{k+1} p_{kl,i}(t), p_{kl,i}(t) = \frac{\varphi_{i-k+1,k-l+1}(t)}{\mu_{i-l+2}},$$

here $l = 2,...,k+1$. Then the dynamic program to identify the optimal strategy is defined as in Algorithm 4.

Algorithm 4 Dynamic program exponential heterogenous case

Let $f_{ki}(t), g_{ki}(t)$ and $p_{k\ell,i}(t)$ be defined as above. We can determine the $C_i(k)$ recursively: for $i = 1, \dots, n-1$ and $k = 1, \dots, i$,

$$C_i(k) = \inf_{t \geqslant 0} \left(\omega f_{ki}(t) + (1-\omega) g_{ki}(t) + \sum_{\ell=1}^{k+1} p_{k\ell,i}(t) C_{i+1}(\ell) \right),$$

whereas, for $k = 1, \dots, n$,

$$C_n(k) = (1-\omega) g_{kn}(\infty) = (1-\omega) \sum_{\ell=0}^{k-1} (k-\ell-1) \frac{1}{\mu_{n-k+\ell+1}}.$$

As for the homogeneous case the quantity $h_{ki} = \sum_{l=1}^{k-1} \frac{1}{\mu_{i-k+l}}$ can be used instead of g_{ki}.

5.2.3 Experiments single server dynamic scheduling

Lastly we will, on the basis of experiments, compare the costs of static scheduling with the costs of dynamic scheduling. We will do this for both the exponential homogeneous and heterogeneous case.

Experiment 5.6. cost of exponential homogeneous case
In this experiment the costs of the dynamic and static schedule are calculated for multiple values of n and ω, when $\mu = 1$ for all jobs, so in the case of homogeneous exponential service times. Let $K_{dyn}(n, \omega)$ be the value of the objective function for the given value of n and ω when the schedule is calculated dynamically and let $K_{stat}(n, \omega)$ be the costs of the static schedule. Moreover, let $r(n, \omega)$ be the ratio of $K_{dyn}(n, \omega)$ and $K_{stat}(n, \omega)$. The results are shown in Table 5. It can be observed that the dynamic schedule has lower costs in all cases and that the benefit of scheduling dynamic instead of static gets bigger for higher values for ω and n, so when idle time gets more important or there are more clients to be scheduled. Nearly the same values as in [1] are obtained.

n	ω	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
5	$K_{dyn}(n, \omega)$	0.94	1.36	1.58	1.67	1.65	1.54	1.34	1.04	0.61
	$K_{stat}(n, \omega)$	0.98	1.46	1.74	1.87	1.88	1.78	1.56	1.21	0.71
	$r(n, \omega)$	0.96	0.93	0.91	0.89	0.88	0.87	0.86	0.86	0.86
10	$K_{dyn}(n, \omega)$	2.13	3.09	3.62	3.85	3.85	3.64	3.21	2.55	1.60
	$K_{stat}(n, \omega)$	2.25	3.39	4.12	4.54	4.69	4.58	4.19	3.44	2.21
	$r(n, \omega)$	0.95	0.91	0.88	0.85	0.82	0.79	0.77	0.74	0.72
15	$K_{dyn}(n, \omega)$	3.32	4.83	5.66	6.03	6.05	5.73	5.08	4.07	2.57
	$K_{stat}(n, \omega)$	3.51	5.33	6.51	7.23	7.55	7.47	6.94	5.85	3.92
	$r(n, \omega)$	0.95	0.91	0.87	0.84	0.80	0.77	0.73	0.70	0.66
20	$K_{dyn}(n, \omega)$	4.51	6.56	7.70	8.21	8.25	7.83	6.96	5.58	3.54
	$K_{stat}(n, \omega)$	4.78	7.26	8.90	9.93	10.41	10.36	9.72	8.32	5.73
	$r(n, \omega)$	0.95	0.90	0.87	0.83	0.79	0.76	0.72	0.67	0.62
25	$K_{dyn}(n, \omega)$	5.70	8.29	9.74	10.40	10.45	9.92	8.82	7.10	4.51
	$K_{stat}(n, \omega)$	6.04	9.21	11.30	12.62	13.28	13.27	12.52	10.82	7.60
	$r(n, \omega)$	0.94	0.90	0.86	0.82	0.79	0.75	0.70	0.66	0.59
30	$K_{dyn}(n, \omega)$	6.89	10.02	11.77	12.59	12.64	12.02	10.70	8.61	5.48
	$K_{stat}(n, \omega)$	7.30	11.14	13.69	15.32	16.14	16.18	15.32	13.33	9.50
	$r(n, \omega)$	0.94	0.90	0.86	0.82	0.78	0.74	0.70	0.65	0.58

Table 5: Cost of dynamic and static schedule.

Experiment 5.7. cost of exponential heterogeneous case
In this experiment we do exactly the same as in Experiment 5.6. but now for the heterogeneous case. We have taken n equally spaced parameters in the interval $[0.5, 1.5]$, $\mu_i = 0.5 + \frac{i-1}{n-1}$ with

$i \in \{1, ..., n\}$. As you can see in Table 6, the costs grow more than linear in the number of clients. Furthermore, as for the homogeneous case, scheduling dynamic is better than scheduling static and nearly the same values as in [1] are obtained.

n	ω	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
5	$K_{dyn}(n, \omega)$	1.23	1.79	2.10	2.24	2.24	2.11	1.87	1.47	0.90
	$K_{stat}(n, \omega)$	1.32	2.01	2.43	2.65	2.70	2.58	2.28	1.79	1.07
	$r(n, \omega)$	0.93	0.89	0.87	0.85	0.83	0.82	0.82	0.82	0.85
10	$K_{dyn}(n, \omega)$	2.52	3.68	4.33	4.63	4.65	4.42	3.94	3.16	2.00
	$K_{stat}(n, \omega)$	2.71	4.16	5.13	5.73	6.00	5.94	5.51	4.60	3.01
	$r(n, \omega)$	0.93	0.88	0.84	0.81	0.78	0.74	0.71	0.69	0.67

Table 6: Costs of dynamic and static schedule for Experiment 5.7.

5.3 Adaptive model

In this subsection adaptive scheduling will be described, following the approach in [5]. The model is described for the phase-type case. The idea of this adaptive scheduling method is that you are able to reschedule by taking into account the state information. The adaptive approach is, as static scheduling, a simultaneous approach, since all patients are scheduled in one go instead of one-by-one. In [5] it is shown that adaptive scheduling results in less costs, which is definitely an advantage. Similarly as for the dynamic method, a disadvantage is that the clients do not have a fixed arrival time and therefore have to be available at all times.

The difference with static scheduling is that at every rescheduling epoch we are given the following state information:

- $k \in \{0, 1, .., n\}$, the number of clients who have already entered the system at time 0. For all clients that already have arrived, the arrival time is set equal to 0, so if $k > 0$, then $t_1 = ... = t_k = 0$.
- u, the value of the elapsed service time of the clients in service. If no client is in service, so $k = 0$, then we set $u = 0$. The remaining service time of the client in service is distributed as B_1 conditional on $B_1 > u$.
- n, the number of clients that remain to be served. Since k of them already entered the system, $n - k$ clients still need to be scheduled.

The same method as described in Subsection 5.1.2 will be used to compute the mean sojourn times. The differences with the static case are that now the vector $\boldsymbol{\gamma}$ is dependent on the value u, the elapsed service time, and if the value $k > 0$, $t_1 = ... = t_k = 0$. In both the case of the Erlang distribution and the hyperexponential distribution, the distribution of B_i, conditional on $B_i > u$, for some $u > 0$ is still a phase-type distribution with the same $\boldsymbol{T}_i$ as the one of B_i, but with a different initial distribution, which now depends on the elapsed service time u. This on u dependent initial distribution is denoted by $\boldsymbol{\gamma}_i(u)$. Recall that the process $\{\boldsymbol{X}_i(t)\}_{t\geqslant 0}$ for $i = 1, ..., n$ denotes the $(d_i + 1)$-dimensional continuous-time Markov chain corresponding to $B_i \sim \mathrm{PH}_{d_i}(\boldsymbol{\gamma}_i, \boldsymbol{T}_i)$. Our objective is then to find an expression for the j-th entry of $\boldsymbol{\gamma}_i(u)$, also denoted by $\boldsymbol{\gamma}_{ij}(u)$. This expression can be interpreted as $\mathbb{P}(\boldsymbol{X}_{u,i} = j | B_i > u)$. We first consider the case where B_i is mixed Erlang distributed with the parameters K_i, μ_i and p_i as described in Appendix A, then $\gamma_{ij}(u) = \frac{\gamma^\circ_{ij}(u)}{\gamma^\circ_i(u)}$ where

$$\gamma^\circ_i(u) := \mathbb{P}(B_i > u) = \sum_{j=1}^{K_i} \left(e^{-\mu_i u} \frac{(\mu_i u)^{j-1}}{(j-1)!} \right) + (1 - p_i) e^{-\mu_i u} \frac{(\mu_i u)^{K_i}}{K_i!} \text{ and} \tag{5.11}$$

$$\begin{aligned} \gamma^\circ_{ij}(u) &:= \mathbb{P}(X_{u,i} = j, B_i > u) \\ &= e^{-\mu_i u} \frac{(\mu_i u)^{j-1}}{(j-1)!} \mathbb{1}_{\{j=1,\dots,K_i\}} + (1 - p_i) e^{-\mu_i u} \frac{(\mu_i u)^{K_i}}{K_i!} \mathbb{1}_{\{j=K_i+1\}}. \end{aligned} \tag{5.12}$$

Secondly consider the case that B_i is hyperexponential, with parameters $\mu_{i,1}$, $\mu_{i,2}$ and p_i. Again $\gamma_{ij}(u) = \frac{\gamma^\circ_{ij}(u)}{\gamma^\circ_i(u)}$ where

$$\gamma^\circ_i(u) := \mathbb{P}(B_i > u) = p_i e^{\mu_{i,1} u} + (1 - p_i) e^{-\mu_{i2} u} \text{ and} \tag{5.13}$$

$$\gamma^\circ_{ij}(u) := \mathbb{P}(X_{u,i} = j, B_i > u) = p_i e^{-\mu_{i1} u} \mathbb{1}_{\{j=1\}} + (1 - p_i) e^{-\mu_{i2} u} \mathbb{1}_{\{j=2\}} [1]. \tag{5.14}$$

Now exactly the same approach for computing the mean sojourn times can be used as described in Subsection 5.1.2, except that every u does not have to be equal to 0. Observe that, when using this method one can calculate the best schedule from that state onwards and that thus the costs that will be optimized are the costs of the future. So only the costs of the future, instead of the costs of the whole makespan, are evaluated. This means that it is hard to compare the costs with the costs of the static model, since the costs of the clients that have been served already are not taken into account for the costs of the adaptive model.

It is possible to periodically adapt the schedule [5], something which we will not look into in this Bachelor Final Project. In that same paper also a method is described on how to evaluate the cost of the adaptive schedule relying on Monte Carlo simulation.

5.3.1 Experiments single server adaptive scheduling

A function to calculate the optimal schedule given the state information as described above has been written. In this section optimal schedules are calculated for different values of k and u and so it will be shown that taking into account the value of k and u indeed has an influence on the optimal schedule.

Experiment 5.8. Exponential case
First some runs are performed for the exponential case. The number of clients is again taken as $n = 5$ and $\mathbb{E}[B_i] = \text{Var}[B_i] = 1$ for $i = 1, ..., 5$. The variables used per run and the results are shown in Table 7. The results contain the costs and the interarrival times x_i for $i = 1, ..., 4$, with $x_i = t_i - t_{i-1}$. As said before, in all cases t_0 is set equal to one.

Run	**k**	**u**	ω	**costs**	**x1**	**x2**	**x3**	**x4**
1	0	0	0.5	1.881	0.976	1.421	1.400	1.113
2	0	0	0.5	1.881	0.979	1.420	1.432	1.114
3	1	0.5	0.5	1.881	0.968	1.442	1.377	1.104
4	1	0.6	0.5	1.881	0.968	1.442	1.377	1.104
5	1	0.7	0.5	1.881	0.968	1.442	1.377	1.104
6	1	0.8	0.5	1.881	0.968	1.442	1.377	1.104
7	2	0.5	0.5	2.106	-	2.133	1.419	1.114
8	3	0.5	0.5	2.738	-	-	3.148	1.137
9	4	0.5	0.5	3.770	-	-	-	3.671

Table 7: Costs and interarrival times for adaptive scheduling Experiment 5.8.

The first run of Table 7 is executed with the function that makes static schedules for the phase-type case as used in Subsection 5.1.2. This one is added as a check, since the outcome should be the same as the outcome of the second run, for which the function that makes adaptive schedules for the phase-type case is used. For run 3 - 6 the same value for k, 1, is used and the value of u is changed. The costs and the interarrival times do not change. This is as expected, since due to the memorylessness property of the exponential distribution, there is no impact of the elapsed service time on the results. For run 3 and 7 up to and including 9 the value of u is kept the same and the value for k differs. As more people are in the system, we also have that the costs and the interarrival times are higher. The first interarrival time seems to be the one that is most impacted, but the other ones have also changed a bit. See for example run 7, where x_2 is increased the most compared to the values of run 3, but also x_3 and x_4 have changed. The costs for run $7, 8$ and 9 are higher than for the other runs. Most of this increase can be explained by the waiting time of the clients that are already in the schedule but are not served yet, so in run 8 this is about the waiting time of client 2 and 3.

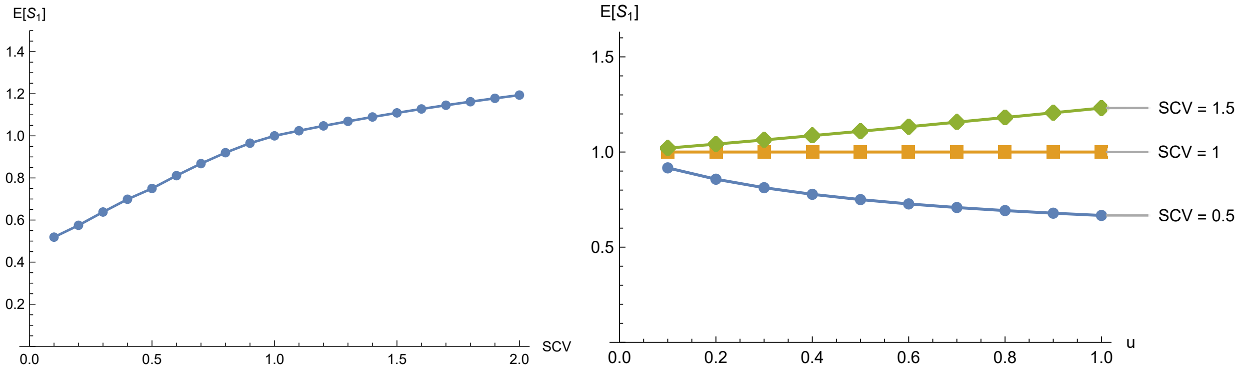
Experiment 5.9. Phase-type - Mixed Erlang and hyperexponential cases
Secondly we look at both the mixed Erlang and the hyperexponential case. For the mixed Erlang case the values $\mathbb{E}[B] = 1$ and $\text{Var}[B] = \frac{1}{2}$ are taken, such that the SCV < 1 and for the hyperexponential case the values $\mathbb{E}[B] = 1$ and $\text{Var}[B] = \frac{1}{2}$ are taken such that SCV > 1. The same values for k and u have been used for the different runs. The results of those runs can be seen in Table 8. Looking at runs 3 - 5, a difference is found between the mixed Erlang and the hyperexponential case. For the mixed Erlang case the costs and the first interarrival times decrease when u increases and for the hyperexponential case the costs and the interarrival times increase. This same difference has been observed when doing the experiments for other combinations of the mean and the variance. This difference can be explained as follows. When we are in the mixed Erlang case and u gets a higher value while the rest of the variables stay the same, so for example when we compare run 2 and 3, the $\boldsymbol{\gamma}$-vector is adapted such that the probability is higher that we are in

a later phase and thus the expected sojourn time of the client in service decreases. When we are in the hyperexponential case and u gets a higher value while the rest of the parameters stays the same, the $\boldsymbol{\gamma}$-vector is adapted such that the probability gets higher that the service time of the client in service is distributed with μ_2 instead of μ_1. By our definition, the probability p is always greater than or equal to 0.5 and thus $\mu_2 \leqslant \mu_1$, what results in $\mathbb{E}[B_1] \geqslant \mathbb{E}[B_2]$. So the higher the value of u, the higher we expect $\mathbb{E}[B]$ to be and thus the expected sojourn time of client 1 becomes longer.

				Mixed Erlang (SCV < 1)					**Hyperexponential** (SCV > 1)				
Run	**k**	**u**	ω	**costs**	**x1**	**x2**	**x3**	**x4**	**costs**	**x1**	**x2**	**x3**	**x4**
1	0	0	0.5	1.337	1.013	1.333	1.320	1.130	2.181	0.883	1.378	1.383	1.035
2	1	0.5	0.5	1.304	0.751	1.311	1.30	1.121	2.275	1.004	1.436	1.384	1.043
3	1	0.6	0.5	1.304	0.720	1.315	1.299	1.127	2.294	1.039	1.451	1.365	1.045
4	1	0.7	0.5	1.299	0.697	1.317	1.278	1.138	2.312	1.062	1.457	1.393	1.048
5	1	0.8	0.5	1.294	0.669	1.304	1.303	1.111	2.331	1.082	1.487	1.390	1.032
6	2	0.5	0.5	1.496	-	1.877	1.320	1.128	2.495	-	2.200	1.407	1.035
7	3	0.5	0.5	2.118	-	-	2.884	1.146	3.099	-	-	3.125	1.067
8	4	0.5	0.5	3.170	-	-	-	3.569	4.080	-	-	-	3.545

Table 8: Costs and interarrival times for adaptive scheduling Experiment 5.9.

Experiment 5.10. Different SCV's and values of u
Due to the results found in the previous experiment plots have been made to show them more clearly. Firstly in Figure 11a the sojourn time of the first client has been plotted for different values of the SCV, while $u = 0.5$ and $\mathbb{E}[B_1] = 1$. From this plot one can observe that the higher the SCV, the higher the expected sojourn time of the first client. The inflection point at SCV$= 1$ is caused by the transition from the mixed Erlang to the hyperexponential distribution. Secondly, in Figure 11, the influence of u in the case of different SCV is plotted. It can be observed that, as also observed in Experiment 5.9., the expected sojourn time decreases when SCV < 1, stays the same when SCV $= 1$ and increases when SCV> 1.



(a) $\mathbb{E}[S_1]$ for different SCV's.

(b) $\mathbb{E}[S_1 - u|S_1 > u]$ for different u's.

Figure 11: $\mathbb{E}[S_1 - u|S_1 > u]$ Experiment 5.10.

Lastly, also a general expression of the sojourn time of the first client under the condition that already u time of his service time has elapsed, $\mathbb{E}[S_1 - u|S_1 > u]$, has been calculated for the mixed Erlang and the hyperexponential case. As stated in Algorithm 2, the expected sojourn time of the first calculated equals $\mathbb{E}[S_i] = -\boldsymbol{\gamma}_1\boldsymbol{T}_1^{-1}\mathbf{1}_{D_i}$. So in the adaptive case, we get $\mathbb{E}[S_1 - u|S_1 > u] = -\boldsymbol{\gamma}_1(u)\boldsymbol{T}_1^{-1}\mathbf{1}_{D_i}$ In the mixed Erlang case $\boldsymbol{\gamma}_1(u)$ is given by Equation (5.11) and Equation (5.12) and

$$\boldsymbol{T} = \begin{pmatrix} -\mu & \mu & 0 & \cdot & \cdot & 0 \\ 0 & -\mu & \mu & & & \cdot \\ \cdot & & \cdot & \cdot & & \cdot \\ \cdot & & & \cdot & \mu & 0 \\ \cdot & & & & -\mu & \mu q \\ 0 & \cdot & \cdot & \cdot & 0 & -\mu \end{pmatrix} \text{ so } \boldsymbol{T}^{-1} = \begin{pmatrix} -\frac{1}{\mu} & -\frac{1}{\mu} & \cdot & \cdot & -\frac{1}{\mu} & -\frac{q}{\mu} \\ 0 & -\frac{1}{\mu} & -\frac{1}{\mu} & \cdot & -\frac{1}{\mu} & \cdot \\ \cdot & & \cdot & \cdot & \cdot & \cdot \\ \cdot & & 0 & -\frac{1}{\mu} & -\frac{1}{\mu} & \cdot \\ \cdot & & & & -\frac{1}{\mu} & -\frac{q}{\mu} \\ 0 & \cdot & \cdot & \cdot & 0 & -\frac{1}{\mu} \end{pmatrix}.$$

This results in

$$\begin{aligned} \mathbb{E}[S_1 - u | S_1 > u] &= -\left(\sum_{v=1}^{K_1} \left(\sum_{j=1}^{K_1+1} (-\frac{1}{\mu} \boldsymbol{\gamma}_{1j}(u) \mathbb{1}_{\{j \leqslant v\}} + 0 \cdot \boldsymbol{\gamma}_{1j}(u) \mathbb{1}_{\{j > v\}}) \right) + \sum_{j=1}^{K_1} -\frac{q}{\mu} \boldsymbol{\gamma}_{1j}(u) + -\frac{1}{\mu} \boldsymbol{\gamma}_{1,K_1+1}(u) \right) \\ &= -\left(\sum_{v=1}^{K_1} \left(\sum_{j=1}^{K_1+1} -\frac{1}{\mu} \boldsymbol{\gamma}_{1j}(u) \mathbb{1}_{\{j \leqslant v\}} \right) + \sum_{j=1}^{K_1} -\frac{q}{\mu} \boldsymbol{\gamma}_{1j}(u) - \frac{1}{\mu} \boldsymbol{\gamma}_{1,K_1+1}(u) \right) \\ &= \left(\sum_{v=1}^{K_1} \left(\sum_{j=1}^{K_1+1} \frac{1}{\mu} \boldsymbol{\gamma}_{1j}(u) \mathbb{1}_{\{j \leqslant v\}} \right) + \sum_{j=1}^{K_1} -\frac{q}{\mu} \boldsymbol{\gamma}_{1j}(u) - \frac{1}{\mu} \boldsymbol{\gamma}_{1,K_1+1}(u) \right). \end{aligned} \tag{5.15}$$

As, due to Equation (5.11) and Equation (5.12), the value of $\boldsymbol{\gamma}_{1,j}(u)$ decreases for small j when u increases and increases for larger j when u increases, one can relatively easy observe that Equation (5.15) increases when the elapsed service time of client 1, u, becomes bigger, since then the positive parts in this equation become smaller and the negative parts become bigger. In the hyperexponential case $\boldsymbol{\gamma}_1(u)$ is determined by Equation (5.13) and Equation (5.14), resulting in

$$\boldsymbol{\gamma}_1(u) = \left(\tfrac{p_1 e^{-\mu_{1,1} u}}{p_1 e^{-\mu_{1,1} u} + (1-p_1) e^{-\mu_{1,2} u}}, \tfrac{(1-p_1) e^{-\mu_{1,2} u}}{p_1 e^{-\mu_{1,1} u} + (1-p_1) e^{-\mu_{1,2} u}} \right).$$

Moreover, for $\boldsymbol{T}$ we have that $\boldsymbol{T} = \begin{pmatrix} -\mu_1 & 0 \\ 0 & -\mu_2 \end{pmatrix}$ and thus $\boldsymbol{T}^{-1} = \begin{pmatrix} -\frac{1}{\mu_1} & 0 \\ 0 & -\frac{1}{\mu_2} \end{pmatrix}$. Combining this in $\mathbb{E}[S_1 - u | S_1 > u] = -\boldsymbol{\gamma}_1(u) \boldsymbol{T}_1^{-1} \mathbf{1}_{D_i}$ gives

$$\begin{aligned} \mathbb{E}[S_1 - u | S_1 > u] &= -\left(\frac{-p_1 e^{-\mu_{1,1} u}}{\mu_{1,1}(p_1 e^{-\mu_{1,1} u} + (1-p_1) e^{-\mu_{1,2} u})} + \frac{-(1-p_1) e^{-\mu_{1,2} u}}{\mu_{1,2}(p_1 e^{-\mu_{1,1} u} + (1-p_1) e^{-\mu_{1,2} u})} \right) \\ &= \frac{\mu_{1,2} p_1 e^{-\mu_{1,1} u} + \mu_{1,1}(1-p_1) e^{-\mu_{1,2} u}}{\mu_{1,1}\mu_{1,2}(p_1 e^{-\mu_{1,1} u} + (1-p_1) e^{-\mu_{1,2} u})}. \end{aligned} \tag{5.16}$$

This function is plotted for different SCV values, and thus different $\mu_{1,2}, \mu_{1,2}$ and p_1 in the following figure.

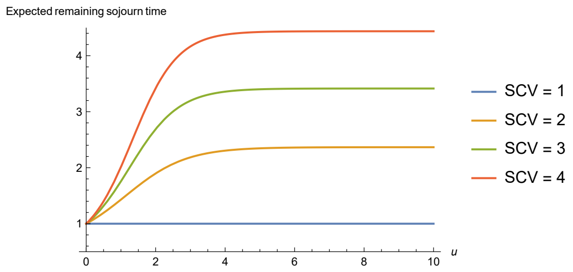


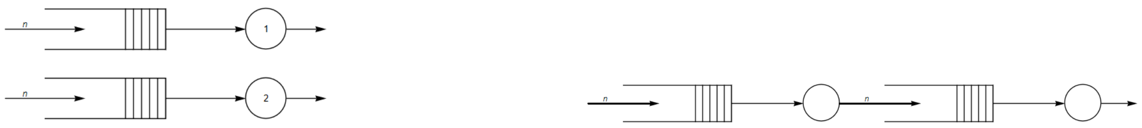
Figure 12: $\mathbb{E}[S_1 - u | S_1 > u]$ hyperexponential case.

From this figure it can be observed that Equation (5.16) is a non-decreasing function when SCV $\geqslant 1$. So now we have derived mathematically the same result as we found in the experiments above and thus we can conclude that when the elapsed service time of the first client increases the remaining sojourn time of this client decreases in the case of a mixed Erlang distributed service time and increases in the case of a hyperexponential distributed service time.

6 Multi server model

Until now we have only looked at single server models. We will now also have a look at the multiserver case. Since there are multiple definitions of a multiple server setting, we have to define which one we are talking about. Therefore we first mention the different options briefly, to clarify the different possibilities, and then go in depth about the last option.

First there is the setting of having multiple servers that all have their own queue of clients, see Figure 13a. In healthcare you find this setting for example when multiple general practitioners all have their own set of patients. This setting is not that interesting since this is just a combination of single server settings, which are unconnected.



(a) Two servers with both n customers.

(b) Two servers in line.

Figure 13: Not in depth discussed multiple server settings.

Secondly there is the setting of one queue of clients that are all first served by one server and then by a next one, see Figure 13b. In healthcare you find this setting for example when patients first have an appointment with a doctor and then have a MRI scan afterwards. There are of course extended versions of this setting, with more than two of those servers in a row, for example if the patients have a follow-up appointment. The setting of two servers in tandem is discussed by Kuiper [22].

Thirdly we have the setting that will be discussed more in depth in this Bachelor Final Project. This is the setting in which multiple servers share the same queue of clients that need to be served, also called pooling, see Figure 14. You can find such a setting for example at the emergency room.

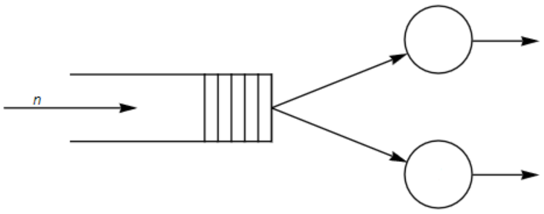


Figure 14: Illustration of two servers with together n customers.

6.1 Phase-type static multiserver

In this subsection the phase-type static case of the multiserver setting, using the third definition described above, will be discussed. We thus relax upon the first assumption from the model description; in this section it is possible that the number of servers, denoted by s, is greater than 1. We will first describe the general method in Subsection 6.1.1 and subsequently work out the expressions for a specific case, the case of $n = 3$, $s = 2$ and SCV > 1, in Subsection 6.1.3 to provide a better explanation of how the model is set up. The method that is used is the one described in [23] by Kuiper and Lee. As in that paper it is assumed that the servers and clients are homogeneous and without loss of generality, we normalize the mean service time to one for all clients, thus $\mathbb{E}[B_i] = \mu_i = 1$ for all i. Those two assumptions compress the state space.

6.1.1 Method

We will first have a look at how the evolution of the system can be described. Then we will describe the initial probability vector and transition matrix of the system, so when the first s clients have arrived, and subsequently we will describe the representation of the system after the arrival of the rest of the clients.

As in the single server setting, a bivariate process is used to describe the evolution of the system, $(Y_{s+i}(t), Z_{s+i}(t))$ for $i = 0, ..., n-s$. Here $Y_{s+i}(t) \in \{0, 1, ..., s+i\}$ clients are in the system as we start with s clients in service and $Z_{s+i} = (Z_1(t), ..., Z_\xi(t))$, with $\xi = \min\{Y_{s+i}(t), s\}$ for each $l = 1, ..., \xi$ and $Z_l(t) \in 1, ..., k$ denotes a vector of the phases of the jobs that are being processed at time t. Here, Z_l can be tought of as the l-th server because the servers are assumed to be homogeneous, k denotes the number of phases of the phase-type counterpart and there are at most ξ servers to keep track of, as either there are Y_{s+i} clients to be served or all s servers are active.

We define the probabilities of finding j clients in the system, with $j \in \{0, ..., s+i\}$ and the server(s) in phase(s) $m_l \in \{0, ..., k\}$ for $l \in \{1, ..., \xi\}$ as

$$\boldsymbol{p}^{(s+i)}_{j,(m_1,...,m_\xi)}(t) = \mathbb{P}[(Y_{s+i}(t), Z_{s+i}(t)) = (j, (m_1, ..., m_\xi))].$$

Moreover, define the row vector that contains all possible phases for j clients in service by

$$\boldsymbol{p}^{(s+i)}_{j}(t) = (\boldsymbol{p}^{(s+i)}_{j,(k,...,k)}(t), ..., \boldsymbol{p}^{(s+i)}_{j,(k,...,1)}(t), ..., \boldsymbol{p}^{(s+i)}_{j,(1,...,k)}(t), ..., \boldsymbol{p}^{(s+i)}_{j,(1,...,1)}(t)). \tag{6.1}$$

This vector has size $k^{\min\{j,s\}}$ since as at maximum s machines are serving customers.

The initial probability vector can be written as

$$\boldsymbol{\alpha}_s = (\boldsymbol{\alpha} \otimes ... \otimes \boldsymbol{\alpha}, 0_{\sum_{j=1}^{s-1} k^j}). \tag{6.2}$$

In the latter the Kronecker product is applied $s-1$ times. The first part, represented by $\boldsymbol{\alpha} \otimes ... \otimes \boldsymbol{\alpha}$, describes all options for s clients being active and the second part, represented by $0_{\sum_{j=1}^{s-1} k^j}$, has exactly the length of all options of having less than s clients. Those should all clearly be zero, as we assume that in the initial state all s servers start serving jobs immediately.

The initial transition matrix is then given by :

$$\boldsymbol{S}_s = \begin{pmatrix} \boldsymbol{S}^{(s)} & \boldsymbol{U}^{(s)} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{S}^{(s-1)} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \boldsymbol{U}^{(3)} & \boldsymbol{0} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{S}^{(2)} & \boldsymbol{U}^{(2)} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{S}^{(1)} \end{pmatrix}, \tag{6.3}$$

where the $\boldsymbol{S}^{(l)}$ are defined recursively

$$\boldsymbol{S}^{(l)} = \boldsymbol{I}_{|I|} \otimes \boldsymbol{S}^{(l-1)} + \boldsymbol{S} \otimes \boldsymbol{I}_{|S^{(l-1)}|}, \tag{6.4}$$

for $1 \leqslant l \leqslant s$ with $\boldsymbol{S}^{(1)} = \boldsymbol{S}$ and the $\boldsymbol{U}^{(l)}$ are defined recursively by

$$\boldsymbol{U}^{(l)} = \boldsymbol{I}_{|\boldsymbol{U}^{(l-1)}|} \otimes \boldsymbol{U}^{(1)} + \boldsymbol{U}^{(l-1)} \otimes \boldsymbol{I}_{|\boldsymbol{U}^{(1)}|}, \tag{6.5}$$

with $\boldsymbol{U}^{(1)} = -\boldsymbol{S}\boldsymbol{1}$. Here $|\boldsymbol{A}|$ denotes the number of rows in matrix $\boldsymbol{A}$, and thus $\boldsymbol{I}_{|\cdot|}$ denotes the identity matrix with $|\cdot|$ rows and columns. $\boldsymbol{U}^{(1)}$ is the phase-type exit vector that corresponds to service completion. In the transition matrix, the matrix $\boldsymbol{S}^{(l)}$ describes the transition between states in which l servers are busy and $\boldsymbol{U}^{(l)}$ denote the exit matrix that defines the transitions to only $l-1$ servers serving jobs.

The vector $\boldsymbol{p}^{(s)}(t)$, denoting the probability for all possible phases t time units after the arrival of the s-th client, is described by the phase-type distribution $\mathrm{PH}(\boldsymbol{\alpha}_s, \boldsymbol{S}_s)$:

$$\boldsymbol{p}^{(s)}(t) = (\boldsymbol{p}_s^{(s)}(t), \boldsymbol{p}_{s-1}^{(s)}(t), ..., \boldsymbol{p}_1^{(s)}(t)) = \boldsymbol{\alpha}_s \exp(\boldsymbol{S}_s t). \tag{6.6}$$

Now we will describe the phase-type representation of the system after arrival of all other clients. Again we need an initial probability vector $\boldsymbol{\alpha}_{(s+i)}$ and a transition matrix $\boldsymbol{S}_{(s+i)}$ for the phase-type representation $(\boldsymbol{\alpha}_{s+i}, \boldsymbol{S}_{s+i})$ for each client $i \in \{1, ..., n-s\}$ to keep track of the probabilities of the vector $\boldsymbol{p}^{(s+i)}(t)$, denoting the probabilities for all possible phases t time units after the arrival of the $(s+i)$-th client. This vector equals

$$\boldsymbol{p}^{(s+i)}(t) = (\boldsymbol{p}_{s+i}^{(s+i)}(t), \boldsymbol{p}_{s+i}^{(s+i-1)}(t), ..., \boldsymbol{p}_{s+i}^{(s+1)}(t), \boldsymbol{p}_{s+i}^{(s)}(t), \boldsymbol{p}_{s+i}^{(s+-1)}(t), \boldsymbol{p}_{s+i}^{(1)}(t)), \tag{6.7}$$

for $i = 1, ..., n-s$. The transition matrix $\boldsymbol{S}_{s+i}$ is defined as

$$\boldsymbol{S}_{s+i} = \left(\begin{array}{ccccc|cc} \boldsymbol{S}^{(s)} & \boldsymbol{T}^{(s)} & 0 & \cdots & 0 & 0 & 0 \\ 0 & \boldsymbol{S}^{(s)} & \boldsymbol{T}^{(s)} & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \ddots & \ddots & 0 & & \\ \vdots & \ddots & \ddots & \boldsymbol{S}^{(s)} & \boldsymbol{T}^{(s)} & 0 & \\ 0 & 0 & \cdots & 0 & \boldsymbol{S}^{(s)} & \boldsymbol{T}^{(s)} & 0 \\ \hline 0 & & & & & \boldsymbol{S}_s & \end{array}\right) =: \left(\begin{array}{c|c} \boldsymbol{S}_i^{\text{wait}} & \boldsymbol{T}_s^{(s)} \\ \hline \boldsymbol{0} & \boldsymbol{S}_s \end{array}\right). \tag{6.8}$$

This matrix is constructed by extending $\boldsymbol{S}_s$ by i times adding $\boldsymbol{S}^{(s)}$ to the diagonal. Each added block $\boldsymbol{S}^{(s)}$ corresponds to states where one might find an additional client in the waiting queue. Furthermore, $\boldsymbol{T}^{(s)}$ is added on the upperdiagonal to describe the flow from one client being finished to the next one being served. It is therefore an exit matrix that describes transition form a system with $j \geqslant s$ clients in the system to one with precisely $j-1 \geqslant s$ clients in the system. The $\boldsymbol{T}^{(l)}$ are defined by the following recursion

$$\boldsymbol{T}^{(l)} = \boldsymbol{I}_{|\boldsymbol{T}^{(l-1)}|} \otimes \boldsymbol{T}^{(1)} + \boldsymbol{T}^{(l-1)} \otimes \boldsymbol{I}_{|\boldsymbol{T}^{(1)}|}, \tag{6.9}$$

with $\boldsymbol{T}^{(1)} = -\boldsymbol{S}\boldsymbol{1} \otimes \boldsymbol{\alpha}$. The to $\boldsymbol{S}_{s+i}$ corresponding $\boldsymbol{\alpha}_{s+i}$ are given by

$$\begin{aligned} \boldsymbol{\alpha}_{s+i} =& f\left(\boldsymbol{p}^{(s+i-1)}(x_{s+i}), \boldsymbol{\alpha}\right) \\ :=& \Big(\boldsymbol{p}_{s+i-1}^{(s+i-1)}(x_{s+i}), \ldots, \boldsymbol{p}_{s+1}^{(s+i-1)}(x_{s+i}), \boldsymbol{p}_s^{(s+i-1)}(x_{s+i}), \\ & \boldsymbol{\alpha} \otimes \boldsymbol{p}_{s-1}^{(s+i-1)}(x_{s+i}), \ldots, \boldsymbol{\alpha} \otimes \boldsymbol{p}_1^{(s+i-1)}(x_{s+i}), \\ & \boldsymbol{\alpha} \otimes p_0^{(s+i-1)}(x_{s+i})\Big), \end{aligned} \tag{6.10}$$

where the first i states correspond to saturation and the rest corresponds to the start of service of the new client. Moreover, $p_0^{(s+i)}(t) = 1 - \boldsymbol{p}^{(s+i)}(t)\boldsymbol{1}$ denotes the probability of being in the absorbing state of an empty system.

6.1.2 Evaluating the objective function

For evaluating the objective function we can just calculate the total waiting and idle time, so the objective function does not have to be rewritten as for all other models up to now. We first look at the waiting time, therefore we look at the probabilities that correspond to instances in which clients are waiting. This is the case when all s servers are busy, thus for client $i = 1, ..., n-s$ we have

$$\boldsymbol{p}_{wait}^{(s+i)}(t) = (\boldsymbol{p}_{s+i}^{(s+i)}(t), \boldsymbol{p}_{s+i-1}^{(s+i)}(t), ..., \boldsymbol{p}_{s+1}^{(s+1)}(t)).$$

Moreover, the initial vector is defined as $\boldsymbol{\alpha}_i^{wait} = \boldsymbol{p}_{wait}^{(s+i)}(0)$. Then the expected total waiting time can be calculated as

$$\sum_{i=s+1}^{n} \mathbb{E}W_i = \sum_{i=1}^{n-s} -\boldsymbol{\alpha}_i^{\text{wait}} \left(\boldsymbol{S}_i^{\text{wait}}\right)^{-1} \mathbf{1}. \tag{6.11}$$

For the expected idle time, define $F_{M_{s+i}}(t)$ as the cumulative distribution function of the makespan of finishing the first $s+i$ clients t time units after t_{s+i} with $i = 1, ..., n-s$. This can be expressed by

$$F_{M_{s+i}}(t) = p_0^{(s+i)}(t) = 1 - \boldsymbol{p}^{(s+i)}(t)\mathbf{1} = 1 - \boldsymbol{\alpha}_{s+i} \exp\left(\boldsymbol{S}_{s+i}t\right)\mathbf{1}, \tag{6.12}$$

so that $\mathbb{E}M_{s+i} = -\boldsymbol{\alpha}_{s+i}\boldsymbol{S}_{s+i}^{-1}$. As for the expected waiting time we are interested in the total expected idle time that equals

$$\mathbb{E}I^{(s)} = s\left(\mathbb{E}M_n + t_n\right) - \sum_{i=1}^{n} \mathbb{E}B_i, \tag{6.13}$$

since the total idle time is the total time that the system is available minus the time spent is the system. The latter is denoted by the arrival time of the last client plus the expected makespan of the last client after his arrival, multiplied by s, the amount of servers. Here it is assumed that all machines stay available until the last client is served.

6.1.3 Example multiserver $n = 3, s = 2$, SCV> 1

In this example we show what all the vectors and matrices look like and show why they are built up as described in the previous subsection. Since n and s differ by $1 > 0$, we can show with this example how the matrices from previous subsection develop when $i \geqslant s$. But by letting this difference only be 1 and taking SCV> 1, resulting in a hyperexponential distribution, the matrices do not become too large.

In our example the bivariate process is described by (Y_{2+i}, Z_{2+i}) for $i = 0, 1$. $Y_{2+i} \in 0, ..., 3$ and $Z_{2+i} = (Z_1(t), ..., Z_\xi(t))$, where $\xi = \min\{Y_{s+i}(t), 2\}$ for each $l = 1, ..., \xi$, $Z_l(t) \in 1, 2$.

Moreover, $j \in \{0, ..., 3\}$, $m_l \in \{0, 1, 2\}$ for $l \in \{1, ..., \xi\}$. So the row vector that contains all possible phases for 2 clients t time units after the third arrival is described by the vector

$$\boldsymbol{p}_2^{(3)}(t) = (\boldsymbol{p}_{2,(2,2)}^{(3)}(t), \boldsymbol{p}_{2,(2,1)}^{(3)}(t), \boldsymbol{p}_{2,(1,2)}^{(3)}(t), \boldsymbol{p}_{2,(1,1)}^{(3)}(t)).$$

Since SCV> 1, a hyperexponential distribution is fitted and thus $\boldsymbol{\alpha} = (\mu_1, \mu_2)$. Using Equation (6.2) we get $\boldsymbol{\alpha}_1 = \boldsymbol{\alpha}$ and

$$\begin{aligned}\boldsymbol{\alpha}_s = \boldsymbol{\alpha}_2 &= (\boldsymbol{\alpha} \otimes \boldsymbol{\alpha}, \mathbf{0}_{\sum_{j=1}^{1} 2^j}) \\ &= (\boldsymbol{\alpha} \otimes \boldsymbol{\alpha}, \mathbf{0}_2) \\ &= (\mu_1^2, \mu_1\mu_2, \mu_2\mu_1, \mu_2^2, 0, 0).\end{aligned}$$

Since we assume that each server starts by serving a client, we start with 2 machines being active and thus there is a rate 0 for the positions that represent one machine being active serving a job that either has $\mathbb{E}[B_i] = \frac{1}{\mu_1}$ or $\mathbb{E}[B_i] = \frac{1}{\mu_2}$. The other positions represent all combinations of two machines serving jobs; both machines serving a job with $\mathbb{E}[B_i] = \frac{1}{\mu_1}$, the first machine serving a job with $\mathbb{E}[B_i] = \frac{1}{\mu_1}$ and the second machine serving a job with $\mathbb{E}[B_i] = \frac{1}{\mu_2}$, the other way around or both machines serving a job with $\mathbb{E}[B_i] = \frac{1}{\mu_1}$.

Due to the fitted hyperexponential distribution $\boldsymbol{S} = \begin{pmatrix} -\mu_1 & 0 \\ 0 & -\mu_2 \end{pmatrix}$, so by Equation (6.4)

$\boldsymbol{S}^{(1)} = \begin{pmatrix} -\mu_1 & 0 \\ 0 & -\mu_2 \end{pmatrix}$ and

$$\begin{aligned}\boldsymbol{S}^{(2)} &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} -\mu_1 & 0 \\ 0 & -\mu_2 \end{pmatrix} + \begin{pmatrix} -\mu_1 & 0 \\ 0 & -\mu_2 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} -2\mu_1 & 0 & 0 & 0 \\ 0 & -(\mu_1+\mu_2) & 0 & 0 \\ 0 & 0 & -(\mu_1+\mu_2) & 0 \\ 0 & 0 & 0 & -2\mu_2 \end{pmatrix}.\end{aligned}$$

This matrix can be explained by the fact that if there are two machines both serving a job with $\mathbb{E}[B_i] = \frac{1}{\mu_1}$, then the transition rate is $2\mu_1$ and thus $\boldsymbol{S}^{(2)}_{1,1} = -2\mu_1$ and if there is one machine serving a job with $\mathbb{E}[B_i] = \frac{1}{\mu_1}$ and the other machine is serving a job with $\mathbb{E}[B_i] = \frac{1}{\mu_2}$, then the transition rate is $2\mu_2$ and thus $\boldsymbol{S}^{(2)}_{1,1} = -(\mu_1+\mu_2)$ etc. By Equation (6.5) we get $\boldsymbol{U}^{(1)} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}$ and

$$\boldsymbol{U}^{(2)} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} + \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 2\mu_1 & 0 \\ \mu_2 & \mu_1 \\ \mu_2 & \mu_1 \\ 0 & 2\mu_2 \end{pmatrix}.$$

In the latter, the left columns represents the transitions from 2 machines being busy to only one machine being busy, serving a job with $\mathbb{E}[B_i] = \frac{1}{\mu_1}$ and the right column represents to transitions to only one machine being busy, serving a job with $\mathbb{E}[B_i] = \frac{1}{\mu_2}$. Combining $\boldsymbol{U}^{(2)}, \boldsymbol{S}^{(2)}$ and $\boldsymbol{S}^{(1)}$ we get by this we get by Equation (6.6)

$$\boldsymbol{S}^{2} = \begin{pmatrix} -2\mu_1 & 0 & 0 & 0 & 2\mu_1 & 0 \\ 0 & -(\mu_1+\mu_2) & 0 & 0 & \mu_2 & \mu_1 \\ 0 & 0 & -(\mu_1+\mu_2) & 0 & \mu_2 & \mu_2 \\ 0 & 0 & 0 & -2\mu_2 & 0 & 2\mu_2 \\ 0 & 0 & 0 & 0 & -\mu_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\mu_2 \end{pmatrix}.$$

By Equation (6.9) we find $\boldsymbol{T}^{1} = -\begin{pmatrix} -\mu_1 & 0 \\ 0 & -\mu_2 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} p & 1-p \end{pmatrix} = \begin{pmatrix} \mu_1 p & \mu_1(1-p) \\ \mu_2 p & \mu_2(1-p) \end{pmatrix}$ and

$$\begin{aligned}\boldsymbol{T}^{(2)} &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} \mu_1 p & \mu_1(1-p) \\ \mu_2 p & \mu_2(1-p) \end{pmatrix} + \begin{pmatrix} \mu_1 p & \mu_1(1-p) \\ \mu_2 p & \mu_2(1-p) \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 2\mu_1 p & (1-p)\mu_1 & (1-p)\mu_1 & 0 \\ \mu_2 p & \mu_1 p + (1-p)\mu_2 & 0 & (1-p)\mu_1 \\ \mu_2 p & 0 & \mu_1 p + (1-p)\mu_2 & (1-p)\mu_2 \\ 0 & p\mu_2 & p\mu_2 & 2(1-p)\mu_2 \end{pmatrix},\end{aligned}$$

describing the flow from one client being finished to the next one being served. Since $\boldsymbol{T}^{(2)}$ is the exit matrix from a state in which 2 clients are being served to another state in which there are 2 clients being served, this matrix is 4×4 as there are $2 \cdot 2 = 4$ possible states to come from or go to.

This all results in

$$\boldsymbol{S}_3 = \begin{pmatrix} -2\mu_1 & 0 & 0 & 0 & 2\mu_1 p & (1-p)\mu_1 & (1-p)\mu_1 & 0 & 0 & 0 \\ 0 & -(\mu_1+\mu_2) & 0 & 0 & \mu_2 p & \mu_1 p + (1-p)\mu_2 & 0 & (1-p)\mu_1 & 0 & 0 \\ 0 & 0 & -(\mu_1+\mu_2) & 0 & \mu_2 p & 0 & \mu_1 p + (1-p)\mu_2 & (1-p)\mu_2 & 0 & 0 \\ 0 & 0 & 0 & -2\mu_2 & 0 & p\mu_2 & p\mu_2 & 2(1-p)\mu_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2\mu_1 & 0 & 0 & 0 & 2\mu_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -(\mu_1+\mu_2) & 0 & 0 & \mu_2 & \mu_1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -(\mu_1+\mu_2) & 0 & \mu_2 & \mu_1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2\mu_2 & 0 & 2\mu_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\mu_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\mu_2 \end{pmatrix}.$$

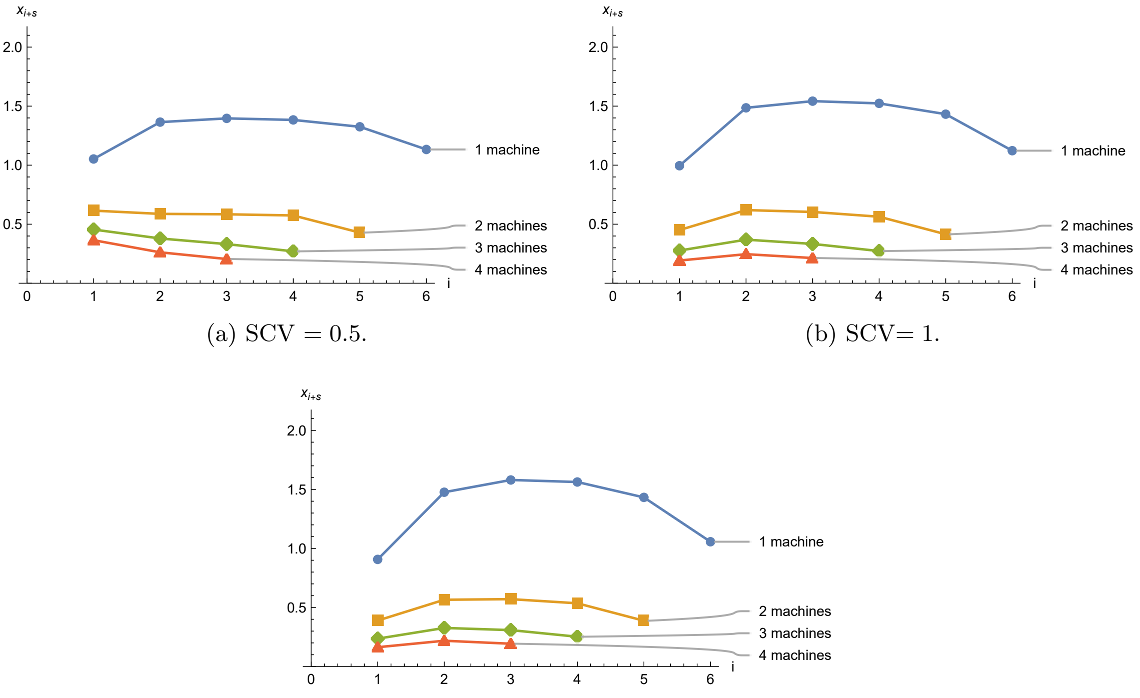
Lastly we show what the corresponding probability vector looks like:

$$\begin{aligned} \boldsymbol{\alpha}_3 = \boldsymbol{\alpha}_{2+1} &= \left(\boldsymbol{p}_2^{(2)}(x_3) \quad \boldsymbol{\alpha} \otimes \boldsymbol{p}_1^{(2)}(x_3) \quad \boldsymbol{\alpha} \otimes \boldsymbol{p}_0^{(2)}(x_3) \right) \\ &= \Big(\boldsymbol{p}_{2,(2,2)}^{(2)}(x_3), \boldsymbol{p}_{2,(2,1)}^{(2)}(x_3), \boldsymbol{p}_{2,(1,2)}^{(2)}(x_3), \boldsymbol{p}_{2,(1,1)}^{(2)}(x_3), \mu_1 \cdot \boldsymbol{p}_{1,(2)}^{(2)}(x_3), \mu_1 \cdot \boldsymbol{p}_{1,(1)}^{(2)}(x_3), \\ &\quad \mu_2 \cdot \boldsymbol{p}_{1,(2)}^{(2)}(x_3), \mu_2 \cdot \boldsymbol{p}_{1,(1)}^{(2)}(x_3), \mu_1 \cdot p_0^{(2)}(x_3), \mu_2 \cdot p_0^{(2)}(x_3) \Big). \end{aligned}$$

6.2 Experiment multiserver scheduling

In this subsection we discuss one experiment for the multiserver setting. Moreover, we briefly compare the results in the case of only one server and discuss the running time of the model.

Experiment 6.1. Interarrival times and costs
In this experiment we ran our function for $s = 1, 2, 3, 4$ and $n = 7$ for various SCV. The SCV values, 0.5, 1 and 1.5, are chosen such that we have a mixed Erlang, exponential and a hyperexponential fit.



(a) SCV = 0.5.

(b) SCV= 1.

(c) SCV = 1.5.

Figure 15: Interarrival times for $n = 7$ and $s = 1, 2, 3$ or 4.

In Figure 15 for the case $s = 1$ the domeshape is visible. The larger the number of machines the less amplified this pattern is. Moreover, this pattern gets more clear for higher values of the SCV, in which case there is more variation. In Figure 15a we observed a steep decline in the interarrival times and for $s = 2$ we also observe a pattern of iterative increasing and decreasing interarrival times. This structure is a so called *reversed bullwhip* [23]. In the results of [23] this pattern is better visible since they have plotted the interarrival times for more clients. The explanation they give for this pattern is that the synchronized start of service is completely absorbed by the randomness in the system if there are sufficient clients to be scheduled. In the beginning you have a relatively long time until the first client is being served, followed by a relatively short time until the second client is served. This pattern of alternating long and shorter interarrival times will proceed, but after some time it will disappear due to randomness.

s|SCV	**0.5**	**1**	**1.5**
1	2.10265	2.99426	3.56438
2	1.47120	1.89673	2.05757
3	1.53349	1.98602	2.27510
4	1.88130	2.54814	3.06381

Table 9: Costs $s = 1, 2, 3, 4$, $n = 7$ and SCV $= 0.5, 1, 1.5$.

In Table 9 we show the costs of the runs we have executed for Figure 15. As one can observe for all values of s the costs get higher when the SCV increases, this effect can easily be explained by the higher variability. The costs are the highest in the case of one server and the costs increase when having more servers. The latter can be explained by the fact that $2, 3$ and 4 are not divisors of seven which is the number of clients, and thus the more machines there are, the more machines that are idle at the end of the system, while the last job is being served. The fact that the costs are the highest when only 1 machine is used can be explained by the expected benefits of pooling. However, we can not investigate the benefits of pooling correctly right now. One would have probably expected the costs to be lower when the amount of machines is higher. This is currently not the case, as machines stay available until all jobs are served, so also when they will not get a job anymore. This increases the expected total idle time drastically. Therefore, to be able to investigate the benefits of pooling, one should adapt the model such that machines will no longer be available, and will thus not contribute to the idle time, when they can not get a job anymore. We will not go more in depth about this, but will refer to section 5.3 of [23].

A last important note is that running the model took a very long time, on a Lenovo Thinkpad P1 laptop with the following specifications: IIntel Core i7 8750H / 2.2 GHz. This is also the reason why we only ran situations in which at maximum five variables were to be optimized. The more machines the longer the running time, which can be explained by the fact that the matrices become bigger when there are more machines.

7 Including no-shows and walk-ins

To make the different models more applicable in practice, no-shows and walk-ins can be considered as well. No-shows are included since it is not realistic to assume that all patients that have made an appointment will actually show up. Walk-ins are included since appointment scheduling is used for healthcare and at the general practitioner and the dentist there are walk-ins. In this section we describe how no-shows can be dealt with in the case of static single server scheduling, dynamic single server scheduling and static multiserver scheduling. So in this section we relax upon assumption five and six described in Section 4. For the static single server scheduling we discuss three different methods. The first one is based on [10] and adapts the expected value, variance and thus SCV of the random variable. For this method it is also discussed how walk-ins can be included. The second one is thought of by ourselves and is based on modelling the random variable as a hyperexponential. This one methods works only for the exponential case. The last method that we have implemented is the one described in [24], which is again specifically for the exponential case. The method we use for including no shows in the model for dynamic single server scheduling is thought of ourselves, but is uses the method of [24] as a baseline. Lastly, for the multiserver setting we have used the method discussed in [23].

7.1 Adjusted expected service time

The method in [10] describes how the expected service times can be adapted if the probability of having a no-show or a walk-in is known. Since the expected value is different, also the variance and thus the SCV become different, resulting in an adapted schedule.

The original variable B_i for the service times had an expected value $\mathbb{E}[B_i]$. There is a probability, $q \in [0, 1)$, that a patient will not show up. Therefore the adjusted service time becomes $\bar{B}_i = B_i$ with probability $1 - q$ and $\bar{B}_i = 0$ with probability q, and thus $\mathbb{E}[\bar{B}_i] = \mathbb{E}[B_i](1 - q)$. The same method will be used to deal with walk-ins. Let $v \in [0, 1]$ be the probability that an unscheduled patient is added to an appointment slot (i.e. two patients take up the timeslot in which only one is scheduled). The adjusted service time in this case is $\bar{B}_i = B_i$ with probability $1 - v$ and $\bar{B}_i$ equal to i.i.d. copies of $\mathbb{E}[B_i]$ with probability v. Observe that the walk-in probability, v, is chosen to be a fixed number, despite it would be more realistic if this value was dependent on the service time B_i, as when the service time increases there is a longer timeframe and thus a higher probability that a walk-in can arrive. In conclusion, when we incorporate both no-shows and walk-ins, the revised service time $\bar{B}_i$ equals:

I. two i.i.d. copies of the service time B_i with probability $(1 - q)v$, when there is no no-show and a walk-in.

II. equal to one service time B_i with probability $(1 - q)(1 - v) + qv$, either when there is no no-show and no walk-in, or there is a no-show and a walk-in.

III. equal to 0 with probability $q(1 - v)$, when there is a no-show and no walk-in.

This results in the following new expressions.

$$\begin{aligned}\mathbb{E}[\bar{B}_i] &= 2(1-q)v\mathbb{E}[B_i] + ((1-q)(1-v)+qv)\mathbb{E}[B_i] = (1-q+v)\mathbb{E}[B_i].\\ \mathbb{E}[\bar{B}_i^2] &= (1-q)v\mathbb{E}[(B_i+B_i')^2] + ((1-q)(1-v)+qv)\mathbb{E}[B^2]\\ &= (1-q)v\mathbb{E}(2\mathbb{E}[B_i^2]+2(\mathbb{E}[B_i])^2) + ((1-q)(1-v)+qv)\mathbb{E}[B_i^2],\end{aligned}$$

where B_i' is an independent copy of B_i. For the variance we then get that

$$\begin{aligned}\mathrm{Var}[\bar{B}_i] &= (1-q)v\mathbb{E}(2\mathbb{E}[B_i^2]+2(\mathbb{E}[B_i])^2) + ((1-q)(1-v)+qv)\mathbb{E}[B^2] - (1-q+v)^2(\mathbb{E}[B_i])^2\\ &= (1-q+v)\mathbb{E}[B_i^2] - (v^2+(1-q)^2)(\mathbb{E}[B_i])^2.\end{aligned}$$

And finally for the adapted SCV we get

$$\begin{aligned}\text{S}\bar{\text{C}}\text{V}(q,v) &= \frac{(1-q+v)\mathbb{E}[B_1^2]-(v^2+(1-q)^2)(\mathbb{E}[B_i])^2}{(1-q+v)^2(\mathbb{E}[B_i])^2}\\ &= \frac{(1-q+v)\text{SCV}+q(1-q)+v(1-v)}{(1-q+v)^2(\mathbb{E}[B_i])^2}.\end{aligned}$$

It can easily be observed that when $q=0$ we are in the situation with only walk-ins and when $v=0$ we are in the situation with only no-shows.

Lastly we also have to adapt the objective function. Equation (4.6) changes consequently to

$$\omega\left(\sum_{i=1}^{n}x_i+\mathbb{E}[S_n]-\sum_{i=1}^{n}(1-q+v)\mathbb{E}[B_i]\right)+(1-\omega)\sum_{i=1}^{n}\left(\mathbb{E}[S_i]-(1-q+v)\mathbb{E}[B_i]\right). \tag{7.1}$$

7.2 Extended exponential distribution to hyperexponential distribution

In the previous method an adjusted expected value, variance and SCV for all the clients is used to make the schedule. We now propose an alternative method, that is generally applicable, but we will for illustration purposes describe it only for the case that service times of all clients are exponentially distributed. We include both no-shows and the walk-ins, but not at the same time, by extending an exponential distribution to an hyperexponential distribution. For the case of no-shows it is assumed that a patient has a probability q of not showing up. The modelled hyperexponential service time $\bar{B}_i$ is therefore equal to 0 with probability q and equal to $\mathbb{E}[B_i]$ with probability $1-q$. So with probability q the service time is distributed with $\mu_1=\infty$, such that the expected service time is 0 and with probability $(1-q)$ distributed with $\mu_2=\mu$. In the case of walk-ins it is assumed that there is a probability v that two clients arrive instead of one. The modelled hyperexponential service time $\bar{B}_i$ is therefore equal to $2\mathbb{E}[B_i]$ with probability v and equal to $\mathbb{E}[B_i]$ with probability $1-v$.

The difference between this and the previous model is that a complete new phase-type fit is performed for $\bar{B}_i$, therefore the phase-type fit in the first model can still result in a mixed Erlang distribution. Note that when implementing this method for taking into account no-shows or walk-ins the objective function should be adapted. To be exact, Equation (4.6) should be changed respectively to either

$$\omega\left(\sum_{i=1}^{n}x_i+\mathbb{E}[S_n]-\sum_{i=1}^{n}(1-q)\mathbb{E}[B_i]\right)+(1-\omega)\sum_{i=1}^{n}\left(\mathbb{E}[S_i]-(1-q)\mathbb{E}[B_i]\right)$$

or

$$\omega\left(\sum_{i=1}^{n}x_i+\mathbb{E}[S_n]-\sum_{i=1}^{n}(1+v)\mathbb{E}[B_i]\right)+(1-\omega)\sum_{i=1}^{n}\left(\mathbb{E}[S_i]-(1+v)\mathbb{E}[B_i]\right),$$

due to the expected value for the new stochast being either

$$\mathbb{E}[\bar{B}_i]=(1-q)\mathbb{E}[B_i]+q\cdot 0=\frac{1-q}{\mu}$$

or

$$\mathbb{E}[\bar{B}_i]=(1-v)\mathbb{E}[B_i]+v\cdot 2\mathbb{E}[B_i]=\frac{1+v}{\mu}.$$

7.3 Homogeneous exponential case

The third method for including no-shows for static single server scheduling is specifically for the exponential homogeneous case as discussed in Subsection 5.1.1. It is the method as described in [24]. This method works basically the same as the previous one, as both model $\bar{B}_i$ being 0 with probability q and being $\mathbb{E}[B_i]$ with probability $1-q$. Let q be the probability that there is a no-show, then the formulas from Subsection 5.1.1 change to

$$\begin{aligned}\mathbb{P}(N(t_i)=j) = (1-q)\cdot &\sum_{k=0}^{i-j-1} \frac{(\mu x_{i-1})^k}{k!} e^{-\mu x_{i-1}} \mathbb{P}(N(t_{i-1})=j+k-1) \\ &+ q\cdot \sum_{k=0}^{i-j-2} \frac{(\mu x_{i-1})^k}{k!} e^{-\mu x_{i-1}} \mathbb{P}(N(t_{i-1})=j+k) \text{ with } j>0, i\geqslant 2.\end{aligned}$$

and

$$\begin{aligned}\mathbb{P}(N(t_i)=0) &= (1-q)\cdot \sum_{k=1}^{i-1} \mathbb{P}(N(t_{i-1})=k-1) \sum_{l=k}^{\infty} \frac{(\mu x_{i-1})^l e^{-\mu x_{i-1}}}{l!} \\ &+ q\cdot \sum_{k=0}^{i-2} \mathbb{P}(N(t_{i-1})=k) \sum_{l=k}^{\infty} \frac{(\mu x_{i-1})^l e^{-\mu x_{i-1}}}{l!} \\ &= (1-q)\cdot \sum_{k=1}^{i-1} \mathbb{P}(N(t_{i-1})=k-1) \left(1-\sum_{l=0}^{k} \frac{(\mu x_{i-1})^l e^{-\mu x_{i-1}}}{l!}\right) \\ &+ q\cdot \sum_{k=0}^{i-2} \mathbb{P}(N(t_{i-1})=k) \left(1-\sum_{l=0}^{k} \frac{(\mu x_{i-1})^l e^{-\mu x_{i-1}}}{l!}\right) \quad \text{i} \geqslant 2.\end{aligned}$$

When implementing this method for taking no-shows into account the objective function, Equation (5.1), should be changed to

$$\omega \sum_{i=1}^{n} \left(x_{i-1} + \mathbb{E}[W_i] - \left(\mathbb{E}[W_{i-1}] + (1-q)\cdot \frac{1}{\mu} \right) \right) + (1-\omega) \sum_{i=1}^{n} \mathbb{E}[W_i].$$

7.4 Including no-shows in dynamic model

As described in the concluding remarks of [1] it could be interesting to include no-shows in the dynamic model that is described in that paper. The method that they propose is the one that uses an adapted SCV, mean and variance, as described in Subsection 7.1. This method is only applicable to the phase-type case that we have not studied in this Bachelor Final Project for the dynamic approach. In this subsection we propose another method to include no-shows in the dynamic model for the exponential case. This method uses adapted versions of the formulas described in Subsection 5.2.1.

In the dynamic approach client $n+1$ is scheduled at the moment that the n-th client has arrived. When including no-shows, it is not certain that a client arrives, so this definition of the dynamic approach has to change to that the $(n+1)$-st client is scheduled at the moment that the n-th client should arrive. This change of definition can be done since clients are assumed to be punctual so at the moment that client n is scheduled you know whether it is a no-show or not. In Subsection 5.2.1 $\omega f_k(t)$ is said to be the contribution of the idle time to the cost function, due to the interval $[0,t]$ where k is the number of clients in the system immediately after the arrival of client i. This definition has to change, since we are not sure whether client i actually arrives. Therefore the contribution of the idle time to the cost function due to the interval $[0,t]$ become $\omega((1-q)\cdot f_k(t) + q\cdot f_{k-1}(t))$, since if client i shows up there are indeed k clients and if client i is a no-show there are only $k-1$ clients after the scheduled arrival time of client i, with again q denoting the no-show probability. The same argument is used to rewrite the contribution to the waiting time due to $[0,t]$ from $(1-\omega)g_k(t)$ to $(1-\omega)\cdot(1-q)g_k(t) + (1-\omega)\cdot q\cdot g_{k-1}(t)$. Lastly also the transition probabilities are rewritten, they become:

$$p_{k1}(t) = (1-q)\cdot \sum_{m=k}^{\infty} e^{-\mu t}\frac{(\mu t)^m}{m!} + q\cdot \sum_{m=k-1}^{\infty} e^{-\mu t}\frac{(\mu t)^m}{m!},$$

$$p_{kl}(t) = (1-q) \cdot e^{-\mu t} \frac{(\mu t)^{k-l+1}}{(k-l+1)!} + q \cdot e^{-\mu t} \frac{(\mu t)^{k-1-l+1}}{(k-1-l+1)!},$$

for $k = 1, ..., i$ and $l = 2, ..., k+1$ and $t \geqslant 0$. The objective function does not have to be changed additionally, as all terms that it contains are already adapted.

7.5 Including no-shows in multiserver model

Lastly we look at the method of including no-shows in the multiserver as discussed in [23]. It uses the same approach as Subsection 7.2 and Subsection 7.3, but is now, of course, applied to the multiserver model as discussed in Subsection 6.1.

As before, q is the no-show probability, thus with probability q the service time of a client equals 0 and with probability $1-q$ the service time of client is approximated by a phase-type distribution. To include this in our model, we only need to adapt the initial probability vectors. This is done as follows. Define

$$\boldsymbol{\alpha}^q_{s,j} = (\boldsymbol{\alpha} \otimes ... \otimes \boldsymbol{\alpha})(1-q)^j q^{(s-j)} \binom{s}{j}, \tag{7.2}$$

where the Kronecker product is applied $j-1$ times. This gives the part of the adapted initial probability vector representing all states in which there are $s-j$ clients who showed up and thus j no-shows. When we combine this quantity for all possible values for j, so for 1 up to and including s, we get the new initial probability vector:

$$\boldsymbol{\alpha}^q_s = (\boldsymbol{\alpha}^q_{s,s}, \boldsymbol{\alpha}^q_{s,s-1}, ..., \boldsymbol{\alpha}^q_{s,1}). \tag{7.3}$$

Moreover, when we take no-shows into account for subsequent clients, Equation (6.10) changes to

$$\boldsymbol{\alpha}^q_{s+i} = (1-q)(f(\boldsymbol{p}^{(s+i-1)}(x_{s+i}), \boldsymbol{\alpha})) + q(\mathbf{0}_{k^s}, \boldsymbol{p}^{(s+i-1)}(x_{s+i})), \tag{7.4}$$

since with probability $1-q$ there is indeed a new client and with probability q there is not. Lastly, we should also adapt the objective function. The formula for the expected waiting time, Equation (6.11), can stay the same, but the formula for the expected idle time, Equation (6.13), changes to

$$\mathbb{E}[I^{(s)}] = s\,(\mathbb{E}[M_n] + t_n) - (1-q)\sum_{i=1}^{n} \mathbb{E}[B_i], \tag{7.5}$$

since a client shows up with probability $1-q$.

7.6 Experiments including no-shows and walk-ins

As for the other chapters, we end this one with some experiments. We will discuss all implementations of no-shows as described above and compare them when possible. Moreover, we will study in Experiment 7.5. whether including no-shows is really necessary.

Experiment 7.1. Analyzing the impact of no-shows and walk-ins
We start with analyzing the impact of no-shows and walk-ins on the interarrival times and the costs, using the method of including them in our model as described in Subsection 7.1. This is implemented by first calculating the adapted mean, variance and SCV values and then applying the phase-type algorithm for finding the schedule as described in Subsection 5.1.2 with those adjusted SCV's. The results are shown in Table 10 and Figure 16.

Run	**q**	**v**	ω	**costs**
1	0	0	0.5	1.88126
2	0.5	0	0.5	1.34134
3	0.5	0.5	0.5	2.18146
4	0	0.5	0.5	2.50886

Table 10: Costs of runs.

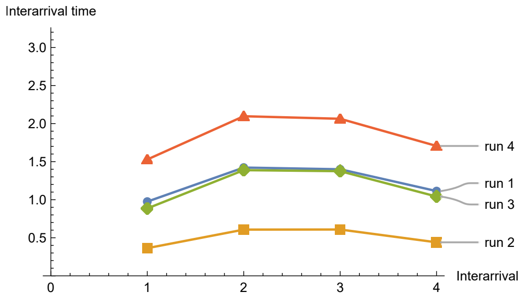


Figure 16: interarrival times.

one can observe that the first run and the corresponding results correspond to the results of the static case with $\mathbb{E}[B_i] = 1$ in Subsection 5.1.3, see Figure 6a. As can be seen from run 2 and run 4, the interarrival times decrease when the no-shows are taken into account and they increase when walk-ins are taken into account. The costs are lower when no-shows are included, due to the expected idle times being lower. The costs are higher when walk-ins are included, due to both the expected idle times and expected waiting times being higher. For run 1 and 3 the interarrival times are nearly the same. This can be explained by the fact that the expected value of the service times of all clients is the same in those runs, since in the third run q and v cancel each other out. However, the costs for run 3 are higher than for run 1, due to the variability being higher.

Experiment 7.2. Adjusted expected service time
Secondly we study the method that uses an adjusted expected service time more in depth and we study the impact of differing the values for q and v. We calculate the costs and interarrival times for the schedules, that are again found with the method described in Subsection 7.1. The input values used for the different runs can be read in Table 12.

As can be seen in Table 12, when the value of q is increased, and thus when the expected value of the service times decreases, the interarrival times decrease. So an increase of the no-show probability has the same effect as decreasing the expected value. When the value of v is increased, and thus when the expected value of the service times increases, the interarrival times increase. So an increase of the walk-in probability has the same effect as increasing the expected value.

Run	q	v	ω	costs
1	0	0	0.5	1.88126
2	0.2	0	0.5	1.74516
3	0.4	0	0.5	1.50643
4	0.6	0	0.5	1.14177
5	0.8	0	0.5	0.63502
6	0	0	0.5	1.88126
7	0	0.2	0.5	2.20311
8	0	0.4	0.5	2.42768
9	0	0.6	0.5	2.57188
10	0	0.8	0.5	2.65011

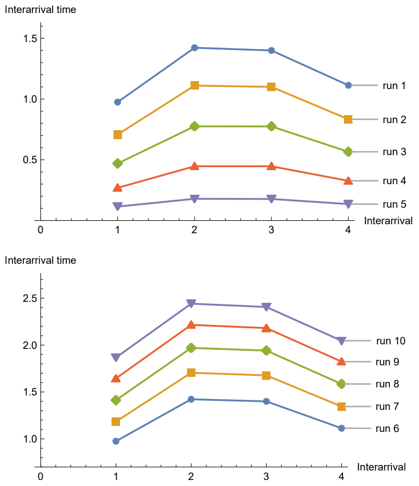


Table 12: Costs and interarrival times including no-shows and walk-ins n = 5.

Experiment 7.3. Extended exponential distribution to hyperexponential distribution Now we calculate the costs and interarrival times of the optimal schedules using the second method of including no-shows and walk-ins, as described in Subsection 7.2. The chosen parameters for the different runs and the results can be found in Table 14. For all runs n equals five.

Run	q	v	ω	costs
1	0	0	0.5	1.88
2	0.2	0	0.5	1.86
3	0.4	0	0.5	1.73
4	0.6	0	0.5	1.46
5	0.8	0	0.5	0.94
6	0	0	0.5	1.88
7	0	0.2	0.5	2.42
8	0	0.4	0.5	2.87
9	0	0.6	0.5	3.24
10	0	0.8	0.5	3.53

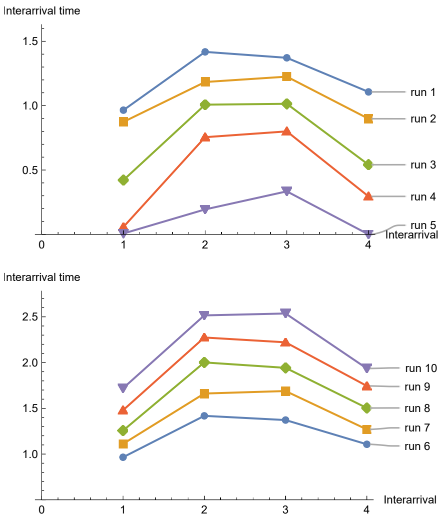


Table 14: Costs and interarrival times including no-shows and walk-ins $n = 5$.

As for the first method the interarrival times and the costs decrease when q is increased and they increase when v is increased. When q increases the costs seem to not decrease linearly, but more exponentially. When comparing the costs from method 1 and 2 differences are found. In Table 15 the costs of both methods and their ratio are presented for different no-show probabilities. For all runs $n = 5$ and $\omega = 0.5$ are used. A well-founded explanation for the big difference between the costs of method 1 and 2 has not been found, for now, we assume it can be explained by the two different methods just having a different impact.

q	**0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
costs method 1	1.88	1.82	1.75	1.64	1.51	1.34	1.14	0.91	0.64	0.33
costs method 2	1.88	1.87	1.86	1.80	1.74	1.61	1.47	1.28	1.04	0.91
ratio	1	0.97	0.94	0.91	0.87	0.83	0.78	0.71	0.62	0.36

Table 15: Comparison of costs of method 1 and 2.

Experiment 7.4. Homogeneous exponential case

In this experiment the costs and interarrival times for scheduling $n = 5$ clients for different no-show probabilities are calculated using the third method that we have described for including no-shows, that included no-shows in the exponential homogeneous static model described in Subsection 5.1.1, see Subsection 7.3. Again $\omega = 0.5$ is used for all runs. The results can be found in Table 16 and Figure 17.

Run	**q**	**costs**
1	0	1.88
2	0.2	1.85
3	0.4	1.72
4	0.6	1.46
56	0.8	0.94

Table 16: Costs of runs.

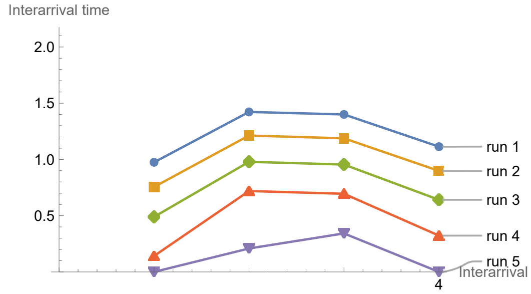


Figure 17: Interarrival times.

As for the other methods the costs and the interarrival times decrease when the no-show probability increases. The most interesting observation is that the costs (and the interarrival times) coincide with the costs of the second method (see Table 14). On the other hand it is not that interesting as the similarity can easily be explained by the fact that in method 2 and 3 we schedule clients that all still have the original expected service time, but sometimes don't show up. So for both methods it holds that the service time is either 0 or the original service time, while for the first method there is always an adapted expected service time.

We also ran the model for $q = 1$, so for the case that there is a no-show probability of one, in which nobody shows up at all. It is expected that both the costs and the interarrival times are equal to zero, since there are no customers that need to arrive and thus also no costs made. This was indeed the case.

Experiment 7.5. Benefits of including no-shows

This experiment is performed to look into the benefits, considering the costs of the schedule, of including no-shows, and thus whether it is important to do so or not. For this, we have first calculated the costs of the optimal schedule for $n = 5$, $\mathbb{E}[B_i]$= $\mathrm{Var}[B_i] = \mathrm{SCV}[B_i] = 1$, $\omega = 0.5$ and $q = 0.1, 0.2, ...0.9$, using the method of including no-shows that extends a exponential distribution to a hyperexponential distribution, as discussed in Subsection 7.2. Those costs are referred to as $cost_{qincluded}$. Then the optimal interarrival times in the case of no-show probability $q = 0$, that thus corresponds to the values of Figure 4, are used to calculate the costs of the schedule when there are no-shows, but those were not taken into account when making the schedule,, defined as $cost_{qnotincluded}$. Per no-show probability the ratio of those two costs, defined as $\frac{cost_{qnotincluded}}{cost_{qincluded}}$, is calculated and the results are presented in Figure 18.

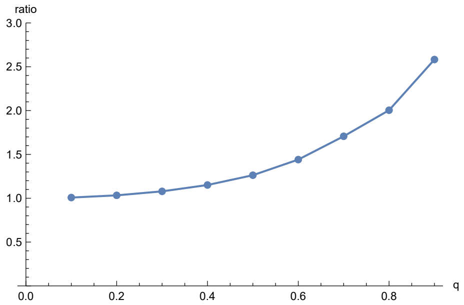


Figure 18: ratio of interarrival times for $q = 0.1, 0.2, ..., 0.9$.

From this figure it can be observed that including no-shows in your model is definitely worth it. The higher the no-show probability, the higher the gain of including them in your model.

Experiment 7.6. Dynamic
In this experiment we ran the dynamic model including no-shows, as described in Subsection 7.4, with different no-show probabilities and different values of ω. The results can be found Table 17. One can observe that also in the dynamic case the costs decreases when no-shows are included. The same pattern as for the static case in Experiment 7.2. appears, the higher the no-show probability the higher the decrease of costs. In Table 18 the costs for scheduling dynamic and static when no-shows are included with $\omega = 0.5$ are plotted. Moreover, the ratio of those costs is calculated, defined as $r(NS) = \frac{K_{dyn}(NS)}{K_{stat}(NS)}$. The gain of scheduling dynamic instead of static seems to become more substantial when the no-show probability increases.

q| ω	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
0	0.94	1.36	1.58	1.67	1.65	1.54	1.34	1.04	0.62
0.2	0.935	1.34	1.56	1.64	1.61	1.50	1.29	0.99	0.58
0.4	0.90	1.28	1.46	1.51	1.47	1.33	1.12	0.83	0.47
0.6	0.82	1.12	1.24	1.23	1.14	0.99	0.81	0.60	0.33
0.8	0.63	0.76	0.74	0.69	0.62	0.54	0.44	0.32	0.18

Table 17: dynamic with no-shows.

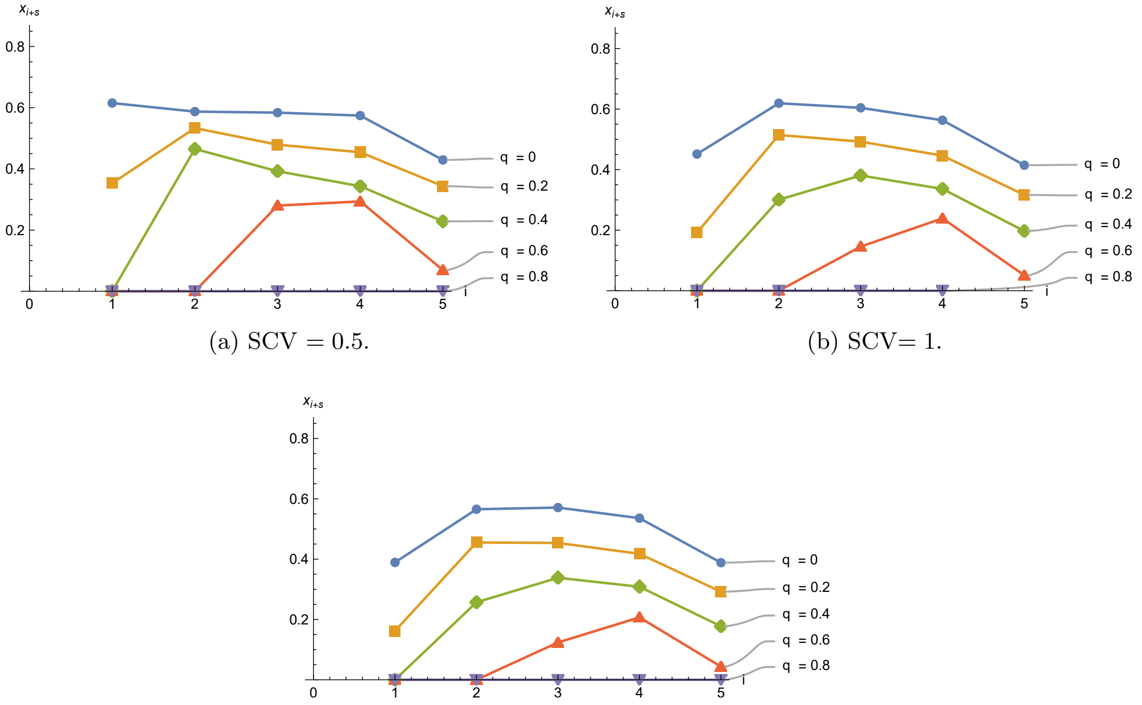
$\boldsymbol{NS}$	$\boldsymbol{K_{dyn}(NS)}$	$\boldsymbol{K_{stat}(NS)}$	$\boldsymbol{r(NS)}$
0	1.65	1.88	0.88
0.2	1.61	1.86	0.87
0.4	1.47	1.73	0.84
0.6	1.14	1.46	0.78
0.8	0.62	0.94	0.65

Table 18: Costs dynamic and static scheduling including no-shows with $\omega = 0.5$.

Experiment 7.7. Multiserver
Lastly we also implemented no-shows in the multiserver model. The function got a longer running time, due to extra matrix multiplications that have to be done, compared to the multiserver model without no-shows. Therefore we have ran the model only for the case with two servers. To generate the results both our Mathematica code and the Matlab code of Kuiper, the writer of [23], are used.

We have generated the results for $s = 2$, $n = 7$ and no-show probability $q = 0, 0.2, ..., 0.8$. The interarrival times are presented in Figure 19 and the costs in Table 19. Comparing the results from this experiment with the results from experiment 6.1 might be difficult, due to a difference in the objective function.

As expected the interarrival times and the costs decrease when the no-show probability increases, as in the single server case. The reversed bullwhip that was visible in Figure 15a disappears when there are no-shows, see Figure 19a.



(a) SCV = 0.5.

(b) SCV= 1.

(c) SCV = 1.5.

Figure 19: Interarrival times for $n = 7$ and $s = 1, 2, 3$ or 4.

q|SCV	**0.5**	**1**	**1.5**
0	1.4712	1.8968	2.0576
0.2	1.4293	1.7096	1.8095
0.4	1.2485	1.4106	1.4655
0.6	0.9371	1.0024	1.0288
0.8	0.4649	0.4966	0.5134

Table 19: Costs for $s = 2$, $q = 0, 0.2, ..., 0.8$, $n = 7$ and SCV $= 0.5, 1, 1.5$.

8 Conclusion and discussion

In this Bachelor Final Project different methods for appointment scheduling in the single server setting, static, dynamic and adaptive have been discussed. Additionally a multiserver setting has been studied and no-shows and walk-ins are included in different models. For all models some experiments are performed and their results are analysed.

For the single server setting we started with the static case. For this, we have worked out a model for the case of homogeneous exponential distributed service times and one for the case of general distributed service times. For the latter, we have used a phase-type fit, as is generally done in literature. In Subsection 5.1.3 we investigated the influence of the mean, variance and ω on the costs and interarrival times. All results that we have found are similar to results discussed in already existing literature. Moreover, we have presented a method how appointment scheduling can be used in practice. By using data from different countries about the mean and standard variation of the consultation length, we have shown how doctors per country can be provided an advised number of clients to be scheduled per day. In Subsection 5.2 we discussed the dynamic approach, following a paper by Mahes, Mandjes, Boon and Taylor [5]. We got nearly the same results as them, so we can follow their conclusion that the dynamic approach results in lower costs than the static approach. The last method we discussed for the single server setting is an adaptive one, following [5]. We found that in the case of a mixed Erlang distribution (SCV < 1) the expected sojourn time of the client in service decreases when u increases, while in the case of a hyperexponential distribution (SCV > 1) this value increases. Moreover, we managed to calculate a general expression for the expected sojourn time in both cases.

Furthermore, we have discussed the multiserver setting in Section 6, following a recent paper by Kuiper and Lee [23]. We have described their method and have provided a worked out example in Subsection 6.1.3, to make it easier to understand the inner workings of the method. Due to the long running time we have ran the multiserver model only for two, three or four servers and seven clients. This relatively low number of servers and clients makes it hard to conclude something from the results, but so far we have found results along the ones Kuiper and Lee discussed in their paper.

Finally we have discussed in Section 7 for multiple models how no-shows (and for some models also walk-ins) can be dealt with. For the exponential static case we have done this in four different ways. We have used an adjusted expected service time as in [10], we have extended the exponential distribution to a hyperexponential distribution, adapted the probabilities in the static exponential model as discussed in Subsection 5.1.1 and we have used the multiserver method as discussed in [23]. Extending the exponential distribution to a hyperexponential distribution and adapting the probabilities in the static exponential model gave the same results. However, the first method gave different results. We expected that those methods would yield the same results, as it is known that phase-type distributions approximate other distributions well. An explanation for those differing results has not been found yet. It could still be the case that this is due to an optimization or modelling error. Including no-shows in the multiserver model yielded the same results as discussed in [23]. Lastly we have also included no-shows in the exponential homogeneous dynamic model, which has not been done before. For this we have changed the definition of dynamic scheduling from "scheduling the arrival epoch of client $i+1$ at the moment that client i enters the system" to "scheduling the arrival epoch of client $i+1$ at the moment that client i *should* enter the system". It is found that also when no-shows are included, dynamic scheduling results in lower costs than static scheduling. The gain of dynamic scheduling instead of static scheduling seems to become even more substantial when the no-show probability increases.

The main part of this Bachelor Final Project that could be improved upon is the amount of clients for which the schedules are calculated. Due to long running time, especially for the multiserver case, results are only gathered for a small amount of customers. The running time could be made shorter by either using a better computer or by making the code more efficient. With a lower running time, for example, Experiment 5.5 could be performed for a day instead of an hour, such that an advice could be given for the number of clients that should be scheduled per day instead of per hour. Moreover, the multiserver model including no-shows could be ran for more than 1 server.

Further research could consider more aspects discussed in [23] for the multiserver setting, for example the benefit of pooling. Moreover, it would be a great development if one could improve the multiserver model, such that it is also applicable for heterogeneous phase-type distributed service times, since as of now, the model only works if the service times of all clients follow the same distribution. Additionally it would be interesting to look into a dynamic and adaptive approach for the multiserver model, since that would probably result in lower costs as for the single server setting. Regarding including no-shows, finding an argument for the difference between the optimal interarrival times and costs when using the methods described in Subsection 7.1 and Subsection 7.2 would be an interesting next step. A last aspect for further research is to look into whether the fourth assumption, that is about customers being punctual, is realistic. If not, relaxing upon this assumption would give possibilities to make the models better applicable in reality. To conclude, all improvements that close the gap between theory and practice are welcome.

.

References

[1] R. Mahes, M. Mandjes, M. Boon, and P. Taylor, “Dynamic Appointment Scheduling,” 2022.

[2] F. Sabria and C. F. Daganzo, “Approximate Expressions for Queueing Systems with Scheduled Arrivals and Established Service Order,” University of California, Berkely, Tech. Rep. 3, 8 1989. [Online]. Available: https://www.jstor.org/stable/25768375?seq=1&cid=pdf-

[3] E. L. Lawler, J. K. Lenstra, A. H. G. R. Kan, and D. B. Shmoys, “Deterministic machine scheduling problems,” in *Elements of Scheduling*, 2021, ch. 1. [Online]. Available: https://elementsofscheduling.nl/

[4] I. Adan, C. Comte, J. Resing, and R. Timmerman, *Queueing Systems*. Eindhoven: Department of Mathematics and Computer Science, 3 2022.

[5] R. Mahes, M. Mandjes, and M. Boon, “Adaptive Appointment Scheduling,” 2022.

[6] T. Cayirli, E. Veral, and H. Rosen, “Designing appointment scheduling systems for ambulatory care services,” *Health Care Management Science*, vol. 9, no. 1, pp. 47–58, 2 2006. [Online]. Available: https://doi.org/10.1007/s10729-006-6279-5

[7] T. Cayirli and E. A. Veral, “Outpatient scheduling in health care: A review of literature,” *Production and Operations Management*, vol. 12, pp. 519–549, 2009. [Online]. Available: https://www.semanticscholar.org/paper/OUTPATIENT-SCHEDULING-IN-HEALTH-CARE%3A-A-REVIEW-OF-Cayirli-Veral/5351df157d87f12608aeefd6e37315d8575b5f5a

[8] William E. Boyce and Richard C. DiPrima, *Elementary differential equations and boundary value problems*, 9th ed. John Wiley & Sons,Inc., 2009.

[9] R. Verbelen, “A study of theoretical concepts, calibration techniques & actuarial applications Phase-type distributions & mixtures of Erlangs,” Tech. Rep., 2012.

[10] A. Kuiper, M. Mandjes, J. de Mast, and R. Brokkelkamp, “A flexible and optimal approach for appointment scheduling in healthcare,” *Decision Sciences*, 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/deci.12517

[11] R. Bierbooms, I. J. Adan, and M. van Vuuren, “Approximate analysis of single-server tandem queues with finite buffers,” *Annals of Operations Research*, vol. 209, no. 1, pp. 67–84, 10 2013. [Online]. Available: https://link.springer.com/article/10.1007/s10479-011-1021-1

[12] B. F. Nielsen, *Lecture notes on phase-type distributions for 02407 Stochastic Processes*, 2017.

[13] A. Ahmadi-Javid, Z. Jalali, and K. J. Klassen, “Outpatient appointment systems in healthcare: A review of optimization studies,” *European Journal of Operational Research*, vol. 258, no. 1, pp. 3–34, 4 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221716305239

[14] C. Dennis Pegden and M. Rosenshine, “Scheduling arrivals to queues,” *Computers & Operations Research*, vol. 17, no. 4, pp. 343–348, 1990. [Online]. Available: https://www.sciencedirect.com/science/article/pii/030505489090012V

[15] A. Kuiper, B. Kemper, and M. Mandjes, “A Computational approach to optimized appointment scheduling,” *Queueing Systems*, vol. 79, no. 1, pp. 5–36, 1 2015. [Online]. Available: https://doi.org/10.1007/s11134-014-9398-6

[16] N. T. J. Bailey, “A Study of Queues and Appointment Systems in Hospital Out-Patient Departments, with Special Reference to Waiting-Times,” *Source: Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 14, no. 2, pp. 185–199, 1952. [Online]. Available: https://www.jstor.org/stable/2983867

[17] Soren Asmussen, *Applied Probability and queues*. Aarhus: Springer, 2003.

[18] P. P. Wang, "Optimally scheduling n customer arrival times for a single-server system," *Computers & Operations Research*, vol. 24, no. 8, pp. 703–716, 1997. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305054896000937

[19] C.-J. Ho and H.-S. Lau, "Minimizing total cost in scheduling outpatient appointments," *Management Science*, vol. 38, pp. 1750–1764, 12 1992.

[20] W. Vink, A. Kuiper, B. Kemper, and S. Bhulai, "Optimal appointment scheduling in continuous time: The lag order approximation method," *European Journal of Operational Research*, vol. 240, no. 1, pp. 213–219, 1 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S037722171400527X

[21] H. Y. Mak, Y. Rong, and J. Zhang, "Appointment scheduling with limited distributional information," *Management Science*, vol. 61, no. 2, pp. 316–334, 2 2015. [Online]. Available: https://doi.org/10.1287/mnsc.2013.1881

[22] A. Kuiper, "Appointment scheduling in healthcare," Ph.D. dissertation, Amsterdam Business School Research Institute, Amsterdam, 6 2016. [Online]. Available: https://dare.uva.nl/search?identifier=c929e434-7756-470f-b581-d17d5fae1707

[23] A. Kuiper and R. H. Lee, "Appointment Scheduling for Multiple Servers," *Articles in Advance*, 2 2022. [Online]. Available: http://pubsonline.informs.org/doi/10.1287/mnsc.2021.4221

[24] R. Hassin and S. Mendel, "Scheduling arrivals to queues: A single-server model with no-shows," *Management Science*, vol. 54, no. 3, pp. 565–572, 3 2008. [Online]. Available: https://doi.org/10.1287/mnsc.1070.0802

[25] M. Deveugele, A. Derese, A. Van Den Brink-Muinen, J. Bensing, and J. de Maeseneer, "Consultation length in general practice: Cross sectional study in six European countries," *BMJ: British Medical Journal*, vol. 325, no. 7362, pp. 472–474, 2002. [Online]. Available: http://www.jstor.org/stable/25452211

[26] G. Irving, A. L. Neves, H. Dambha-Miller, A. Oishi, H. Tagashira, A. Verho, and J. Holden, "International variations in primary care physician consultation time: A systematic review of 67 countries," *British Medical Journal Publishing Group*, vol. 7, no. 10, 10 2017. [Online]. Available: https://bmjopen.bmj.com/content/7/10/e017902

A Parameters phase-type fit

In this appendix we will derive the expressions for the parameters μ, K and p for the phase-type fit as described in Subsection 2.3.

Case SCV $\leqslant$ 1 As written in Subsection 2.3, in this case we approximate a non-negative random variable B by a mixture of Erlang distributions. Let us denote by $E(K,\mu)$ an Erlang distributed random variable with shape parameter K and scale parameter μ, and U an independent uniform random variable on $[0,1]$. Then for some $K \in \mathbb{N}$, $\mu > 0$ and $p \in [0,1]$,

$$B \sim E(K,\mu)1_{\{U<p\}} + E(K+1,\mu)1_{\{U>p\}}.$$

Then we have

$$\mathbb{E}[B] = p \cdot \frac{K}{\mu} + (1-p) \cdot \frac{K+1}{\mu}$$

and

$$\mathbb{E}[B^2] = p \cdot \frac{K(K+1)}{\mu^2} + (1-p) \cdot \frac{(K+1)(K+2)}{\mu^2}.$$

From this first equality we get

$$\begin{aligned}
\mathbb{E}[B] &= p \cdot \frac{K}{\mu} + (1-p) \cdot \frac{K+1}{\mu} \\
\mathbb{E}[B] &= \frac{pK}{\mu} + \frac{(1-p)(K+1)}{\mu} \\
\mathbb{E}[B] &= \frac{pK + K + 1 - pK - p}{\mu} \\
\mathbb{E}[B] &= \frac{K+1-p}{\mu} \\
\mu &= \frac{K+1-p}{\mathbb{E}[B]}.
\end{aligned}$$

So $\frac{\mathbb{E}[B^2]}{\mathbb{E}[B]^2}$ becomes

$$\begin{aligned}
\frac{\mathbb{E}[B^2]}{\mathbb{E}[B]^2} &= \frac{\frac{(K+1)(pK+(1-p)(K-2))}{\mu^2}}{\left(\frac{(K+1-p)^2}{\mu^2}\right)} \\
&= \frac{(K+1)(pK+(1-p)(K-2))}{\mu^2} \cdot \frac{\mu^2}{(K+1-p)^2} \\
&= \frac{(K+1)(pK+(1-p)(K-2))}{(K+1-p)^2} \\
&= \frac{(K+1)(K+2(1-p))}{(K+1-p)^2}.
\end{aligned}$$

Resulting in

$$\begin{aligned}
\mathrm{SCV} &= \frac{\mathbb{E}[B^2]}{\mathbb{E}[B]^2} - 1 \\
&= \frac{(K+1)(K+2(1-p))}{(K+1-p)^2} - 1 \\
&= \frac{(K+1)(K+2(1-p))}{(K+1-p)^2} - \frac{(K+1-p)^2}{(K+1-p)^2} \\
&= \frac{(K+1)(K+2(1-p)) - (K+1-p)^2}{(K+1-p)^2} \\
&= \frac{K+1-p^2}{(K+1-p)^2}.
\end{aligned}$$

Define the function $f(\cdot)$ and its antiderivative through $f(p) = \frac{K+1-p^2}{(K+1-p)^2}$ and $f(p) = \frac{2(K+1)(1-p)}{(K+1-p)^3} > 0$ with $p \in [0,1]$. Then $f(0) = \frac{1}{K+1}$ and $f(1) = \frac{K}{K^2} = \frac{1}{K}$. So the SCV lies in between those two values. If $\mathrm{SCV} \leqslant 1$, then K is the floor of $\frac{1}{\mathrm{SCV}}$. Lastly we need to find p. To find this value, we solve the following equations from above

$$\begin{aligned}
\mathrm{SCV} &= \frac{K+1-p^2}{(K+1-p)^2} \\
K+1-p^2 &= \mathrm{SCV}(K+1-p)^2 \\
K+1-p^2 &= \mathrm{SCV}(K+1)^2 - 2\mathrm{SCV}(K+1)p + p^2\mathrm{SCV} \\
0 &= (\mathrm{SCV}+1)p^2 - 2\mathrm{SCV}(K+1)p + (K+1)^2\mathrm{SCV} - (K+1).
\end{aligned}$$

So by solving the latter quadratic equation we get

$$p = \frac{(K+1)\mathrm{SCV} \pm \sqrt{(K+1)(1-K\cdot\mathrm{SCV})}}{\mathrm{SCV}+1}.$$

We choose the one with the minus sign, since this one lies in $[0,1]$, what is needed since p is a probability.

Case SCV > 1 If the SCV is larger than 1, we approximate the non-negative random variable B by a hyperexponential distribution. For some $\mu_1, \mu_2 > 0$ and $p \in [0,1]$,

$$B \sim \exp(\mu_1)\mathbb{1}_{\{U<p\}} + \exp(\mu_2)\mathbb{1}_{\{U>p\}}.$$

We then get $\mathbb{E}[B] = p \cdot \frac{1}{\mu_1} + (1-p)\cdot\frac{1}{\mu_2}$ and $\mathbb{E}[B^2] = p\cdot\frac{2}{\mu_1^2} + (1-p)\cdot\frac{2}{\mu_2^2}$. Using the principle of balanced means, so choose $\mu_1 = 2p\mu$ and $\mu_2 = 2(1-p)\mu$, we get

$$\begin{aligned}
\mathbb{E}[B] &= \frac{p}{\mu_1} + \frac{1-p}{\mu_2} \\
&= \frac{p}{2p\mu} + \frac{1-p}{2(1-p)\mu} \\
&= \frac{1}{2\mu} + \frac{1}{2\mu} = \frac{1}{\mu},
\end{aligned}$$

thus $\mu = \frac{1}{\mathbb{E}[B]}$. To get the SCV, we first determine $\frac{\mathbb{E}[B^2]}{\mathbb{E}[B]^2}$.

$$\begin{aligned}
\frac{\mathbb{E}[B^2]}{\mathbb{E}[B]^2} &= \mu^2 \left(p \cdot \frac{2}{(2p\mu)^2} + (1-p) \cdot \frac{2}{4(1-p)^2\mu^2} \right) \\
&= \frac{1}{2p} + \frac{1}{2(1-p)} \\
&= \frac{1}{2p(1-p)}.
\end{aligned}$$

The SCV then becomes

$$\begin{aligned}
\text{SCV} &= \frac{\mathbb{E}[B^2]}{\mathbb{E}[B]^2} - 1 \\
&= \frac{1}{2p(1-p)} - 1 \\
&= \frac{1}{2p(1-p)} - \frac{2p-2p^2}{2p-2p^2} \\
&= \frac{1-2p+2p^2}{2p-2p^2}.
\end{aligned}$$

After rewriting this formula, we can get the value for p.

$$\begin{aligned}
\text{SCV} &= \frac{1-2p+2p^2}{2p-2p^2} \\
(2p-2p^2)\text{SCV} &= 1-2p+2p^2 \\
-2p^2(\text{SCV}+1) + 2p(\text{SCV}+1) - 1 &= 0 \\
p^2 - p + \frac{1}{2(\text{SCV}+1)} &= 0.
\end{aligned}$$

When solving the latter quadratic equation we get

$$p = 2\left(1 \pm \sqrt{\frac{\text{SCV}-1}{\text{SCV}+1}}\right).$$

We always take $p = 2\left(1 - \sqrt{\frac{\text{SCV}-1}{\text{SCV}+1}}\right)$, since we assume $\mu_1 > \mu_2$.

B Objective function

As written in the model description we use a linear objective function. However, there are also other options for objective functions. A brief description of possible cost functions is provided in this appendix.

Most of the papers in literature regarding appointment scheduling include the idle time and waiting time in the objective function. There are mainly two variants of doing so. Firstly there is the linear objective function, the one that we use, which is also used in [5, 1, 22]. It is a weighted sum of the expected idle times of the server and the expected waiting times of the client. Such a function looks as follows:

$$C[x_1, x_2, ..., x_i] = \omega \sum_{i=1}^{n} \mathbb{E}[I_i] + (1-\omega) \sum_{i=1}^{n} \mathbb{E}[W_i], \tag{B.1}$$

with I_i the idle time of client i, W_i the waiting time associated with client i and ω a variable factor to define which of the latter two quantities is the most important in the weighted sum.

Secondly there is the quadratic objective function. Which is a weighted sum of the squared expected idle times of the server and the squared expected waiting times of the client. Such a function looks as follows:

$$C[x_1, x_2, ..., x_i] = \omega \sum_{i=1}^{n} \mathbb{E}[I_i^2] + (1-\omega) \sum_{i=1}^{n} \mathbb{E}[W_i^2]. \tag{B.2}$$

When using such a quadratic objective function, you are punished harder for higher values than when using a linear objective function. This variant is studied in a.o. [22, 15].

Furthermore there are two other quantities that are relatively often included in the objective function. Firstly there is the *overtime*. The overtime is the difference between the real and the expected makespan, the time that the server should be available. Mathematically, when T_{real} denotes the real makespan and T_{exp} the expected makespan, the overtime is defined as $T_{real} - T_{exp}$. Overtime is clearly something that is bad for the server. Overtime results in the server having to work longer than expected. To deal with this extra term, $\bar{\omega}\mathbb{E}[O]$, can be included in the objective function, such that the objective function becomes

$$C[x_1, x_2, ..., x_i] = \omega \sum_{i=1}^{n} \mathbb{E}[I_i] + (1-\omega) \sum_{i=1}^{n} \mathbb{E}[W_i] + \bar{\omega}\mathbb{E}[O]. \tag{B.3}$$

Here O denotes the overtime and $\bar{\omega}$ is a factor that defines the weight of the overtime. When overtime is included into the objective function it has, in all papers that we studied, always been added to an objective function that already contained expected idle and waiting times, resulting in an expression as Equation (B.3), instead of that it replaced one of those two quantities. Including overtime in the objective function is studied in a.o. [22].

Secondly there is the availability of a server. When this quantity is used it mostly replaces the server's idle time, see for example [14] or [24]. When doing so the objective function becomes:

$$C[x_1, x_2, ..., x_i] = \omega_w \sum_{i=1}^{n} w_i + \omega_s \left(t_1 + \sum_{i=1}^{n-1} x_i + w_n + \mathbb{E}[B_n] \right), \tag{B.4}$$

where ω_w and ω_s denote the weight of customers waiting time and the servers availability. In [14] those quantities are defined as the customer waiting cost per unit time and the server availability cost per unit time.

C Proof convexity single server

Theorem: C$[x_2, x_3, ..., x_n]$ is convex in $\boldsymbol{x} = (x_2, x_3, ..., x_n)$, and therefore, there is a unique minimum on $\mathbb{R}_+^{n-1}$. Here $x_n = t_i - t_{i-1}$ and x_1 is fixed at 0.

Proof. C$[x_2, x_3, ..., x_n]$ is defined as

$$C[x_2, x_3, ..., x_n] = \omega \sum_{i=1}^{n} \mathbb{E}[I_i] + (1-\omega) \sum_{i=1}^{n} \mathbb{E}[W_i]. \tag{C.1}$$

By using

$$\begin{aligned} B_i + I_i &= x_i + S_i - S_{i-1} \\ I_i &= x_i + S_i - B_i - S_{i-1} \\ I_i &= x_i + W_i - W_{i-1} - B_{i-1} \end{aligned} \tag{C.2}$$

we can rewrite the objective function to

$$\begin{aligned} C[x_2, x_3, ..., x_n] &= \underset{x_2,...,x_n}{\arg\min} \left(\omega \sum_{i=1}^{n} \mathbb{E}[I_i] + (1-\omega) \sum_{i=1}^{n} \mathbb{E}[W_i] \right) \\ &= \underset{x_2,...,x_n}{\arg\min} \left(\omega \sum_{i=1}^{n} \mathbb{E}[x_i + W_i - W_{i-1} - B_{i-1}] + (1-\omega) \sum_{i=1}^{n} \mathbb{E}[W_i] \right) \\ &= \underset{x_2,...,x_n}{\arg\min} \left(\omega \sum_{i=1}^{n} \mathbb{E}[x_i - B_{i-1}] + \mathbb{E}[W_n] + (1-\omega) \sum_{i=1}^{n-1} \mathbb{E}[W_i] \right) \end{aligned} \tag{C.3}$$

where in the last line the $\mathbb{E}[B_{i-1}]$ terms can be dropped for the optimization, as these are constants and will thus not affect the optimal interarrival times. This then results in the following equation

$$= \omega \sum_{i=1}^{n} \mathbb{E}[x_i] + \mathbb{E}[W_n] + (1-\omega) \sum_{i=1}^{n-1} \mathbb{E}[W_i]. \tag{C.4}$$

From the equality in Equation C.3 it follows that the objective function is a linear combination of expected waiting times minus a linear combination of $\boldsymbol{x}$, which is convex. Therefore the only thing left to prove is that expected waiting times are convex. Define by $W_i(x)$ the waiting time of the i-th patient if the vector of interarrival times is given by $\boldsymbol{x}$. Then we have to prove

$$\mathbb{E}[W_i(\nu \boldsymbol{x_1} + (1-\nu)\boldsymbol{x_2})] \leqslant \nu[W_i(\boldsymbol{x_1})] + (1-\nu)\mathbb{E}[W_i(\boldsymbol{x_2})], \tag{C.5}$$

for $\boldsymbol{x_1}, \boldsymbol{x_2} \in \mathbb{R}_+^{n-1}$ and $\nu \in [0,1]$.

For a given i define

$$Z_j = \sum_{k=i-j+1}^{i} B_k$$

and

$$y_j = \sum_{k=i-j+1}^{i} x_{k+1},$$

where B_k denotes the service time of client k, Z_j the total service time of client i and the $j-1$ clients before them and y_j the sum of the interarrival times of client i and the $j-1$ before them. By repeatedly applying $W_i = \max\{W_{i-1} + B_{i-1} - x_i, 0\}$, that followed from the *Lindley Recursion*, the following distributional equality is obtained:

$$W_i(\boldsymbol{x}) \overset{\mathrm{d}}{=} \max_{j \in \{0,1,...,i-1\}} \{Z_j - y_j(\boldsymbol{x})\}.$$

Then

$$\begin{aligned}
\mathbb{E}[W_i\left(\nu \boldsymbol{x_1}+(1-\nu)\boldsymbol{x_2}\right)] &= \mathbb{E}\left[\max_{j\in\{0,\ldots,i-1\}}\left\{Z_j - y_j\left(\nu\boldsymbol{x_1}+(1-\nu)\boldsymbol{x_2}\right)\right\}\right] \\
&= \mathbb{E}\left[\max_{j\in\{0,\ldots,i-1\}}\left\{Z_j - \nu y_j\left(\boldsymbol{x_1}\right)-(1-\nu)y_j\left(\boldsymbol{x_2}\right)\right\}\right] \\
&= \mathbb{E}\left[\max_{j\in\{0,\ldots,i-1\}}\left\{\nu\left(Z_j - y_j\left(\boldsymbol{x_1}\right)\right)+(1-\nu)\left(Z_j - y_j\left(\boldsymbol{x_2}\right)\right)\right\}\right] \\
&\leqslant \mathbb{E}\left[\max_{j\in\{0,\ldots,i-1\}}\left\{\nu\left(Z_j - y_j\left(\boldsymbol{x_1}\right)\right)\right\}\right] \\
&\quad + \mathbb{E}\left[\max_{j\in\{0,\ldots,i-1\}}\left\{(1-\nu)\left(Z_j - y_j\left(\boldsymbol{x_2}\right)\right)\right\}\right]
\end{aligned}$$

where the last inequality follows from the triangle inequality. So we have proven Equation (C.5) and thus as a consequence, it can be concluded that there is a unique minimum on $\mathbb{R}_+^{n-1}$ [10].