BACHELOR

Waiting time approximations for systems with multi-type jobs and multi-type servers

Boeve, S.

Award date:
2017

Technische Universiteit
Eindhoven
University of Technology

Eindhoven University of Technology
PO Box 513, 5600 MB Eindhoven
www.tue.nl/en

Author
Serge Boeve
Identity number: 0830773
Department: M&CS
Course code: 2WH40/50

Supervisors
dr.ir. M.A.A. Boon
prof.dr.ir. I.J.B.F. Adan

Date
Feb. 2016 – Feb. 2017

Waiting time approximations for systems with multi-type jobs and multi-type servers

Bachelor Final Project

Technische Universiteit
Eindhoven
University of Technology

Table of contents

Abstract

Consider a parallel service system with Poisson input with servers $\mathcal{M} = \{\boldsymbol{m}_1, \ldots, \boldsymbol{m}_K\}$, and with job types $\mathcal{C} = \{a, b, c, \ldots\}$. Service is skill-based, so that server $\boldsymbol{m}_i$ can serve a subset of job types $\mathcal{C}(\boldsymbol{m}_i)$. Waiting jobs are served on a first-come-first-served basis, while arriving jobs that find multiple idle servers are assigned to a feasible server randomly. Previous studies of memoryless service systems showed the existence of specific assignment probabilities under which the system has a product-form stationary distribution, and provide explicit expressions for it. Systems under the combined first-come-first-served–assign-longest-idle-server policy surprisingly possess the same stationary distribution. Furthermore, experimental results suggest that the waiting probabilities are insensitive to processing time distributions, allowing approximate results for general service requirements as well.

In this report we derive two explicit approximation formulas for the mean waiting time; one based on weighted average residual service and one based on second order light traffic-heavy traffic interpolation. Both approximations perform quite well, with the one based on weighted average residual service yielding the best results.

1 Introduction

Consider a service system with some servers (machines), which serves several types of jobs. Jobs arrive at the system in independent Poisson streams and have independent service requirements. Each machine is capable of handling a specific subset of job types.

These systems are commonly referred to as *skill-based parallel service systems*. In recent years they have been analysed quite extensively because of their many applications. They can be used to describe, for example, manufacturing processes, skill-based routing of calls to operators in call centres, routing of wireless messages to ad hoc nodes, processing of chips, multiprocessor scheduling and mounting of printed circuit boards; see in particular [1-4].

The behaviour of such systems is highly dependent on the type of policy which is used to assign servers to jobs. Many such policies have been researched, such as shortest queue arrival [5], fastest servers first [6], queue-ratio routing [7] and assign to longest idle server [8]. After Jackson's discovery that Jackson networks have a product-form stationary distribution, many have searched for product-form results for this kind of systems, as they are believed to be the best way to get explicit results for more general models.

Visschers, Adan and Weiss were able to first derive a product-form solution for systems in which service is First-Come-First-Served (FCFS), and arriving jobs that find several idle servers are assigned to a compatible server randomly [9]. However, the result only held for a singular choice of the assignment distribution.

Adan and Weiss then derived another product-form solution for an important class of skill-based parallel service systems: those under the combined First-Come-First-Served–Assign-Longest-Idle-Server (FCFS-ALIS) policy [8]. This means that a machine on completion of service takes the first job in the queue that it can process and that a job which on arrival finds multiple feasible available machines will be assigned to the longest idle one. Surprisingly it turned out that both systems possess exactly the same stationary distribution. As a consequence, both systems also have the same waiting time distributions, which Visschers, Adan and Weiss derived for Poisson arrivals and exponential service requirements [9].

Currently there are no results for general service requirements. However, since the waiting probabilities are believed to be almost insensitive to processing time distributions (see also [10]), we can approximate the waiting time for each job type in case of general service requirements using these results.

In this report we present two ways of approximating the mean waiting time for skill-based parallel service systems. The first, which will be referred to as approximation 1 or the first approximation and is derived in Chapter 3, is based on weighted average residual service times and is inspired by the results obtained in [9]. The second, derived in Chapter 4 and inspired by the results obtained by Reiman and Simon in [12], is based on second order interpolation between light-traffic and heavy-traffic behaviour, and is referred to as approximation 2 or the second approximation. For ease of use of earlier results obtained for exponential service requirements we assume Poisson arrivals and a combination of FCFS and random assignment, as described in [9], however it should be noted that the combined FCFS-ALIS policy should yield similar results.

Both approximations will be tested using simulation results (Chapter 5). It turns out that they perform quite well for some common service time distributions and particular choices of the assignment distribution. The first approximation performs slightly better than the second one. Higher order polynomials might yield better results for the second approximation, but this will not be tested here.

2 Model description, theory and examples

Consider a service system with K servers (machines), labelled $\boldsymbol{m}_1, \ldots, \boldsymbol{m}_K$, and several types of jobs, labelled $a, b, c, \ldots$. Let $\mathcal{C}$ denote the set of job types, and $\mathcal{M}$ the set of machines. Arrivals are independent and Poisson distributed with rates $\lambda_i,\ i \in \mathcal{C}$. Arriving jobs require independent, generally distributed service times with rate $\mu_{\boldsymbol{m}_i}$ for machine $\boldsymbol{m}_i$. Each machine can handle a specific subset of job types, denoted by $\mathcal{C}(\boldsymbol{m}_i) \subset \mathcal{C}$ for machine $\boldsymbol{m}_i$.

The service discipline is a combination of First-Come-First-Served (FCFS) and random assignment. Jobs which upon arrival find no feasible available machine wait in a single queue, and are processed in an FCFS order as long as it is possible. This means that on completion of service the machine takes the first job in the queue that it can handle, possibly skipping several jobs that it cannot handle. Arriving jobs which find multiple feasible available machines are assigned to one of them randomly and go into service immediately.

We assume that an arriving job will choose a feasible machine from the set of idle machines according to a specified *assignment probability distribution* which depends on the job type and on the set of idle machines. There is one assignment probability distribution for each type of job and for each subset of idle machines which contains at least one feasible machine for that type of job. These assignment probability distributions are treated as control parameters for the system. Figure 1 shows two examples of such systems, which we will refer to as System A and System B.

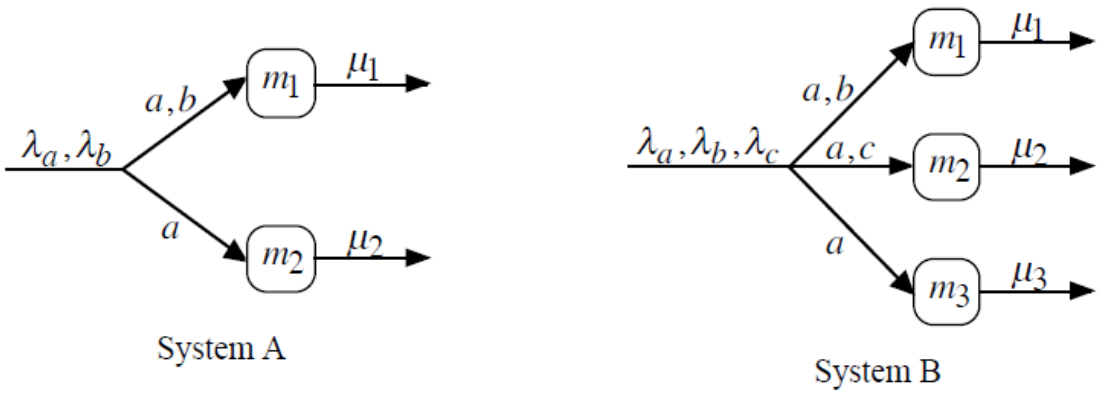

Figure 1 *Some service systems with multi-type jobs and multi-type servers.*

System A has two machines, $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$, and two job types, type a and type b, hence $\mathcal{M} = \{\boldsymbol{m}_1, \boldsymbol{m}_2\}$ and $\mathcal{C} = \{a, b\}$. Jobs of both types arrive in independent Poisson streams with rates λ_a and λ_b for job types a and b respectively. Machine $\boldsymbol{m}_1$ can serve both job types, while machine $\boldsymbol{m}_2$ can serve only type-a jobs, hence $\mathcal{C}(\boldsymbol{m}_1) = \{a, b\}$ and $\mathcal{C}(\boldsymbol{m}_2) = \{a\}$.

System B has three machines, $\boldsymbol{m}_1$, $\boldsymbol{m}_2$ and $\boldsymbol{m}_3$, and three job types, a, b and c. Arrivals are Poisson with rates λ_a, λ_b and λ_c for job types, a, b and c respectively. Machine $\boldsymbol{m}_1$ can serve both job types a and b, machine $\boldsymbol{m}_2$ can serve both job types a and c and machine $\boldsymbol{m}_3$ can serve only type-a jobs, hence $\mathcal{C}(\boldsymbol{m}_1) = \{a, b\}$, $\mathcal{C}(\boldsymbol{m}_2) = \{a, c\}$ and $\mathcal{C}(\boldsymbol{m}_3) = \{a\}$.

To give a state description for the system, all the jobs in the system are listed in their order of arrival, including jobs which are being processed, and the machines are imagined to be situated in the queue on the position of the job that they are processing. As an example, consider System B in Figure 1, in which there are three job types, a, b and c, and three machines, $\boldsymbol{m}_1$, $\boldsymbol{m}_2$ and $\boldsymbol{m}_3$, with $\mathcal{C}(\boldsymbol{m}_1) = \{a, b\}$, $\mathcal{C}(\boldsymbol{m}_2) = \{a, c\}$ and $\mathcal{C}(\boldsymbol{m}_3) = \{a\}$.

Figure 2 depicts a possible state for System B. The jobs are denoted by circles and the machines by rectangles. Jobs in service have a rectangle drawn around them with the identity of the machine inside. There are 12 jobs in the system and all three machines are busy. Machine $\boldsymbol{m}_1$ is processing the first job in line, which must therefore be either of types a or b. Following the line, machine $\boldsymbol{m}_2$ is processing the first job in the line which it can process, which is job 5 in the line, and must therefore be either of types a or c. Machine $\boldsymbol{m}_3$ is processing the first job in the line (apart from jobs 1 and 5) which it can process, which must be of type a. There are three jobs waiting between machines $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$. These cannot be processed by either machine $\boldsymbol{m}_2$ or by machine $\boldsymbol{m}_3$, so they must be of type b. There are four jobs waiting between machines $\boldsymbol{m}_2$ and $\boldsymbol{m}_3$. Those cannot be processed by machine $\boldsymbol{m}_3$, so they must be either of types b or c. Finally, there are two jobs at the back of the queue, behind machine $\boldsymbol{m}_3$, which may be either of types a, b, or c.

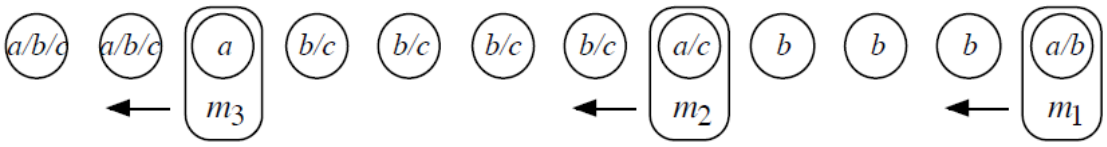

Figure 2 *A possible state for System B.*

Some of the states in this description can be aggregated to simplify the model. We will retain the identity and location of the busy machines, but we will not specify the type of job which they are working on, and we will only record the number of jobs between the busy machines, and not specify the string of job types. Thus the state of Figure 2 will be denoted as $(2, \boldsymbol{m}_3, 4, \boldsymbol{m}_2, 3, \boldsymbol{m}_1)$. Note that with this reduced description the system is still Markovian [9].

2.1 Product-form stationary distribution

We will now describe the general K machine system and give its product-form stationary distribution, derived by Visschers, Adan and Weiss. Note that these results hold only for exponential service requirements. Therefore, we assume in this section that service times are exponential. In Chapter 5 it is shown that the waiting probabilities are almost insensitive to processing time distributions, and hence can be used for those systems as well.

For the general K machine system we introduce the following notation:

- M := an arbitrary machine M from the set of machines $\mathcal{M} = \{\boldsymbol{m}_1, \dots, \boldsymbol{m}_K\}$. M points to one of the machines $\boldsymbol{m}_i$. M_i points to the machine at the i^{th} position, not necessarily to machine $\boldsymbol{m}_i$.
- $\lambda_{\mathcal{X}}$:= $\sum_{c \in \mathcal{X}} \lambda_c$, where $\mathcal{X} \subset \mathcal{C}$.
- $\mu_{\mathcal{Y}}$:= $\sum_{M \in \mathcal{Y}} \mu_M$, where $\mathcal{Y} \subset \mathcal{M}$.
- $\mathcal{C}(\mathcal{Y})$:= total set of job types that can be handled by the machines in $\mathcal{Y} \subset \mathcal{M}$, which is equal to $\bigcup_{M \in \mathcal{Y}} \mathcal{C}(M)$.
- $\mathcal{U}(\mathcal{Y})$:= set of job types unique to the machines in $\mathcal{Y} \subset \mathcal{M}$, thus the set of job types that cannot be handled by machines outside $\mathcal{Y}$.

For models with K machines, we use the following aggregated state description:

$(n_i, M_i, n_{i-1}, M_{i-1}, \dots, n_1, M_1)$: States in which there are i machines busy. These machines are denoted by $M_1, \dots, M_i$, where $\{M_1, \dots, M_i\} \subset \mathcal{M}$. The number of jobs between machines M_j and M_{j+1} is denoted by n_j (≥ 0), with $j = 1, \dots, i-1$. The number of waiting jobs at the end of the queue, behind machine M_i, is denoted by n_i. Note that $M_1, \dots, M_i$ point to the location of the machine (M_1 is working on the first job, M_2 on the $n_1 + 2^{\text{nd}}$ etc.) and not to the identity.

The state space is denoted by $\mathcal{S}$ and to simplify the notation we use $\mathfrak{s}$ to denote an arbitrary state $(n_i, M_i, n_{i-1}, M_{i-1}, \dots, n_1, M_1) \in \mathcal{S}$. Figure 3 shows a system in state $\mathfrak{s}$.

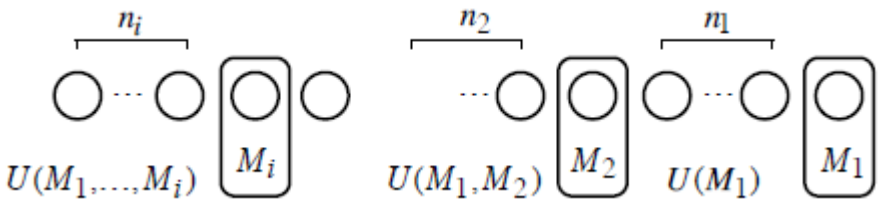

Figure 3 *General system in state* $\mathfrak{s} = (n_i, M_i, n_{i-1}, M_{i-1}, \dots, n_1, M_1)$.

Note that the n_j jobs waiting between machines M_j and M_{j+1} can only be handled by machines $M_1, \dots, M_j$, and not by any of the machines $M_{j+1}, \dots, M_i$ or any of the idle machines, due to the First-Come-First-Served processing order. Thus waiting jobs between machines M_j and M_{j+1} can only be of type $c \in \mathcal{U}(\{M_1, \dots, M_j\})$. The n_i waiting jobs in the back of the queue cannot be handled by any of the idle machines and thus have to be of type $c \in \mathcal{U}(\{M_1, \dots, M_i\})$.

For the system to be ergodic, it is necessary that both

$$\frac{\lambda_{\mathcal{U}(\boldsymbol{m}_i)}}{\mu_{\boldsymbol{m}_i}} < 1 \text{ for all } \boldsymbol{m}_i \in \mathcal{M}, \text{ and } \frac{\lambda_{\mathcal{C}}}{\mu_{\mathcal{M}}} < 1 \tag{1}$$

should hold. These conditions are also sufficient [9].

Let S denote the set of idle servers at a given moment. Note that S contains exactly those servers $M_{i+1}, \dots, M_K$ that are not busy when the system is in state $(n_i, M_i, n_{i-1}, M_{i-1}, \dots, n_1, M_1)$. There are several control parameters for the system:

$\eta_{M_j}(S)$ is the rate at which server $M_j \in S$ is activated.

This rate is also referred to as the *activation rate* and can be calculated recursively from the formulas derived by Adan, Hurkens and Weiss in [10]. They considered a closely related system: the reversible Erlang loss system. However, it was shown that the assignment probabilities for this system are exactly those derived in [10]. Hence, the activation rates are given by

$$\eta_{M_j}(S) = \eta(S) \cdot \left(1 + \sum_{M_k \in S \setminus \{M_j\}} \frac{\eta_{M_k}(S \setminus \{M_j\})}{\eta_{M_j}(S \setminus \{M_k\})} \right)^{-1}, \tag{2}$$

where

$$\eta(S) = \sum_{M_j \in S} \eta_{M_j}(S) = \sum_{c \in \mathcal{C}(S)} \lambda_c \,. \tag{3}$$

Let $P(c, M_j|S)$ denote the probability that an arriving job of arbitrary type $c \in \mathcal{C}(S)$ will choose server $M_j \in S$ when the system is in state S. Note that if S holds only one server or if $c \in \mathcal{U}(M_j)$ this probability equals one. In all other cases these probabilities can be obtained by solving the following set of equations

$$\eta_{M_j}(S) = \sum_{c \in \mathcal{C}(M_j)} \lambda_c \, P(c, M_j|S). \tag{4}$$

Now that the transition behaviour of the system is known, one could formulate the set of equilibrium equations and decompose the equations into partial balance equations. Visschers, Adan and Weiss showed that, if partial balance is satisfied, the model has a unique product-form solution [9]. This product-form solution, however, does not exist if one expands the state space to include the identity of the job in service. Hence, for the product-form solution to exist we need to have the aggregated state description. Also, partial balance will not be satisfied for every value of the control parameters. With the use of partial balance, Visschers, Adan and Weiss derived a

candidate product-form solution which, if the above activation rates are chosen as control parameters for the system, is the solution to the global balance equations.

Let $\pi(\mathfrak{s})$ denote the stationary probability of state $\mathfrak{s}$. After normalization the stationary probability $\pi(\mathfrak{s})$ of state $\mathfrak{s} = (n_i, M_i, n_{i-1}, M_{i-1}, \dots, n_1, M_1) \in \mathcal{S}$ can be obtained from

$$\pi(\mathfrak{s}) = \alpha_1^{n_1} \dots \alpha_i^{n_i} \frac{\eta_{M_1}(\mathcal{M}) \dots \eta_{M_i}(\mathcal{M} \backslash \{M_1, \dots, M_{i-1}\})}{\mu_{M_1} \cdots \mu_{\{M_1,\dots,M_i\}}} \pi(0)\,, \tag{5}$$

where

$$\alpha_j = \frac{\lambda_{\mathcal{U}(\{M_1,\dots,M_j\})}}{\mu_{\{M_1,\dots,M_j\}}}, \qquad j = 1,2,\dots,i\,. \tag{6}$$

As an example, consider System A of Figure 1, in which there are two job types, a and b, and two machines, $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$, with $\mathcal{C}(\boldsymbol{m}_1) = \{a, b\}$ and $\mathcal{C}(\boldsymbol{m}_2) = \{a\}$. For ease of presentation, let the service rates be denoted by μ_1 and μ_2 for machines $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ respectively. The system is ergodic if

$$\frac{\lambda_b}{\mu_1} < 1, \text{ and } \frac{\lambda_a + \lambda_b}{\mu_1 + \mu_2} < 1\,. \tag{7}$$

In the aggregated description the system can be in any of the following states:

- $(n, \boldsymbol{m}_2, m, \boldsymbol{m}_1)$, where $m, n \geq 0$, with $m + n + 2$ jobs, machine $\boldsymbol{m}_1$ is working on the first job, followed by m jobs waiting between machines $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$, all of which must be of type b, followed by machine $\boldsymbol{m}_2$ working on the $m + 2^{\text{nd}}$ job in line, followed by n jobs of unidentified type waiting behind machine $\boldsymbol{m}_2$.
- $(n, \boldsymbol{m}_1, 0, \boldsymbol{m}_2)$, where $n \geq 0$, with $n + 2$ jobs, machine $\boldsymbol{m}_2$ is working on the first job, machine $\boldsymbol{m}_1$ is working on the second job, and there are n jobs of unidentified type waiting behind the two machines.
- $(m, \boldsymbol{m}_1)$, where $m \geq 0$, with $m + 1$ jobs, machine $\boldsymbol{m}_1$ is working on the first job, followed by m jobs which must all be of type b, and machine $\boldsymbol{m}_2$ is idle.
- $(0, \boldsymbol{m}_2)$, machine $\boldsymbol{m}_2$ is working on a single job, and machine $\boldsymbol{m}_1$ is idle.
- (0), the empty system state.

Note that here only jobs of type a have a choice of machines, and that happens only when they arrive to find both machines idle; in other words, when the system is empty. Hence, the assignment probability distributions reduce simply to the probability η of assigning an arriving job of type a to machine $\boldsymbol{m}_1$ when the system is empty. The choice of η is unique for this system and can be calculated from (2), (3) and (4) (see also [10]). First, we calculate the activation rates:

$$\begin{aligned}
\eta_{\boldsymbol{m}_1}(\{\boldsymbol{m}_1,\boldsymbol{m}_2\}) &= \eta(\{\boldsymbol{m}_1,\boldsymbol{m}_2\})\left(1+\frac{\eta_{\boldsymbol{m}_2}(\{\boldsymbol{m}_2\})}{\eta_{\boldsymbol{m}_1}(\{\boldsymbol{m}_1\})}\right)^{-1} \\
&= (\lambda_a+\lambda_b)\left(1+\frac{\lambda_a}{\lambda_a+\lambda_b}\right)^{-1} = \frac{(\lambda_a+\lambda_b)^2}{2\,\lambda_a+\lambda_b}, \\
\eta_{\boldsymbol{m}_2}(\{\boldsymbol{m}_1,\boldsymbol{m}_2\}) &= \eta(\{\boldsymbol{m}_1,\boldsymbol{m}_2\})\left(1+\frac{\eta_{\boldsymbol{m}_1}(\{\boldsymbol{m}_1\})}{\eta_{\boldsymbol{m}_2}(\{\boldsymbol{m}_2\})}\right)^{-1} \\
&= (\lambda_a+\lambda_b)\left(1+\frac{\lambda_a+\lambda_b}{\lambda_a}\right)^{-1} = \frac{\lambda_a(\lambda_a+\lambda_b)}{2\,\lambda_a+\lambda_b}.
\end{aligned} \tag{8}$$

The assignment probabilities can now be calculated by solving

$$\begin{aligned}
\eta_{\boldsymbol{m}_1}(\{\boldsymbol{m}_1,\boldsymbol{m}_2\}) &= \lambda_a P(a,\boldsymbol{m}_1|\{\boldsymbol{m}_1,\boldsymbol{m}_2\}) + \lambda_b P(b,\boldsymbol{m}_1|\{\boldsymbol{m}_1,\boldsymbol{m}_2\}) \\
&= \lambda_a P(a,\boldsymbol{m}_1|\{\boldsymbol{m}_1,\boldsymbol{m}_2\}) + \lambda_b\,, \\
\eta_{\boldsymbol{m}_2}(\{\boldsymbol{m}_1,\boldsymbol{m}_2\}) &= \lambda_a P(a,\boldsymbol{m}_2|\{\boldsymbol{m}_1,\boldsymbol{m}_2\})
\end{aligned} \tag{9}$$

to obtain $\eta = P(a,\boldsymbol{m}_1|\{\boldsymbol{m}_1,\boldsymbol{m}_2\}) = \frac{\lambda_a}{2\,\lambda_a+\lambda_b}$ and $P(b,\boldsymbol{m}_1|\{\boldsymbol{m}_1,\boldsymbol{m}_2\}) = 1-\eta = \frac{\lambda_a+\lambda_b}{2\,\lambda_a+\lambda_b}$.

The stationary probabilities for System A can now be obtained from (5). After normalization the unique product-form stationary distribution for System A becomes

$$\begin{aligned}
&\pi(n, \boldsymbol{m}_2, m, \boldsymbol{m}_1) = B \left(\frac{\lambda_b}{\mu_1}\right)^m \left(\frac{\lambda_a + \lambda_b}{\mu_1 + \mu_2}\right)^n, \qquad m, n \geq 0, \\
&\pi(n, \boldsymbol{m}_1, 0, \boldsymbol{m}_2) = B \, \frac{\mu_1}{\mu_2} \left(\frac{\lambda_a + \lambda_b}{\mu_1 + \mu_2}\right)^n, \qquad n \geq 0, \\
&\pi(m, \boldsymbol{m}_1) = B \, \frac{\mu_1 + \mu_2}{\lambda_a} \left(\frac{\lambda_b}{\mu_1}\right)^m, \qquad m \geq 0, \\
&\pi(0, \boldsymbol{m}_2) = B \, \frac{\mu_1}{\mu_2} \, \frac{\mu_1 + \mu_2}{\lambda_a + \lambda_b}, \\
&\pi(0) = B \, \frac{2\,\lambda_a + \lambda_b}{\lambda_a + \lambda_b} \, \frac{\mu_1}{\lambda_a} \, \frac{\mu_1 + \mu_2}{\lambda_a + \lambda_b},
\end{aligned} \tag{10}$$

where the normalizing constant is given by

$$B = \frac{\mu_2 \, \lambda_a \, (\lambda_a + \lambda_b)^2 \, (\mu_1 - \lambda_b) \, (\mu_1 + \mu_2 - \lambda_a - \lambda_b)}{\mu_1(\mu_1 + \mu_2) \left(\mu_2^2 \, \lambda_a^2 + \mu_1 \lambda_a \, (\mu_1 - \lambda_b) \, (\lambda_a + \lambda_b) + \mu_1 \, \mu_2 \, (2\,\lambda_a + \lambda_b) \, (\mu_1 + \mu_2 - \lambda_b)\right)}. \tag{11}$$

2.2 Waiting time distribution

Consider again System A of Figure 1. From its stationary distribution (10) we can almost immediately get the waiting distributions of both job types a and b, but only in case of exponential service requirements. Hence, consider System A is in steady state and service times are exponential. Jobs of both types a and b arrive as a Poisson stream, and hence they see the queue in steady state, and find machines $\boldsymbol{m}_1$, $\boldsymbol{m}_2$ busy with probability $\pi(\cdot, \boldsymbol{m}_1, \cdot, \boldsymbol{m}_2)$.

First, note that in states (0) and $(0, \boldsymbol{m}_2)$ none of the job types have to wait, and hence the waiting time is zero. When the system is in state $(m, \boldsymbol{m}_1)$ only jobs of type b have to wait for the m jobs, which must all be of type b, already in line. Machine $\boldsymbol{m}_1$ is the only machine in System A that can process type-b jobs. Due to the FCFS processing order, machine $\boldsymbol{m}_1$ will not serve any other arriving job until the job in service, followed by the m jobs in line and finally the arriving job of type b have been processed. Hence, the conditional waiting time, given that the system is in state $(m, \boldsymbol{m}_1)$, is the same as that in an M/M/1 queue with arrival rate λ_b and service rate μ_1, and hence is exponentially distributed with rate $\mu_1 - \lambda_b$. This was hinted at by the term $(\lambda_b/\mu_1)^m$ in the expression for $\pi(m, \boldsymbol{m}_1)$.

Similarly one could argue that the waiting time for both job types a and b is exponentially distributed with rate $\mu_1 + \mu_2 - \lambda_a - \lambda_b$ when the system is in state $(n, \boldsymbol{m}_1, 0, \boldsymbol{m}_2)$. This is indeed the case, for the n jobs waiting behind the two machines are of either type and can be processed by both machines $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$, hence the waiting time is the same as that in an M/M/1 queue with arrival rate $\lambda_a + \lambda_b$ and service rate $\mu_1 + \mu_2$. Note that for type a jobs there is no difference in waiting time distribution between states $(n, \boldsymbol{m}_1, 0, \boldsymbol{m}_2)$ and $(n, \boldsymbol{m}_2, m, \boldsymbol{m}_1)$. This is due to the fact that in both cases the queue behaves as though it is an M/M/1 queue with arrival rate $\lambda_a + \lambda_b$ and service rate $\mu_1 + \mu_2$. For type-b jobs this is not true. An arriving job of type b has to wait for the m jobs of type b waiting between machines $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$, as well as the n jobs of unidentified type waiting behind the two machines. This is due to the fact that machine $\boldsymbol{m}_1$ can serve both type of jobs and therefore, due to the FCFS processing order, has to process the leftover of the n jobs at the end of the queue as well. The waiting time distribution is the sum of two exponentially distributed waiting times, one with rate $\mu_1 - \lambda_b$ and the other with rate $\mu_1 + \mu_2 - \lambda_a - \lambda_b$, as indicated by the presence of λ_b in both terms, $(\lambda_b/\mu_1)^m$ and $\big((\lambda_a + \lambda_b)/(\mu_1 + \mu_2)\big)^n$, while λ_a appears in just one of the terms.

In summary, the Laplace-Stieltjes transforms (LST's) of the steady-state waiting times W_a and W_b of type-a and type-b jobs respectively are equal to

$$\begin{aligned}\mathbb{E}(e^{-s\,W_a}) = &\left(\pi(0) + \pi(0, \boldsymbol{m}_2) + \sum_{m=0}^{\infty} \pi(m, \boldsymbol{m}_1)\right) \cdot 1 \\ &+ \left(\sum_{m=0}^{\infty}\sum_{n=0}^{\infty} \pi(n, \boldsymbol{m}_2, m, \boldsymbol{m}_1) + \sum_{n=0}^{\infty} \pi(n, \boldsymbol{m}_1, 0, \boldsymbol{m}_2)\right) \cdot \frac{\mu_1 + \mu_2 - \lambda_a - \lambda_b}{\mu_1 + \mu_2 - \lambda_a - \lambda_b + s},\end{aligned} \tag{12}$$

and

$$\begin{aligned}\mathbb{E}(e^{-s\,W_b}) = &\left(\pi(0) + \pi(0, \boldsymbol{m}_2)\right) \cdot 1 \\ &+ \sum_{m=0}^{\infty} \pi(m, \boldsymbol{m}_1) \cdot \frac{\mu_1 - \lambda_b}{\mu_1 - \lambda_b + s} \\ &+ \sum_{n=0}^{\infty} \pi(n, \boldsymbol{m}_1, 0, \boldsymbol{m}_2) \cdot \frac{\mu_1 + \mu_2 - \lambda_a - \lambda_b}{\mu_1 + \mu_2 - \lambda_a - \lambda_b + s} \\ &+ \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} \pi(n, \boldsymbol{m}_2, m, \boldsymbol{m}_1) \cdot \frac{\mu_1 - \lambda_b}{\mu_1 - \lambda_b + s} \frac{\mu_1 + \mu_2 - \lambda_a - \lambda_b}{\mu_1 + \mu_2 - \lambda_a - \lambda_b + s},\end{aligned} \tag{13}$$

Consider again the general K machine system in steady-state. A job of arbitrary type $c \in \mathcal{C}$ arrives to find the system in state $\mathfrak{s} = (n_i, M_i, n_{i-1}, M_{i-1}, \dots, n_1, M_1) \in \mathcal{S}$. It has to wait only if all machines that can serve type-c jobs are busy, hence if $c \in \mathcal{U}(\{M_1, \dots, M_i\})$, otherwise the waiting time is zero. If it has to wait, it waits a sum of exponential waiting times. There is a term for each $j = 1, \dots, i$ if among the n_j jobs waiting between machines M_j and M_{j+1}, there may be jobs of type c, hence there is a term only if $c \in \mathcal{U}(\{M_1, \dots, M_j\})$. The rate is then equal to $\mu_{\{M_1,\dots,M_j\}} - \lambda_{\mathcal{U}(\{M_1,\dots,M_j\})}$. In summary, the LST of the steady-state waiting time W_c is equal to

$$\begin{aligned}\mathbb{E}(e^{-s\,W_c}) = &\sum_{\substack{\mathfrak{s}\in\mathcal{S} \\ c\notin\mathcal{U}(\{M_1,\dots,M_i\})}} \pi(\mathfrak{s}) \cdot 1 \\ &+ \sum_{\substack{\mathfrak{s}\in\mathcal{S} \\ c\in\mathcal{U}(\{M_1,\dots,M_i\})}} \pi(\mathfrak{s}) \cdot \sum_{\substack{j=1 \\ c\in\mathcal{U}(\{M_1,\dots,M_j\})}}^{i} \frac{\mu_{\{M_1,\dots,M_j\}} - \lambda_{\mathcal{U}(\{M_1,\dots,M_j\})}}{\mu_{\{M_1,\dots,M_j\}} - \lambda_{\mathcal{U}(\{M_1,\dots,M_j\})} + s}.\end{aligned} \tag{14}$$

2.3 Mean waiting time

We just derived the distribution of the waiting time in case of exponential service times, however to be able to approximate the waiting time in case of general service requirements, we need only its mean. The average waiting time can of course be calculated from the LST. Then for exponential service times, we have

$$\mathbb{E}(W_c) = \sum_{\substack{\mathfrak{s}\in\mathcal{S} \\ c\in\mathcal{U}(\{M_1,\dots,M_i\})}} \pi(\mathfrak{s}) \cdot \sum_{\substack{j=1 \\ c\in\mathcal{U}(\{M_1,\dots,M_j\})}}^{i} \frac{1}{\mu_{\{M_1,\dots,M_j\}} - \lambda_{\mathcal{U}(\{M_1,\dots,M_j\})}} . \tag{15}$$

The mean waiting time can also be obtained from the Pollaczek-Khinchine mean value formula. By PASTA we know that an arriving job of arbitrary type $c \in \mathcal{C}$ with probability $\pi(\mathfrak{s})$ finds the system in state $\mathfrak{s} = (n_i, M_i, n_{i-1}, M_{i-1}, \dots, n_1, M_1) \in \mathcal{S}$. Again, it has to wait only if $c \in \mathcal{U}(\{M_1, \dots, M_i\})$, otherwise the waiting time is zero. There is a waiting time term for each $j = 1, \dots, i$ only if $\in \mathcal{U}(\{M_1, \dots, M_j\})$. The arriving job first has to wait for the residual service times of the j jobs in service and then continues to wait for the departure of all n_j jobs which were already waiting in the queue upon arrival. An inter-departure time is the minimum of j exponential (residual) service times with means $1/\mu_{M_j}$, and hence is exponential with mean $1/\mu_{\{M_1,\dots,M_j\}}$. Hence,

$$\mathbb{E}(W_c) = \sum_{\substack{\mathfrak{s}\in\mathcal{S} \\ c\in\mathcal{U}(\{M_1,\dots,M_i\})}} \pi(\mathfrak{s}) \cdot \sum_{\substack{j=1 \\ c\in\mathcal{U}(\{M_1,\dots,M_j\})}}^{i} \frac{\mathbb{E}\left(R_{\{M_1,\dots,M_j\}}\right)}{1-\rho_{\{M_1,\dots,M_j\}}}, \tag{16}$$

where $\rho_{\{M_1,\dots,M_j\}} = \lambda_{\mathcal{U}(\{M_1,\dots,M_j\})}/\mu_{\{M_1,\dots,M_j\}}$ and $\mathbb{E}(R_{\{M_1,\dots,M_j\}}) = 1/\mu_{\{M_1,\dots,M_j\}}$.

Note that, if we multiply both the numerator and the denominator by $\mu_{\{M_1,\dots,M_j\}}$ for each $j = 1, \dots, i$, we find that both formulas for the mean waiting time are equivalent.

The $1/\mu_{\{M_1,\ldots,M_j\}}$ terms in this equation can be decomposed in terms of the mean residual service times of each job. With probability $\mu_{M_1}/\mu_{\{M_1,\ldots,M_j\}}$ machine M_1 will have completed its current task first, and hence can serve the next job in line. The residual service time in that case is exponential with mean $1/\mu_{M_1}$. With probability $\mu_{M_2}/\mu_{\{M_1,\ldots,M_j\}}$ machine M_2 will have completed its current task first, and hence residual service is exponential with mean $1/\mu_{M_2}$, etc. After normalization we obtain

$$\mathbb{E}\left(R_{\{M_1,\ldots,M_j\}}\right) = \frac{1}{\mu_{\{M_1,\ldots,M_j\}}} = \frac{1}{j}\left(\frac{\mu_{M_1}}{\mu_{\{M_1,\ldots,M_j\}}}\frac{1}{\mu_{M_1}} + \cdots + \frac{\mu_{M_j}}{\mu_{\{M_1,\ldots,M_j\}}}\frac{1}{\mu_{M_j}}\right). \tag{17}$$

Hence, the average steady-state waiting time W_c of an arriving job of arbitrary type $c \in \mathcal{C}$ in case of exponential service requirements can be obtained from

$$\mathbb{E}(W_c) = \sum_{\substack{\mathfrak{s}\in\mathcal{S} \\ c\in\mathcal{U}(\{M_1,\ldots,M_i\})}} \pi(\mathfrak{s}) \cdot \sum_{\substack{j=1 \\ c\in\mathcal{U}(\{M_1,\ldots,M_j\})}}^{i} \frac{1}{1-\rho_{\{M_1,\ldots,M_j\}}} \cdot \frac{1}{j} \cdot \left(\sum_{k=1}^{j} \frac{\mu_{M_k}}{\mu_{\{M_1,\ldots,M_j\}}}\frac{1}{\mu_{M_k}}\right) \tag{18}$$

where $\rho_{\{M_1,\ldots,M_j\}} = \lambda_{\mathcal{U}(\{M_1,\ldots,M_j\})}/\mu_{\{M_1,\ldots,M_j\}}$ and $\pi(\mathfrak{s})$ can be obtained from (5).

Until now we have just derived the waiting time distribution and its mean in case of exponential service times. Results for general service requirements cannot so easily be obtained analytically. However, we can approximate the mean waiting time in case of general service requirements using the results for exponential service times. Experimental results suggest that the waiting probabilities are almost insensitive to processing time distributions, in that the stationary distribution depends on the service time distributions approximately only through their means. That is to say that, as long as we do not change the service rates, the stationary probabilities $\pi(\mathfrak{s})$ do not change much either. In Chapter 5 we will show by simulation that this is indeed true.

Now we explore two methods of approximation. The first is based on weighted average residual service and is explored in Chapter 3. The second is based on light traffic-heavy traffic interpolation and is explored in Chapter 4.

3 Weighted average residual service

In this section we present the first approximation formula, which will be referred to as approximation 1 or the first approximation, for the average waiting time in case of general service requirements. This approximation is based on weighted average residual service times and is inspired by the results obtained in the previous section.

Consider again the general K machine system, but now with generally distributed service requirements. Let the service rates be defined as $\mu_M = 1/\mathbb{E}(B_M)$, $M \in \mathcal{M}$, where B_M denotes the service time distribution of machine M. Recall that, for exponential service times, we have that

$$\mathbb{E}(W_c) = \sum_{\substack{\mathfrak{s}\in\mathcal{S} \\ c\in\mathcal{U}(\{M_1,\dots,M_i\})}} \pi(\mathfrak{s}) \cdot \sum_{\substack{j=1 \\ c\in\mathcal{U}(\{M_1,\dots,M_j\})}}^{i} \frac{1}{1-\rho_{\{M_1,\dots,M_j\}}} \cdot \frac{1}{j} \cdot \left(\sum_{k=1}^{j} \frac{\mu_{M_k}}{\mu_{\{M_1,\dots,M_j\}}} \frac{1}{\mu_{M_k}} \right).$$

We have seen that the last two terms in this equation form some kind of weighted average residual service, where the mean residual service time of a job in service at machine M_k in case of exponential service times is $1/\mu_{M_k}$. We can now substitute these values with the general expression $\mathbb{E}(R_M)$, where R_M denotes the residual service time at machine M, to obtain the desired approximation formula:

$$\mathbb{E}(W_{c,app}) = \sum_{\substack{\mathfrak{s}\in\mathcal{S} \\ c\in\mathcal{U}(\{M_1,\dots,M_i\})}} \pi(\mathfrak{s}) \cdot \sum_{\substack{j=1 \\ c\in\mathcal{U}(\{M_1,\dots,M_j\})}}^{i} \frac{1}{1-\rho_{\{M_1,\dots,M_j\}}} \cdot \frac{1}{j} \cdot \left(\sum_{k=1}^{j} \frac{\mu_{M_k}}{\mu_{\{M_1,\dots,M_j\}}} \mathbb{E}(R_{M_k}) \right) \quad (19)$$

4 Light traffic-heavy traffic interpolation

In this section we present the second approximation formula, which will be referred to as approximation 2 or the second approximation, for the average waiting time in case of general service requirements. This approximation is based on light traffic-heavy traffic interpolation and is inspired by the results obtained by Reiman and Simon [12].

The system's behaviour, besides service time distributions and server policies, primarily depends on the arrival rate λ. It is often useful to study the system's behaviour when the arrival rate nears its minimum or maximum value. When λ nears its minimum value, which is zero, we say that the system is in *light traffic*. Similarly, when λ nears its maximum value, which is equal to the sum of the service rates of all servers due to the system's ergodicity (1), we say that the system is in *heavy traffic*. It is often easier to look at the occupation rate ρ instead of the arrival rate λ. The occupation rate is defined as $\rho = \lambda\, \mathbb{E}(B)$. Note that in light traffic ρ nears zero and in heavy traffic ρ nears one, as $\mathbb{E}(B)$ is defined as $1/\mu$ where μ is the service rate.

Analysing the system's behaviour in light traffic and heavy traffic is often easier than analysing its behaviour in moderate traffic. In heavy traffic, for example, complex systems tend to behave the same as simple ones, such as M/M/1 or M/G/1 queues. In light traffic it is easier to visualise the situation to see how the system behaves.

Using both light traffic and heavy traffic limits, one could consider using a Taylor series to approximate the waiting time for all traffic intensities. Reiman and Simon did exactly that for several types of systems with Poisson input. Here we use the same techniques to approximate the mean waiting time of System A and System B.

The proposed approximation formula for the mean waiting time is equal to

$$\mathbb{E}\left(W_{c,app}\right) = \frac{A + B\,\rho + C\,\rho^2}{1-\rho}\,, \qquad c \in \mathcal{C} \tag{20}$$

where A, B and C are constants. These constants can be determined by simply imposing the requirements that the approximation formula results in the same mean waiting time for $\rho = 0$ as the light traffic limit, and for $\rho \to 1$ as the heavy traffic limit.

Using only light traffic information leads to an approximation that may be quite poor in moderate and heavy traffic. For example, the average waiting time increases to infinity as $\rho \to 1$, so no polynomial will approximate it well. Heavy traffic theory shows that $\lim_{\rho\to 1}(1-\rho)\,\mathbb{E}(W)$ has a finite nonzero limit. Therefore, if we "normalize" $\mathbb{E}(W)$ by $(1-\rho)$, a polynomial may be a much more reasonable approximation.

First, consider a system in heavy traffic, that is, as the occupation rate ρ nears one. All servers work almost constantly, hence an arriving job virtually never has a choice of server. Intuitively the system then behaves as though it is an M/G/1 queue with arrival rate $\lambda_{\mathcal{C}}$ and service rate $\mu_{\mathcal{M}}$, which in turn implies that all job types have the same limit. Simulation results (Section 5.2) show that this is indeed the case.

In an M/G/1 queue the (unnormalized) average waiting time is equal to

$$\mathbb{E}(W) = \frac{\rho\,\mathbb{E}(R)}{1-\rho}. \tag{21}$$

Hence, the heavy traffic limit of the normalized waiting time process should equal

$$(1-\rho)\mathbb{E}\big(W_{app}\big)\big|_{\rho=1} = (1-\rho)\mathbb{E}(W)|_{\rho=1} = \mathbb{E}(R)\,, \tag{22}$$

where $\mathbb{E}(R)$ is the mean residual service time of all servers put together. Note that $\mathbb{E}(R)$ can be obtained from (17). One could check that $(1-\rho)\,\mathbb{E}(W_c)$, where $\mathbb{E}(W_c)$ is given by (19), indeed converges to the same limit for all types of jobs:

$$(1-\rho)\mathbb{E}(W_c)|_{\rho=1} = \frac{1}{j}\cdot\left(\sum_{k=1}^{j}\frac{\mu_{M_k}}{\mu_{\{M_1,\dots,M_j\}}}\mathbb{E}\big(R_{M_k}\big)\right) = \mathbb{E}\left(R_{\{M_1,\dots,M_j\}}\right). \tag{23}$$

Now consider a system in light traffic. When the occupation rate ρ equals zero, there are no jobs in the system, hence the waiting time must be zero. To obtain more sensitive information, we calculate what are essentially first order derivatives. In theory, with n derivatives, this approach leads to an nth degree approximating polynomial. Here we only consider second degree approximating polynomials.

In light traffic the queue is virtually always empty. If at some point a job has to wait, its waiting time is at most one residual service time. Therefore, in light traffic, the average waiting time equals the probability that a job has to wait, $P(W > 0)$, multiplied by that one residual service time, $\mathbb{E}(R)$, at the appropriate servers.

The waiting probabilities, $P(W > 0)$, are known, as they are just equal to the sum of the steady-state probabilities of the states in which the job has to wait. As those are believed to be almost insensitive to processing time distributions, the steady-state probabilities, and hence the sum, can be obtained from (5).

The waiting times are equal to the minimum of the residual service times of all servers that can handle that type of job. Calculation of this minimum can be tricky, especially when non-standard processing time distributions are used. In that case, one could consider using simulation results to determine the limit. Though, for most common service time distributions the minimum is easily obtained.

The probability density function of the residual service time R of a server with processing time distribution B is given by

$$f_R(t) = \frac{1 - F_B(t)}{\mathbb{E}(B)}. \tag{24}$$

The distribution of the minimum can then be obtained from

$$F_{R_{min}}(t) = 1 - \prod_{\substack{i=1 \\ c \in \mathcal{C}(\boldsymbol{m}_i)}}^{K} \left(1 - F_{R_{\boldsymbol{m}_i}}(t)\right), \tag{25}$$

from which the mean residual service time easily follows.

If we now take the first Taylor coefficient of the average waiting time in light traffic, $\mathbb{E}(W) = P(W > 0)\,\mathbb{E}(R_{min})$, about $\rho = 0$, we have our first order derivative. Note that this first order derivative is different for each type of job.

Now that the system's limiting behaviour is known, the constants A, B and C in the approximation formula (20) can be determined by imposing the following requirements:

$$\begin{aligned} &\mathbb{E}\big(W_{c,app}\big)\big|_{\rho=0} = \mathbb{E}(W_c)|_{\rho=0}\,, \\ &\frac{d}{d\rho}\,\mathbb{E}\big(W_{c,app}\big)\big|_{\rho=0} = \frac{d}{d\rho}\,\mathbb{E}(W_c)|_{\rho=0}\,, \\ &(1-\rho)\mathbb{E}\big(W_{c,app}\big)\big|_{\rho=1} = (1-\rho)\mathbb{E}(W_c)|_{\rho=1}\,, \end{aligned} \tag{26}$$

where the first two requirements represent a system in light traffic and the last requirement represents a system in heavy traffic.

5 Simulation results

In this section we describe the means which were used to simulate the system and we analyse the results. The results show that the waiting probabilities are almost insensitive to processing time distributions and that in heavy traffic all job types have the same limit, which is equal to the average waiting time in an M/G/1 queue. The results also show that both approximation formulas perform quite well for some common service time distributions and particular choices of the assignment distribution, where the first approximation performs slightly better than the second one, probably due to its order.

Simulation of System A and System B of Figure 1 was done in Java, creating jobs, assigning them to specified servers and serving them until some time has passed, collecting data such as the waiting time along the way. Servers were created using the specified set of job types they can handle. Inter-arrival times for each job type were chosen randomly from an exponential distribution with rates λ_c , $c \in \mathcal{C}$, which were chosen such that (1) holds. Departure times were chosen randomly from an exponential, a deterministic and a uniform distribution with rates $\mu_{\boldsymbol{m}_i}$, $\boldsymbol{m}_i \in \mathcal{M}$.

Upon arrival of a job it is either assigned to a server randomly from the assignment distribution, after which its departure is planned, or it joins the end of a single queue, after either of which a new job of that type along with its arrival time is created. Upon departure of a job, either the longest waiting compatible job, according to the FCFS processing order, is taken into service or the server becomes idle.

Upon taking a job into service, its waiting time, current time minus arrival time, is recorded. After each loop, the average waiting times and waiting probabilities are calculated and stored in a text file.

Finally, calculation of residual service times, assignment probabilities, stationary probabilities and approximation formulas was done in Mathematica, after which the corresponding expressions were directly implemented in Java. The graphics were also created in Mathematica using simulation results.

The following values for λ_c and $\mu_{\boldsymbol{m}_i}$ were chosen:

System A:
$\mu_{\boldsymbol{m}_1} = 2, \quad \mu_{\boldsymbol{m}_2} = 1$
$\lambda_a = [1/9, \dots, 17/9]$ with steps of $1/9$
$\lambda_b = [1/18, \dots, 17/18]$ with steps of $1/18$

System B:
$\mu_{\boldsymbol{m}_1} = 3, \quad \mu_{\boldsymbol{m}_2} = 2, \quad \mu_{\boldsymbol{m}_3} = 1$
$\lambda_a = [7/144, \dots, 497/144]$ with steps of $7/144$
$\lambda_b = [1/48, \dots, 71/48]$ with steps of $1/48$
$\lambda_c = [1/72, \dots, 71/72]$ with steps of $1/72$

Furthermore, a maximum runtime of one million units was used.

5.1 Insensitivity assumption

Both approximation formulas for the mean waiting time assume that the steady-state waiting probabilities are almost insensitive to processing time distributions. While we will not prove that this is true, we will show that simulation results suggest that it is.

First, the exact results were compared to simulation results for exponential service times, which should be exact. We used the mean squared error (MSE) as a measure of similarity between the probabilities. The MSE in this case is obtained from

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (\hat{\pi}_i - \pi_i)^2 , \tag{27}$$

where π_i represents the data, $\hat{\pi}_i$ the exact result and n is the number of observations (number of lambdas used during simulation). In this case the MSE should equal zero. The following data was observed:

System A:
$MSE_a = 0.000245$
$MSE_b = 0.000298$

System B:
$MSE_a = 0.000265$
$MSE_b = 0.000014$
$MSE_c = 0.000138$

Table 1 *Mean squared error of the waiting probabilities for exponential service times.*

These values will function as a benchmark for the deterministic and uniform cases as they represent the simulation error. Now we compare the simulation results for deterministic and uniform service requirements to the exact results using the MSE. Note that the MSE should be close to zero for the assumption to hold for these cases.

	System A		System B		
	MSE_a	MSE_b	MSE_a	MSE_b	MSE_c
Uniform	0.000148	0.000265	0.000146	0.000009	0.000146
Deterministic	0.000092	0.000266	0.000076	0.000009	0.000146

Table 2 *Mean squared error of the waiting probabilities for other service requirements.*

5.2 Heavy traffic limit

In Chapter 4 we assumed that, in heavy traffic, the system behaves as though it is an M/G/1 queue, hence the normalized waiting time should equal the mean residual service time. These mean residual service times for the system are presented below:

	Exponential	Deterministic	Uniform
System A	0.333333	0.166667	0.222222
System B	0.166667	0.083333	0.111111

Table 3 *Mean residual service times of both systems for some service requirements.*

Figure 4 and 5 below show the normalized waiting time process of both job types for both systems in case of exponential, deterministic and uniform service requirements. Clearly, the normalized waiting time processes of both job types converge towards a single limit, which is close to the corresponding value presented in the above table. While this does not prove anything, it does support the aforementioned assumption.

In heavy traffic there appear to be tremors in the plotted functions. This is probably caused by the large uncertainty in the data in heavy traffic, due to the unnormalized average waiting time rapidly increasing to infinity. A longer runtime is needed to stabilise the waiting time process in heavy traffic, though in that case the simulation would take considerably longer to finish.

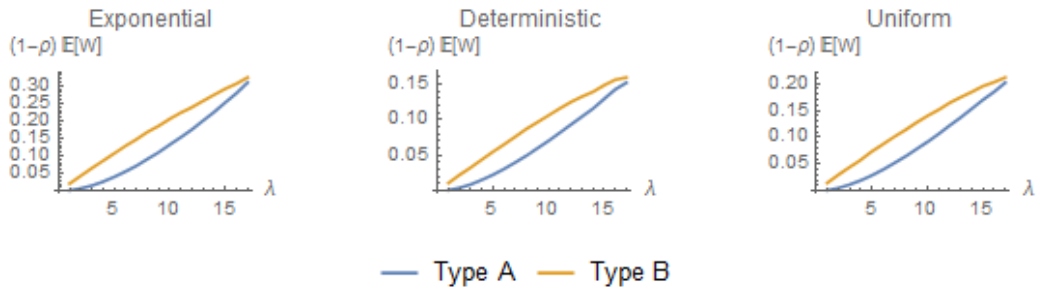

Figure 4 *Normalized waiting time process of System A in heavy traffic.*

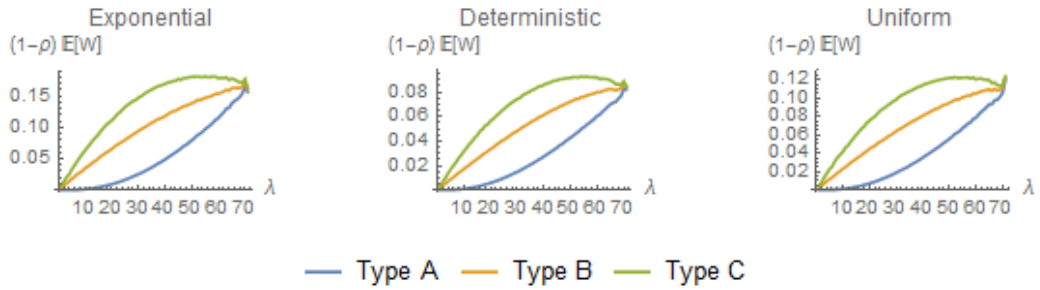

Figure 5 *Normalized waiting time process of System B in heavy traffic.*

5.3 Approximation comparison

Now that both assumptions are satisfied, we can study the performance of the approximation formulas. As a measure we again use the above defined MSE, where π_i represents the approximation result and $\hat{\pi}_i$ the data obtained by simulation. While not an absolute measure, it can be used to compare both formulas to the data and to one another. A smaller MSE suggests a better fit to the data. Values close to zero indicate a perfect fit. Note that any deviations in light traffic or heavy traffic alone are not picked up. The following data was observed:

	Exponential	Deterministic	Uniform
Type a	$MSE_{app1} = 0.001170$	$MSE_{app1} = 0.000425$	$MSE_{app1} = 0.000345$
	$MSE_{app2} = 0.007706$	$MSE_{app2} = 0.025030$	$MSE_{app2} = 0.017649$
Type b	$MSE_{app1} = 0.004019$	$MSE_{app1} = 0.000933$	$MSE_{app1} = 0.002452$
	$MSE_{app2} = 0.006293$	$MSE_{app2} = 0.026300$	$MSE_{app2} = 0.019125$

Table 4 *Mean squared error of both approximation formulas for System A.*

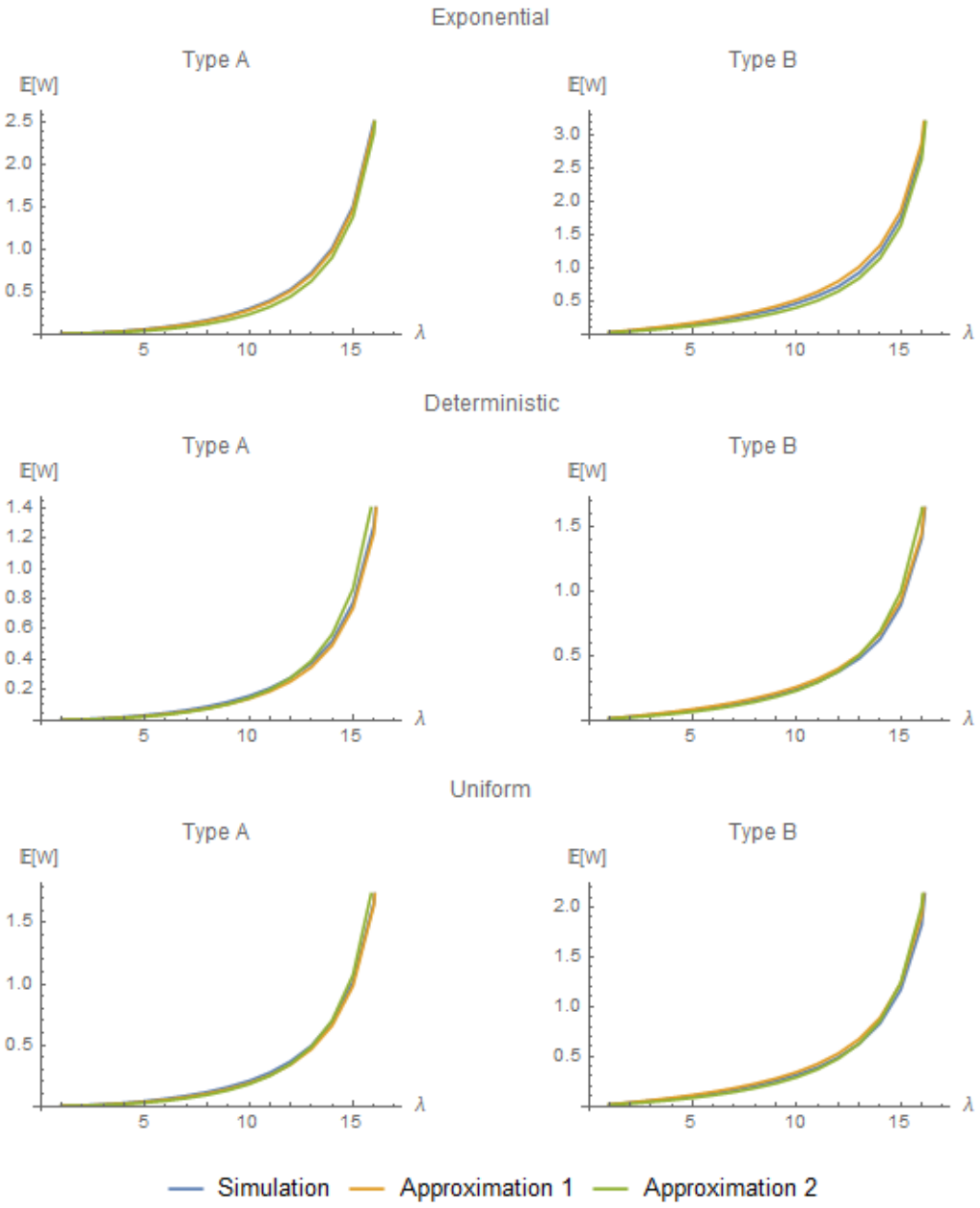

Figure 6 *Mean waiting time approximation comparison of System A.*

	Exponential	Deterministic	Uniform
Type a	$MSE_{app1} = 0.003308$ $MSE_{app2} = 0.003236$	$MSE_{app1} = 0.000598$ $MSE_{app2} = 0.109313$	$MSE_{app1} = 0.010106$ $MSE_{app2} = 0.041926$
Type b	$MSE_{app1} = 0.004095$ $MSE_{app2} = 0.009389$	$MSE_{app1} = 0.000252$ $MSE_{app2} = 0.095687$	$MSE_{app1} = 0.008811$ $MSE_{app2} = 0.030784$
Type c	$MSE_{app1} = 0.011318$ $MSE_{app2} = 0.030945$	$MSE_{app1} = 0.001263$ $MSE_{app2} = 0.090092$	$MSE_{app1} = 0.009064$ $MSE_{app2} = 0.028484$

Table 5 *Mean squared error of both approximation formulas for System B.*

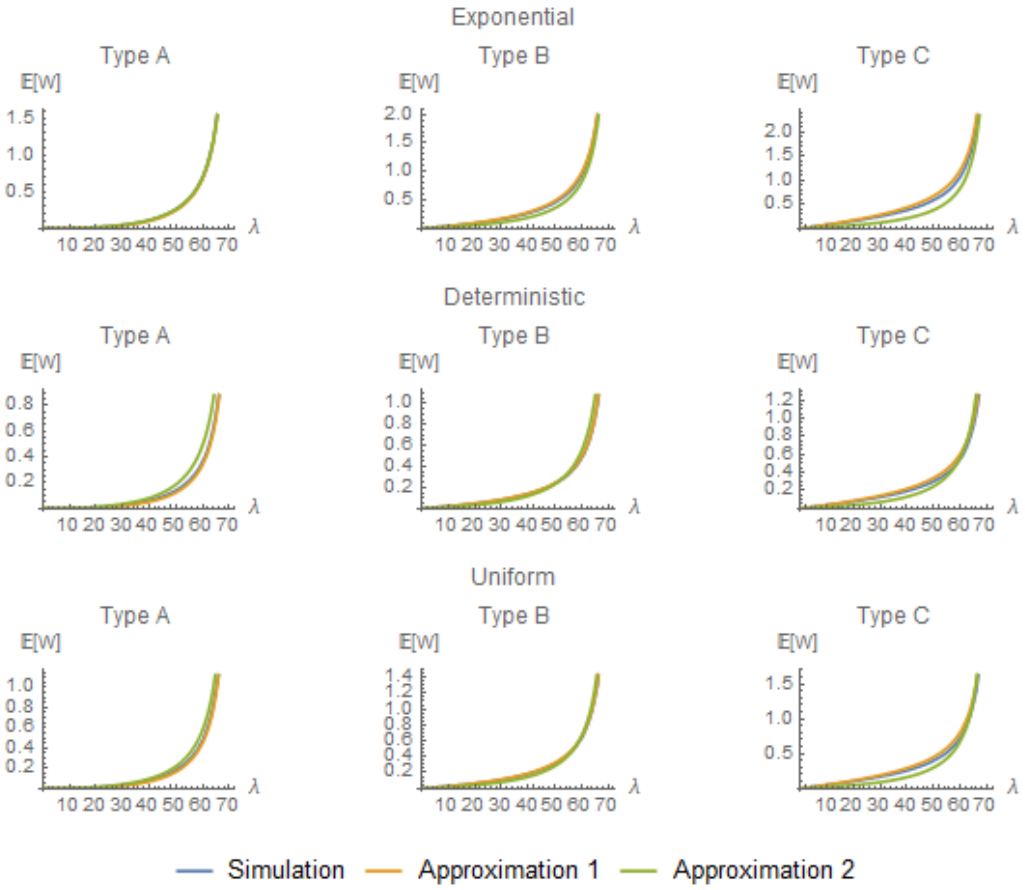

Figure 9 *Mean waiting time approximation comparison of System B.*

In all except one cases the MSE of the first approximation is (considerably) smaller than the MSE of the second approximation. This is true even if the service rates are more apart, say $\mu_{m_1} = 5$ and $\mu_{m_2} = 1$ for System A. It is clear that the approximation based on weighted average residual service fits the simulated data better than the approximation based on light traffic-heavy traffic interpolation. The difference in the one exception is so small that the 'win' is very likely caused by simulation error.

In most cases there are large differences between the errors of the two formulas. These differences are even larger when the service rates are equal. One could argue that higher order polynomials will lower the MSE and hence will fit the data better. This however remains to be tested.

Note also that the approximation performance differs for each job type. For System A type-b jobs appear to perform worse for both approximations, however for System B type-b jobs perform best, followed by type-a jobs and then type-c jobs.

6 Conclusions

The goal of this report was to find well-performing approximation formulas for the mean waiting time of skill-based parallel service systems in which service is FCFS, and arriving jobs are assigned to a feasible server randomly. Those formulas also work for systems under the combined FCFS-ALIS policy.

Two methods of approximation were tested and compared to one another; one based on weighted average residual service and the other based on light traffic-heavy traffic interpolation. Both methods assumed that the steady-state waiting probabilities, derived by Visschers, Adan and Weiss, are almost insensitive to processing time distributions. Experimental results show that this is indeed the case, at least for deterministic and uniform service requirements. The mean squared errors neared zero in all cases, indicating a near perfect fit to the data.

Simulation results also showed that in heavy traffic the system behaves as though it is an M/G/1 queue. In that case the heavy traffic limit of the normalized waiting time process should equal the mean residual service time. Plots of the waiting time process showed that indeed the normalized waiting time converges to the corresponding residual service time for all job types.

Comparing both formulas we saw that the approximation based on weighted average residual service performed considerably better than the one based on light traffic-heavy traffic interpolation. The performance however is not equal across all job types, with some job types performing better than others. Perhaps another approximation which somehow corrects for the type of job will perform uniformly better.

One could also argue that using higher order polynomials for the second approximation method will yield better results. This however remains to be tested.

In conclusion both approximations perform quite well and hence can be used to find the mean waiting time of skill-based parallel service systems under these policies.

References

[1] Adan, I.J.B.F., Wessels, J., Zijm, W.H.M..: *Queueing analysis in a flexible assembly system with a job-dependent parallel structure*. Operations Research Proceedings, 1988: 551-558

[2] Aerts, J., Korst, J., Verhaegh, W.: *Load balancing for redundant storage strategies: Multiprocessor scheduling with machine eligibility*. Journal of Scheduling, 4(5): 245-257, 2001

[3] Aksin, O.Z., Armony, M., Mehrotra, V.: *The modern call-center, a multi-disciplinary perspective on operations management research*. Production and Operations Management, 16(6): 665-688, 2007

[4] Garnett, O., Mandelbaum, A.: *An introduction to skills-based routing and its operational complexities*. Faculty of Industrial Engineering and Management, Technion, Haifa 32000, Israel, 2000

[5] Van Houtum, G.J., Adan, I.J.B.F., Wessels, J., Zijm, W.H.M.: *Performance analysis of parallel identical machines with generalized shortest queue arrival mechanism*. OR Spektrum, 23(3): 411-427, 2001

[6] Armony, M.: *Dynamic routing in large-scale service systems with heterogenuous servers*. Queueing Systems: Theory and Applications, 51(3): 287-329, 2005

[7] Gurvich, I., Whitt, W.: *Service-level differentiation in many-server service systems via queue-ratio routing*. Operations Research, 58(2): 316-328, 2010

[8] Adan, I.J.B.F., Weiss, G.: *A skill based parallel service system under FCFS-ALIS: steady state, overloads, and abandonments*. Stochastic Systems, 4(1): 250-299, 2014

[9] Visschers, J.W.C.H., Adan, I.J.B.F., Weiss, G.: *A product from solution to a system with multi-type jobs and multi-type servers*. Queueing Systems: Theory and Applications, 70(3): 269-298, 2012

[10] Adan, I.J.B.F., Hurkens, C.A.J., Weiss, G.: *A reversible Erlang loss system with multitype customers and multitype servers*. Probability in the Engineering and Informational Sciences. 24(4): 535-548, 2010

[11] Adan, I.J.B.F., Foley, R.D., McDonald, D.R.: *Exact asymptotics of the stationary distribution of a Markov chain: a production model*. Queueing Systems: Theory and Applications, 62(4): 311-344, 2009

[12] Reiman, M.I., Simon, B.: *An interpolation approximation for queueing systems with Poisson input*. Operations Research. 36(3): 454-469, 1988

Appendix

A.1 Insensitivity assumption plots

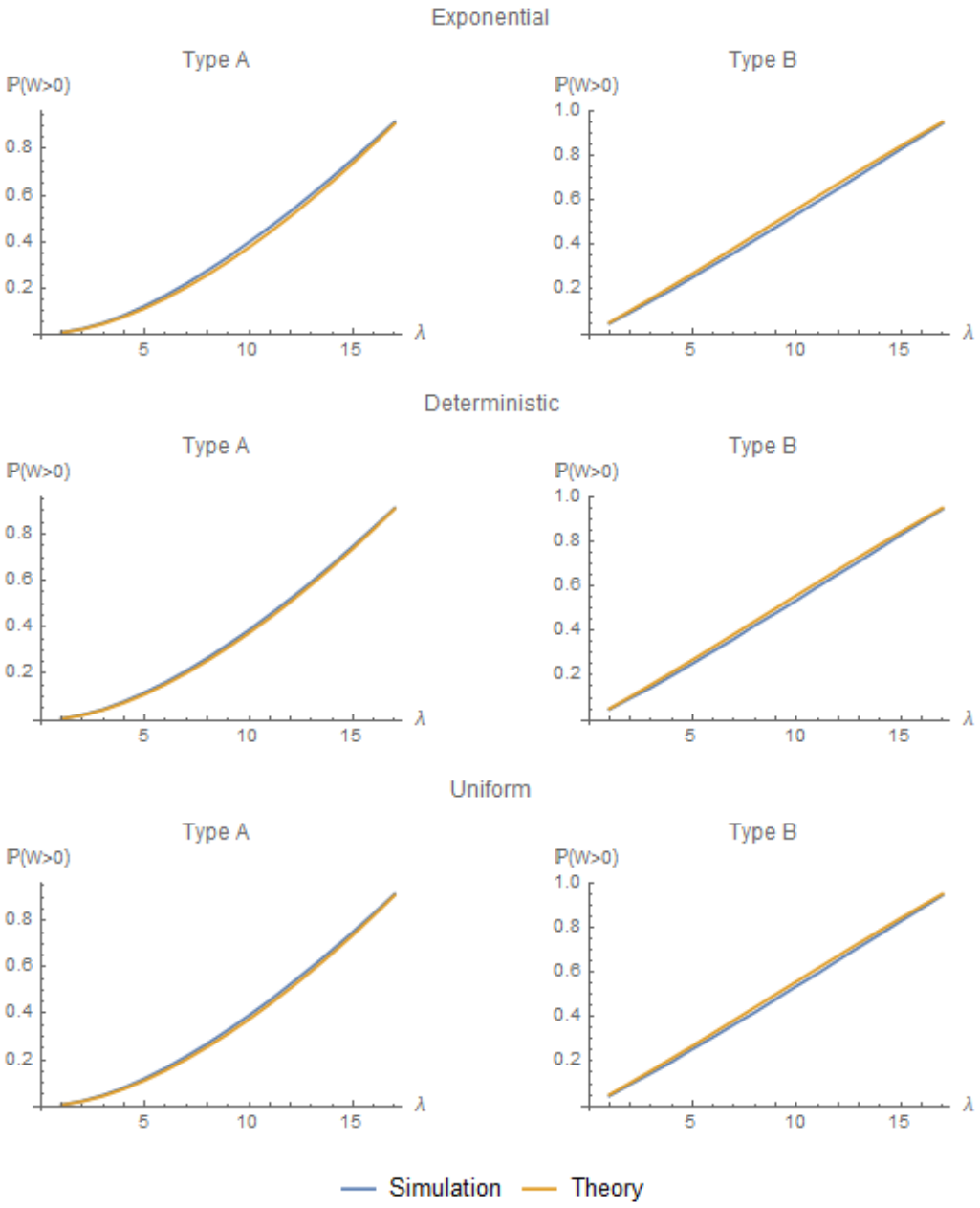

Figure 8 *Insensitivity of steady-state waiting probabilities of System A.*

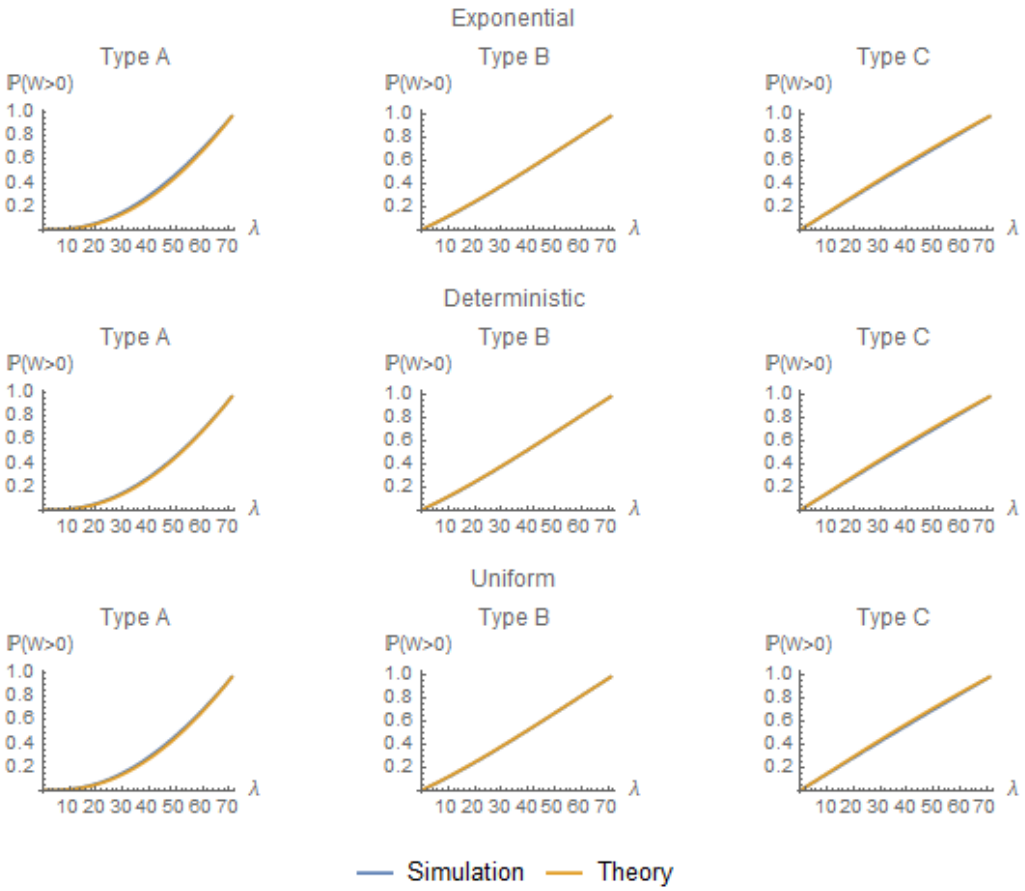

Figure 9 *Insensitivity of steady-state waiting probabilities of System B.*

A.2 Heavy traffic residual service times

For the calculation of the heavy traffic mean residual service times in Table 3 we used the following formula:

$$\mathbb{E}\left(R_{\{M_1,\ldots,M_j\}}\right) = \frac{1}{j} \cdot \left(\sum_{k=1}^{j} \frac{\mu_{M_k}}{\mu_{\{M_1,\ldots,M_j\}}} \mathbb{E}\left(R_{M_k}\right) \right), \tag{17}$$

where $\mathbb{E}(R_M) = \mathbb{E}(B_M^2)/2\mathbb{E}(B_M)$ and B_M is the processing time distribution.

As an example, consider System B where service times are uniformly distributed between 0 and b, hence $\mathbb{E}(B_M) = 1/2 \cdot b$ and $\mathbb{E}(B_M^2) = b^2/3$. Hence, $\mathbb{E}(R_M) = b/3$. Note that for $\mu_{\boldsymbol{m_1}} = 3$ we have that $b = 2/3$, for $\mu_{\boldsymbol{m_2}} = 2$ we have that $b = 1$ and for $\mu_{\boldsymbol{m_1}} = 1$ we have that $b = 2$. Hence,

$$\mathbb{E}(R) = \frac{1}{3} \cdot \left(\frac{3}{3+2+1} \cdot \frac{2}{9} + \frac{2}{3+2+1} \cdot \frac{1}{3} + \frac{1}{3+2+1} \cdot \frac{2}{3} \right) = \frac{1}{9}.$$