Assignment 2: Discrete Event Simulation

January 25, 2023

Student:
xxxx
xxxx

Course:
Stochastic Simulation

Abstract

According to queueing theory the average waiting times are shorter if the amount of servers increase. In this report we investigate this theorem by simulating several queues with Python package Simpy, by using discrete event simulation and examining the behaviour by performing statistical tests. We aim to determine which queues are decreasing rapidly in waiting time and how many time units are needed to simulate a steady system. For the latter case, we found that for a simulation time of 5000 and an occupation rate per server ρ of 0.9 and a M/M/2 queue, the estimated mean of the waiting time is 4.17 (time units) and approximates the expected mean of 4.26 with a confidence interval of $4.0794 \leq X \leq 4.2513$. Moreover, an M/M/1 queue with shortest job first scheduling has significantly lower average waiting times then a 'normal' M/M/1 with a FIFO principle. Also, M/D/c queues with FIFO have a shorter average waiting time and therefore seem preferred. Lastly, A queue where processing times are derived from a long-tail distribution shows that the latter is highly inefficient if a queue is forming. For future research we recommend to elaborate on how a nearly deterministic server capacity can be achieved in order to make the mean waiting time as short as possible.

1 Introduction

In general we want to spend least possible time in a queue. However, reducing lengths of queues and thus the waiting time is often not trivial (Adan & Resing, 2002). Queueing theory can be useful to indicate what happens if certain features of the queue change, possibly indicating how a particular queue can be reduced. More specifically, queueing theory can be used to make estimation of e.g. the mean waiting time for a customer, the mean service time, customer distribution in a queue or in a system, and so on.

One of the most remarkable although obvious facts queueing theory shows us, is that the average waiting time reduces as the amount of servers increases. In this report we will investigate this theorem; not only by making calculations with formulas derived from queueing theory, but particularly by simulating the different behaviour of several types of queues.

We will investigate whether for different types of queues the average waiting time will be shorter if the amount of servers decreases and will emphasize on what extent these waiting times will decrease. Queues with a different arrival, processing rate and distributions will be simulated and compared with each other to highlight both efficient and deficient queues. By comparing the behaviour of queues, we will use sophisticated statistical techniques and tests to be able to draw valid conclusions on their behaviour.

The report is structured as such that firstly, important queueing theories is reflected upon and explained where necessary. In particular, basic features, the notations and interpretations of these features are discussed. This is followed by the statement and explanation of significant formulas from queueing theory, e.g. the expected waiting time for particular systems. Subsequently, Discrete Event Simulation (DES) is highlighted upon, as well as the exponential distribution and the calculation of the simulated waiting time and other statistical methods. After all the background and methods used, we discuss the results. In this chapter, we elaborate upon attaining a steady state for the system, the expected waiting time for M/M/1 and M/M/c systems and the mean simulated waiting time for these systems. Subsequently, we shortly discuss the effects of shortest-job-first scheduling for M/M/1 systems and we discuss M/D/c systems, particularly in relation to systems that incorporate a long-tail distribution for the processing time (μ). Lastly, the results are discussed in the discussion section where we aim to derive plausible conclusions.

2 Queueing Theory and Methods

2.1 Basic features and notation

Queueing theory covers different types of queue's. To distinguish these different queues Kendall notation is commonly used. The notation is divided in 3 letters, a/b/c, where the first letter represents the interarrival time distribution and the second one represents the service time distribution (Adan & Resing, 2002). In these cases, M stands for the exponential distribution (or Memoryless), G for a general distribution and D for a deterministic distribution, i.e. a constant. The c denotes the amount of servers. If the amount of servers c > 1, we have a multi-server system. Particularly we will focus on M/M/c and M/D/c systems with and without several adjustments. Note that the size of the waiting line is infinite and customers are served in arrival order "first come first serve" (FIFO).

For the a modeled queue, e.g. M/M/c, an arrival rate into a system is represented by λ and $\frac{1}{\lambda}$ is the mean of the exponential distribution M. An amount of c equal servers each have an equal capacity μ and thus $\frac{1}{\mu}$ is the mean of the exponential distribution M.

With these parameters, the occupancy per server ρ can be calculated with equation 1:

$$\rho = \frac{\lambda}{c\mu} \tag{1}$$

(Adan & Resing, 2002) If only 1 server (c=1) is used $\rho < 1$.

2.1.1 Little's Law

The Little Law is a relationship between the average number of customers in a system (E(L)), the waiting time mean (E(S) and the arrival rate mean of customers (λ) (Adan & Resing, 2002). The formula is as follows:

$$E(L) = \lambda E(S) \tag{2}$$

2.1.2 Equilibrium Probabilities (M/M/c)

Probability that a job has to wait is Π_W, this is calculated by 14. In this equation n is number of customers.

$$\Pi_w = \frac{(c\rho)^c}{c!}[(1-\rho\sum_{n=0}^{c-1}\frac{(c\rho)^n}{n!}+\frac{(c\rho)^c}{c!}] \tag{3}$$

2.1.3 Mean queue length and waiting time (M/M/c)

We calculate the mean queue length with the following equation:

$$E(L^q) = \Pi_W \frac{\rho}{1-\rho} \tag{4}$$

We calculate the mean waiting time with equation 5.

$$E(W) = \Pi_W \frac{1}{1-\rho}\frac{1}{c\mu} \tag{5}$$

2.1.4 Transient Period and Steady State

Little's law only applies when the system is in steady state; i.e. the average number of people arriving must be equal to the average number of people leaving (Adan & Resing, 2002). At the very beginning of a simulation we always start with an empty system so that there is a period where the average waiting time and queue length highly depend on the samples taken from the exponential distribution of M/M/c. This period is commonly referred to as the *transient period* or the *warm-up period* (Rossetti & Delaney, 1995). Since we want to simulate a system in its steady state, we have to determine when the system reaches a steady state. Only if a system is evaluated in its steady state, Little's Law will be applicable and the formulas in section 2 will be applicable.

2.1.5 Discrete Event Simulation

In this research, Discrete Event Simulation (DES) is used to generate sample paths to characterize and investigate the behaviour of specific queue's (Fishman, 2013). This simulation approach is suitable for examining the behaviour of queues, as the latter involves a complex structure of its elements. Key elements in DES are variables and events and we run the simulations while keeping track of certain variables (Ross, 2012, p.111).

We make use of programming language Python (Python Core Team, 2019) and the built-in package Simpy (*Simpy*, 2020) to simulate the behaviour of different type of queue's. These different type of queue's are mostly M/M/c FIFO, M/M/c shortest job first, and M/D/c FIFO queues.

2.1.6 Exponential distribution

An M in Kendall notation indicates Memoryless or an exponential distribution. Random numbers derived from an exponential distribution with mean $\frac{1}{\lambda}$ or $\frac{1}{\mu}$ indicate the interarrival and processing rates of customers and servers respectively. The random values drawn from this distribution are called exponential random variables, and have a variance of $\frac{1}{\lambda^2}$ (Ross, 2012, p.27).

2.2 Statistical Methods and applications

To evaluate the accuracy of the estimation of the waiting times in the queue we apply several statistical methods and techniques. The most important techniques entail calculating the sample mean, sample variance, standard deviation and constructing a confidence interval.

2.2.1 Mean

The mean is denoted as μ and the sample mean is calculated as denoted in equation 6.

$$\overline{X} = \frac{1}{N}\sum_{i=1}^{n} X_i \tag{6}$$

2.2.2 Simulated Waiting Time

The simulated average waiting time is calculated by using equation 8.

$$\overline{W} = \frac{\sum_{n=1}^{n} t_S - t_A}{n} \tag{7}$$

Where n denotes the amount of customers that has been served or is in service, t_S denotes the time on which a customer is served and t_A is the time on which a customer arrived in the system. This average is calculated for every simulation. The mean waiting time is then calculated as:

$$\mu(\overline{W}) = \frac{\sum_{N=1}^{N} \overline{W}}{N} \tag{8}$$

2.2.3 Variance and Standard Deviation

The Variance is denoted as σ and calculated as 9

$$\sigma^2 = \sum_{i=1}^{n} \frac{(x_i - \mu^2)}{n} \tag{9}$$

Whereas the sample variance is denoted slightly different, namely as 10

$$S^2 = \sum_{i=1}^{n} \frac{(x_i - \overline{X})^2}{N-1} \tag{10}$$

Note that within the sample variance we divide by N-1 instead of N as this yields an unbiased estimator since μ is replaced by $\overline{X}$. From this, we derive the standard deviation by taking the root of the variance. Thus, the standard deviation then equals $\sqrt{\sigma^2} = \sigma$ and the standard deviation of the sample is equals $\sqrt{S^2} = S$.

2.2.4 Confidence Interval

A confidence interval is a valuable method to show the accuracy of an estimation by giving bounds (Ross, 2012). It is calculated as denoted below in equations, 11, 12 and 13. For the λ in equation 11, we commonly use 1.96 to denote a probability of 0.95.

$$a = \frac{\lambda\sigma}{\sqrt{N}} \tag{11}$$

And the confidence interval is then calculated by

$$-a + \overline{X} \leq X \leq a + \overline{X} \tag{12}$$

This results in

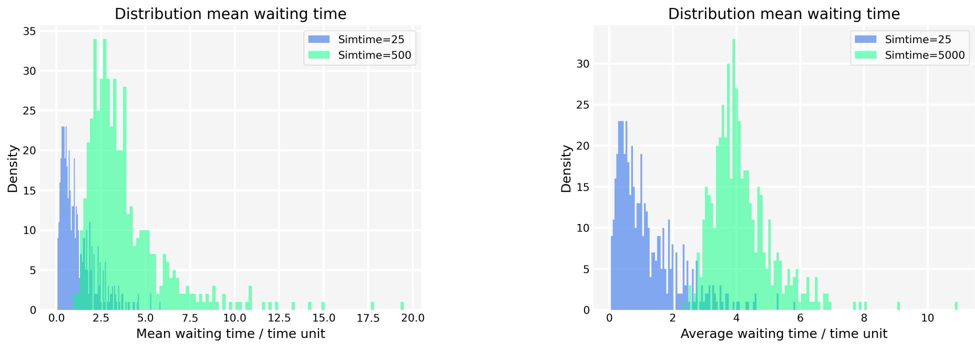
$$-\frac{\lambda\sigma}{\sqrt{N}} + \overline{X} \leq X \leq \overline{X} + \frac{\lambda\sigma}{\sqrt{N}} \tag{13}$$

σ is replaced by the mean standard deviation of the simulations, N. In Python, we use the Scipy package (Virtanen et al., 2020) to generate the λ value, given a certain percentage (usually 95).

3 Results

3.1 Steady State

The transient period or warm-up period in a queue system largely influences the behaviour of the system if the simulation time is short. This phenomenon is illustrated in figure 1. For the experiments to be useful the system needs to reach a steady state. To prevent only simulating the behaviour of the queue in the transient period, the simulation time has to be long enough such that the system is able to reach this steady state. The difference in underlying distribution of average waiting times can indicate whether we simulate a system within a transient period or a steady state.

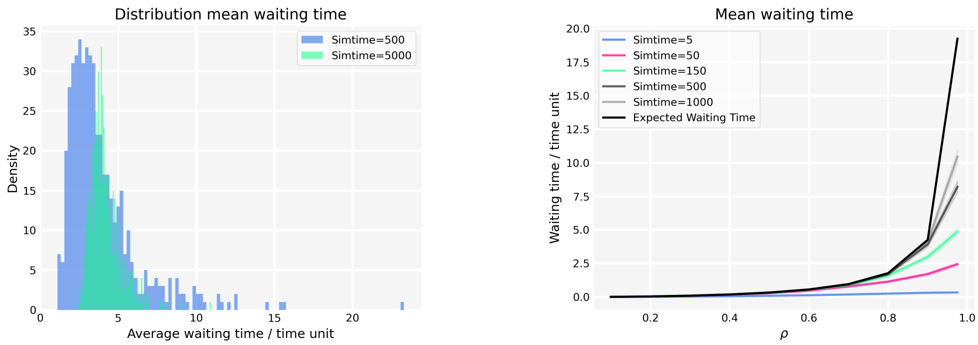


(a) ρ=0.9, c=2, μ=1, simulations=500, x-axis is avg waiting time

(b) ρ=0.9, c=2, μ=1, simulations=500, x-axis is avg waiting time

Figure 1: Distributions of average (not mean!) waiting times

With a simulation time of 25 time units, as plotted in figure 1a, the distribution of mean waiting times is highly skewed towards the left. This is, the average waiting times for a short simulation time are relatively low in comparison to a higher simulation time of 500 time units. In figure 1b, the distribution of the mean waiting time of 25 time units versus 5000 time units is plotted. The difference here is much more evident compared to figure 1a, as the distribution is also highly skewed to the left when using a small simulation time. Moreover, if we simulate the behaviour of a M/M/2 queue system for a different ρ, we get the mean average waiting time per ρ, for several different simulation times, denoted in figure 2.



(a) ρ=0.9, c=2, μ=1, simulations=500, x-axis is avg waiting time

(b) ρ=0.9, c=2, μ=1, simulations=500, x-axis is avg waiting time

Figure 2: Mean waiting times of M/M/2 with c=2. $\mu = 1$, simulation time and ρ varies, λ varies based on ρ.

The conclusion we derive from this is that the lower the simulation time, the more left-skewed the distribution of the average waiting time, and, the lower the mean waiting time is. When increasing the simulation time, the distribution also becomes more 'tight' (figure 2a). In addition, figure 2b supports the argument that for a lower simulation time a lower average waiting time occurs for *every* ρ. It is however important to note, that, the difference in average waiting times per simulation time increases as ρ increases. Thus, as the ρ increases, the simulation time makes a bigger difference. Therefore, if one wants to simulate the behaviour of a queue with a high occupancy per server (ρ), the simulation time must be high. Moreover, figure 2 thereby shows that when the ρ is increasing, the average waiting time in general increases.

We conclude from figure 1 and figure 2 that a simulation time of approximately 500 time units gives plausible results in the sense that it represents a relatively 'slim' normal distribution and that the average waiting times do not highly differ with the average waiting times generated by a higher simulation time (e.g. 1000). Also, a simulation time of 500 gives an acceptable approximation of the expected average waiting time $E(W)$ at ρ=0.9 and therefore seems suitable, especially considering the necessary computing power. Note that the slim normal distribution resembles the exponential distribution used to generate arrival and processing times and therefore seems suitable for further examination of the behaviour of the queue system.

3.2 Waiting time for M/M/1 and M/M/c

3.2.1 Expected waiting time: Mathematical explanation

The waiting time decreases as the amount of servers available increases while the total occupation rate per server (ρ) stays equal. Let c denote the amount of servers, μ the service time and ρ the occupation rate per server. For this example, let ρ and μ be a constant, e.g. ρ is 0.9 and μ is 1. This however means that the λ differs to achieve a similar ρ, since we calculate λ by rewriting the expression in equation 1. To calculate the expected waiting time $E(W)$, we first need to calculate the probability that a job has to wait (Π_W). By the PASTA property, the property "that arriving customers find on average the same situation in the queueing system as an outside observer looking at the system at an arbitrary point in time" (Adan & Resing, 2002, pp.27), it follows that the delay probability i.e. probability that a job has to wait is calculated by using equation 14.

$$\Pi_w = \frac{(c\rho)^c}{c!}[(1-\rho\sum_{n=0}^{c-1}\frac{(c\rho)^n}{n!} + \frac{(c\rho)^c}{c!}]^{-1} \tag{14}$$

And then, by making use of little's law (2.1.1) (Adan & Resing, 2002, pp.44) the expected waiting time $E(W)$ itself is calculated as

$$E(W) = \Pi_W \frac{1}{1-\rho}\frac{1}{c\mu} \tag{15}$$

then for $c = 1$, equation 14 can be written as 16

$$\Pi_{W_1} = \rho[(1-\rho)\frac{1}{1} + \rho]^{-1} = \rho \tag{16}$$

However, for $c = 2$ this becomes

$$\Pi_{W_2} = \frac{(2\rho)^2}{2!}[(1-\rho)\sum_{n=0}^{c-1}\frac{(2\rho)^n}{n!} + \frac{(2\rho)^2}{2!}]^{-1} \tag{17}$$

$$\Pi_{W_2} = 2\rho^2[(1-\rho)(1+2\rho) + 2\rho^2]^{-1} = \frac{2\rho^2}{1-\rho} \tag{18}$$

Now we can state inequality 19

$$\frac{2\rho^2}{1+\rho} < \rho \Longleftrightarrow 2\rho < 1+\rho \tag{19}$$

This inequality holds for all $0 < \rho < 1$. Therefore, $\Pi_{W_2} < \Pi_{W_1} \Longleftrightarrow 0 < \rho < 1$.

The terms in $E(W)$ (15) that determine the magnitude of the waiting time solely rely on Π_W and $\frac{1}{c\mu}$ if we assume that ρ stays equal. Both these terms decrease if c increases and if μ and ρ stay equal. Hence, the expected waiting time decreases for $c = 2$ and is thus smaller compared to the expected waiting time for $c = 1$.

The decrease in expected waiting time can also be explained intuitively; if the amount of servers increase, more customers can be served simultaneously. If more customers are served simultaneously, the queue decreases per definition and the average waiting time decreases herewith.

3.2.2 M/M/c Simulation

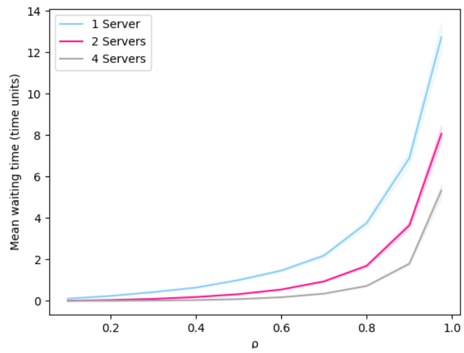


Figure 3: Mean waiting times of M/M/c with varying c between 1,2 and 4. $\mu = 1$, simulation time = 500 time units and ρ varies on x-axis, λ varies based on ρ.

Figure 3 illustrates that when the amount of servers (c) increases, the mean waiting time decreases for every occupancy rate (ρ) per server. The figure also shows an increase in waiting time as ρ increases, which is in line with the conclusion we derived in section 3.2.2.

Table 1: Simulation results for simulations = 500, simulation time = 500, $\rho = 0.1$

Servers (c)	Mean (μ)	St.dev (σ)	Confidence Interval (p = 0.95)	E[W]
1	0.10708	0.07850	$0.1002 \leq X \leq 0.1140$	0.111
2	0.01001	0.01363	$0.0088 \leq X \leq 0.0112$	0.010
4	0.00018	0.00086	$0.0001 \leq X \leq 0.0003$	0.000

Table 2: Simulation results for simulations = 500, simulation time = 500, $\rho = 0.9$

Servers (c)	Mean (μ)	St.dev (σ)	Confidence Interval (p = 0.95)	E[W]
1	7.14009	4.30842	$6.7624 \leq X \leq 7.5177$	9.000
2	3.79927	2.28734	$3.5988 \leq X \leq 3.9998$	4.263
4	1.79883	0.80527	$1.7282 \leq X \leq 1.8694$	1.969

Table 1 and table 2 show the mean, standard deviation, confidence interval and expected waiting time for $\rho = 0.1$ and $\rho = 0.9$ respectively. Remarkably, the the expected mean waiting time (E[X]) lies within the confidence interval for c=1 and c=2 and $\rho = 0.1$. For the simulations with c=1, c=2 and c=4 with a ρ of 0.9, the expected mean waiting time is higher than the simulated mean waiting time and falls not within the constructed confidence intervals. This indicates that the simulation time or the amount of simulations has to be higher in order to more accurately approximate the expected mean. Also, this finding underlines the claim that is made in section 3.2.2; if the ρ is increases, the simulation time has to increase herewith to obtain a similar accuracy, i.e. a similar statistical significance.

Table 3: Simulation results for simulations = 500, simulation time = 5000, $\rho = 0.9$

Servers (c)	Mean (μ)	St.dev (σ)	Confidence Interval (p = 0.95)	E[W]
1	8.86200	2.55865	$8.6377 \leq X \leq 9.0863$	9.000
2	4.16534	0.98059	$4.0794 \leq X \leq 4.2513$	4.263
4	1.94633	0.36864	$1.9140 \leq X \leq 1.9786$	1.969

In table 3 we observe that for a simulation time of 5000 time units the mean increases and more accurately approximates the expected mean waiting time. The expected mean waiting time now even lies within the confidence interval of the simulations where c=1 and c=4. However, for computation purposes we will continue with a simulation time of 500, as this already gives a good approximation of the mean waiting time.

3.3 M/M/1: Shortest Job First Scheduling

In figure 4 the difference between a regular M/M/1 queue and an M/M/1 queue with shortest job first scheduling are plotted.

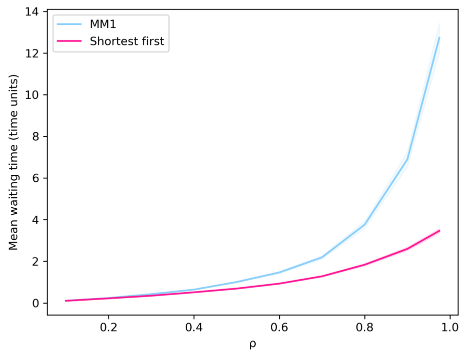


Figure 4: Mean waiting times of M/M/1 in comparison to MM1 with shortest job first scheduling. Simulations= 500, $\mu = 1$, simulation time = 500 time units and ρ varies on x-axis, λ varies based on ρ.

The shortest job first queue leads to a smaller mean waiting time for almost every ρ. Remarkably however, similar as with the simulation times in section 3.2.2, for a smaller ρ the difference for regular M/M/1 and shortest job first M/M/1 becomes smaller. But, as ρ increases, the difference in mean waiting time increases. The increase in difference becomes exponential at approximately a ρ of 0.7, since the mean waiting time of the regular M/M/1 queue system increases rapidly at this ρ.

3.4 M/D/1 and M/D/c

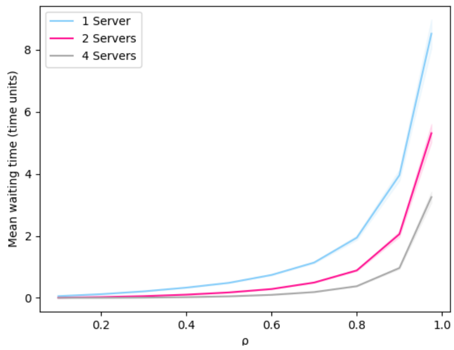
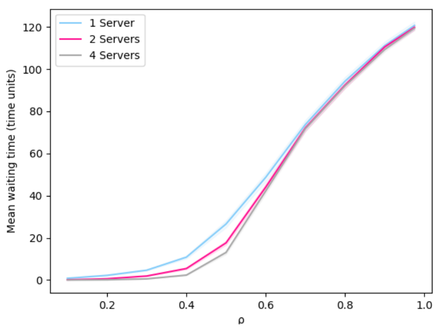


Figure 5: Mean waiting times of M/D/c with varying c between 1, 2 and 4. $\mu = 1$, simulation time = 500 time units and ρ varies on x-axis, λ varies based on ρ.

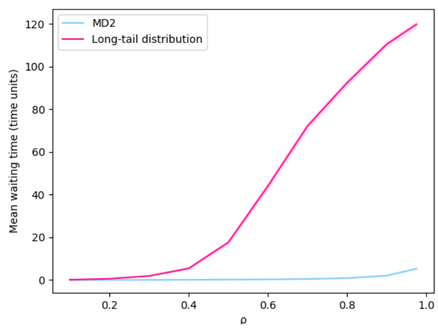
For every amount of servers (c), an M/D/c queue shows similar behaviour as M/M/c queues, but have an overall lower waiting time. For example, for an M/M/1 queue the mean waiting time reaches almost 14 for $\rho = 0.975$, while for M/D/1 this approximates only 8 time units. The shorter waiting time holds too for c=2 and c=4. Therefore we can state that for an M/D/c queue the mean waiting time is shorter compared to a M/M/c queue.

3.5 Long tail distribution

The long tail distribution is composed such that for 75% of the customers the average processing time (i.e. the capacity per server μ) is 1, and for 25% of the customers the average processing time is 5; this indicates a hyperexponential distribution.



(a) Long tail distribution with c= 1, 2, 4.



(b) M/D/2 and Long tail distribution with c=2.

Figure 6: Mean waiting times. $\mu = 1$, simulation time = 500 time units and ρ varies on x-axis, λ varies based on ρ

Figure 6a shows remarkable behaviour. Namely, as the amount of servers (c) increase, the waiting time herewith increases but the amount of servers do not seem to make a difference. The plot shows that eventually, when ρ increases, the waiting time converges to an average of 120 time units. This behaviour does not resemble the behaviour an M/M/c system, where more servers lead to a decrease in average waiting time. In figure 6b, the enormous difference between a M/D/2 system and a Long-tail distribution system with two servers is evident. An M/D/2 system already is more beneficial compared to a M/M/2 system in the sense that it's waiting times are repeatedly shorter. And if we compare an M/D/2 queue to a queue with a processing time randomly drawn from a long tail distribution, the beneficence is even greater.

4 Discussion

In this report we showed that for most queue systems, the average waiting time decreases as the amount of servers increases. The first way in which we showed that is by deriving a formula from queueing theory, we could calculate the expected average waiting time. This formula shows that when the amount of servers increases, the expected average waiting time decreases. Subsequently, we showed that for a short simulation time the calculated mean waiting times are not representative due to the transient period. Ideally, for a ρ of 0.9, a simulation time of ≥ 5000 is preferred but due to computation purposes a simulation time of 500 is used which approximates the expected mean sufficiently. The accuracy of a simulation time of 5000 is reflected in table 3, where the theoretical expected average waiting times lie within or very close to the confidence intervals constructed for a simulation time of 5000. Also worth to mention is the fact that when the ρ i.e. the occupation rate per server decreases, less simulation time is obligatory to obtain a similar accuracy. This is due to the fact that during the transient period less activity generally is observed and when the ρ is low there is less activity in general. This results in a higher accuracy in general.

To determine whether different queueing systems have a different course if ρ increases, the M/M/1 FIFO system is compared to a M/M/1 system that follows a shortest job first policy. The latter policy leads to a smaller mean waiting time compared to a FIFO M/M/1 system. However, for an increasing ρ, the waiting time slightly increases, but not as much as in a FIFO M/M/1 system.

For c=1, c=2 and c=4 a system with a deterministic processing time has a significantly smaller mean waiting time compared to a system with a processing time derived from an exponential distribution. It therefore seems more beneficial to approximate a deterministic processing time (i.e. capacity per server), if possible.

Remarkably, a queue with a processing time randomly derived from a long-tail distribution seems highly inefficient with respect to the average waiting time. As ρ approaches 1, the amount of servers seem to not even make a difference. In contrary, a M/D/c system where the processing time is deterministic seems very beneficial compared to a M/M/c system. The conclusion we derive from this is that making the processing time (nearly) deterministic could be very beneficial. In real-life, this can be done by e.g. making time slots and making them large enough, as long as a deterministic time for the process (i.e. server capacity) is held. Another option to make mean waiting times shorter is to aim for a shortest job first policy. However, as a consequence there will be plenty of customers that do not have to wait, but customers with a job that have a longer processing time will have to wait extremely long.

For future research, the type of simulated queues could be extended or more complex queues that more accurately mimic real-life queues could be simulated. In addition, one should investigate how a nearly deterministic server capacity can be achieved to make the mean waiting times as small as possible.

References

Adan, I., & Resing, J. (2002). Queueing theory.

Rossetti, M. D., & Delaney, P. J. (1995). Control of initialization bias in queueing simulations using queueing approximations. *Proceedings of the 27th conference on Winter simulation*, 322–329.

Fishman, G. S. (2013). *Discrete-event simulation: Modeling, programming, and analysis*. Springer Science & Business Media.

Ross, S. (2012). *Simulation: Fifth edition*. San Diego: Academic Press.

Python Core Team. (2019). *Python: A dynamic, open source programming language* [Python version 3.8]. Python Software Foundation. https://www.python.org/

Simpy. (2020). https://simpy.readthedocs.io/en/latest/

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2