Eindhoven University of Technology

BACHELOR

The influence of order size distributions on order picking times

Heijmans, Luuk H.C.

Award date:
2023

The influence of order size distributions on order picking times

Bachelor final project

Full Name	Student ID	Study
L.H.C. Heijmans	1021121	Applied Mathematics

Eindhoven, July 30, 2023

Abstract

In this report the effects of the distribution of order sizes will be investigated. The major quantities that are used for that are the mean service time and mean sojourn time of an order in the system. These quantities will be examined by an analytical approach and by simulation. Furthermore, the focus will be on different types of systems, for example systems can have 1 aisle or multiple aisles. Also there will be a brief look on different serving policies when there are multiple aisles in play. Eventually, conclusions are drawn for the different systems and an answer is given to the question: How does the distribution of the order sizes influence the mean service time and mean sojourn time of an order in the system.

Contents

Chapter 1

Introduction

Bol.com, Coolblue, Amazon, Zalando, Wehkamp, you name it. All these companies sell tons and tons of products every year. Most of the products, if not all of them, are purchased online by the buyer instead of in a physical store. This is easy for the customer because they do not have to leave the house but instead the product is delivered to their house the very next day. But how does the product end up on your doormat?

A lot of processes happen behind the scenes in order to get the products delivered to your house. Certain properties of the storage in a warehouse are important in order to make these processes as efficient as possible. When talking about warehouses in a mathematical way, a few keywords always arise. Think about topics like zoning (where do we place certain items in a warehouse?), batching (can an order picker take multiple items?) and routing methods (how does an order picker walk in the warehouse?). All these processes are explained in more detail by René de Koster and his colleagues [4]. A lot of research is already done on warehouses [2]. Mathematicians analyse the different aspects of the warehouses in the hope to find a solution to optimize the processes. In this report, the focus is on the part where the actual items of the order are picked inside the warehouse where these companies store their products. This process needs to be accurate and efficient. It is therefore important to reduce the time it takes to get all these items as much as possible.

Different properties of a warehouse play a role in optimizing this. But the question that this report is build upon is: how does the distribution of the size of an order effect the time it takes to pick those orders? In order to answer this question, the total pick time of an order will be analysed but also the time from when an order enters the system until it eventually is picked and leaves the system.

In chapter 2, the first model will be introduced where there is only one aisle in which all the items of an order are. This is probably not very realistic since most of the warehouses have tens if not hundreds of aisles. For this system both analytical and numerical results will be discussed. After that, the conclusions based on the results will be discussed. In chapter 3, a more realistic model with multiple aisles will be introduced. Since it is more difficult to come up with analytical results for the model with multiple aisles, the results will mostly be obtained by simulation. In the last chapter the different models will be compared with each other and possible conclusions will be made and the answer to the main question will be given.

Chapter 2

Single aisle model

In this chapter, the basic model is discussed. Since the aim of this report is to see what the impact of the order size is on the system, some performance measures will be analysed. Like the *mean service time* and the *mean sojourn time* of an order in the system. The goal is to give exact results that are found by an analytic approach. But simulation is also used in order to support the claims which are derived from the exact results.

2.1 Model description

In this model the orders, which have a size of N items, arrive according to a Poisson process with rate λ. N is not a fixed number but a random variable. The new orders will enter a queue before they can be processed. An order is processed by a single picker who can only work on one order at a time. The picker starts from a certain point, walks through the aisle to get the items that belong to the order and then walks back to the starting point. The time it takes for the picker to start walking, walk through the aisle to an item and get back to the starting point is called the *walktime*. For one item, this time (in minutes) is *Unif(0,1)*-distributed. The walktime for item i is denoted by U_i. When the picker stands in front of the storage rack to get an item, it takes a certain amount of time to get the item which is called the *picktime*. This time, in minutes, is denoted by V_i. For the total order of size N the walktime is given by $max(U_1, U_2,..., U_N)$ and the picktime is given by $V_1+V_2+...+V_N$. The system looks as follows:

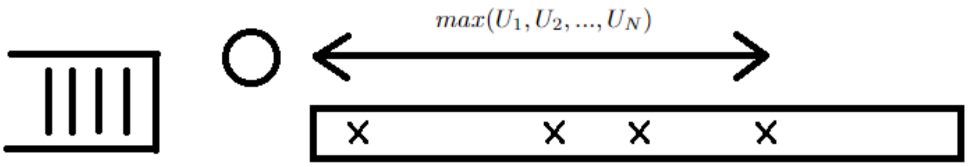


Figure 2.1: Single aisle model.

In the system above, the order has 4 different items which are going to be picked. An item is represented by X in the picture. At every item, the employee stands still to take the item with time V_i until it gets to the last item.

The total service time of an order of N items can then be written as:

$$B = max(U_1, U_2, ..., U_N) + (V_1 + V_2 + ... + V_N)$$

In queueing theory, they would call this system is a M/G/1 queue since the arrival times are modelled by a Poisson process, the service times are generally distributed and there is only 1 server. We are going to use the results of an M/G/1 queue in order to say something about the mean service time and mean sojourn time of this system. Note that rate λ and random variable N must be chosen such that the stability condition $\rho = \lambda \mathbf{E}[B] < 1$ is satisfied.

2.2 Exact calculation

From the result above it is clear that the service time depends on the order size N. In this section of the report different distributions for N will be chosen to compare the mean service times and the mean sojourn times of the orders. We assume that the order size is always bigger than 0 and that $\mathbf{E}[N]$ = c. Since N is a discrete random variable which can only attain integer values, the following distributions for N will be considered (for positive N):

- Deterministic: $\mathbf{P}(N = k) = 1$ for k = c and $\mathbf{P}(N = k) = 0$ for k $\neq$ c.
- Geo(1/c): $\mathbf{P}(N = k) = \left(\frac{c-1}{c}\right)^{k-1} \frac{1}{c}$, for k = 1,2,....
- 1 + Poisson(c-1): $\mathbf{P}(N = k) = \frac{(c-1)^{k-1} e^{-(c-1)}}{(k-1)!}$, for k = 1,2,....
- discunif(c-a, c+a), where a<c: $\mathbf{P}(N = k) = \frac{1}{2a+1}$ for k = c-a,..., c+a.

- 1 + Bin(n,$\frac{c-1}{n}$), where n≥c-1: $\mathbf{P}(N = k+1) = \binom{n}{k}\left(\frac{c-1}{n}\right)^k\left(\frac{n-c+1}{n}\right)^{n-k}$, for k = 0,...,n.

2.2.1 Mean service time

Earlier on, in section 2.1, the expression for the service time was given by $B = max(U_1, U_2, ..., U_N) + (V_1 + V_2 + ... + V_N)$. From this the mean service time is:

$$\begin{aligned}\mathbf{E}[B] &= \mathbf{E}[max(U_1, U_2, ..., U_N) + (V_1 + V_2 + ... + V_N)] \\ &= \mathbf{E}[max(U_1, U_2, ..., U_N)] + \mathbf{E}[(V_1 + V_2 + ... + V_N)] \\ &= \mathbf{E}[max(U_1, U_2, ..., U_N)] + \mathbf{E}[V]\,\mathbf{E}[N]\end{aligned}$$

The term $\mathbf{E}[V]\,\mathbf{E}[N]$ is the same for all distributions and therefore only the first term has to be compared for different distributions in order to say something about the mean service time. By the definition of the ordering of probability generating functions (see the book of Shaked and Shanthikumar [5]) we know that for two different distributed order size the following holds:

$$N_1 \leq_{pgf} N_2 \text{ if and only if } P_{N_2}(z) \leq P_{N_1}(z) \text{ for all } 0 \leq \text{z} \leq 1$$

Furthermore, we have the following lemma.
Lemma:

$$\text{if } N_1 \leq_{pgf} N_2 \text{ then } \mathbf{E}[\max(U_1,U_2,...,U_{N_1})] \leq \mathbf{E}[\max(U_1,U_2,...,U_{N_2})]$$

Proof:
Assume that $N_1 \leq_{pgf} N_2$. Then

$$\begin{aligned}\mathbf{E}[max(U_1, U_2, ..., U_{N_1})] &= \int_0^\infty (1 - \mathbf{P}(max(U_1, U_2, ..., U_{N_1}) \leq x))dx \\ &= \int_0^1 (1 - \mathbf{P}(max(U_1, U_2, ..., U_{N_1}) \leq x))dx \\ &= \int_0^1 (1 - \sum_{n=0}^\infty \mathbf{P}(N_1 = n)\mathbf{P}(max(U_1, U_2, ..., U_n) \leq x))dx \\ &= \int_0^1 (1 - \sum_{n=0}^\infty \mathbf{P}(N_1 = n)\mathbf{P}(U_1 \leq x, U_2 \leq x, ..., U_n \leq x))dx\end{aligned}$$

$$
\begin{aligned}
&= \int_0^1 (1 - \sum_{n=0}^{\infty} \mathbf{P}(N_1 = n) x^n) dx \\
&= \int_0^1 (1 - P_{N_1}(x)) dx \\
&\leq \int_0^1 (1 - P_{N_2}(x)) dx \\
&\vdots \\
&= \int_0^{\infty} (1 - \mathbf{P}(max(U_1, U_2, ..., U_{N_2}) \leq x)) dx \\
&= \mathbf{E}[max(U_1, U_2, ..., U_{N_2})]
\end{aligned}
$$

So combining the definition of the probability generation function and the lemma that is given above gives:

If $P_{N_2}(z) \leq P_{N_1}(z)$ for all $0 \leq$ z ≤ 1 then $\mathbf{E}[\max(U_1, U_2, ..., U_{N_1})] \leq \mathbf{E}[\max(U_1, U_2, ..., U_{N_2})]$

For some combinations of distributions the probability generating functions can be compared such that a conclusion can be made regarding the expected value of the maximum of the walk times. For some combinations, the expressions that are obtained are too difficult to derive any meaningful conclusions. The results are as follows:

1. Geometric and deterministic
Let $N_1 \sim Geo(\frac{1}{c})$ and $N_2 = c$. These random variables have the following probability generating functions:

$$
P_{N_1}(z) = \sum_{n=1}^{\infty} \left(\frac{c-1}{c}\right)^{n-1} \frac{1}{c} z^n = \frac{z}{c - z(c-1)} \text{ and } P_{N_2}(z) = z^c
$$

Now check if $P_{N_2}(z) \leq P_{N_1}(z)$, then the following must hold:

$$
\begin{aligned}
P_{N_2}(z) \leq P_{N_1}(z) &\iff z^c \leq \frac{z}{c - z(c-1)} \\
&\iff z^{c-1} \leq \frac{1}{c - z(c-1)} \\
&\iff z^{c-1}(c - z(c-1)) - 1 \leq 0 \\
&\iff z^c(c-1) - z^{c-1}c + 1 \geq 0
\end{aligned}
$$

The derivative of $z^c(c-1) - z^{c-1}c + 1$ is $c(c-1)(z-1)z^{c-2}$. The roots are $z = 0$ and $z = 1$ and for $0 < z < 1$ the derivative is strictly negative since

$c \geq 1$. Therefore the minimum of $z^{c-1}(c-z(c-1))-1$ on $z \in [0,1]$ is found in z $= 1$ and this minimum is equal to 0. So indeed $z^c(c-1)-z^{c-1}c+1 \geq 0$ and therefore $P_{N_2}(z) \leq P_{N_1}(z)$ for $0 \leq z \leq 1$. From this it can be concluded that $\mathbf{E}[\max(U_1,U_2,...,U_{N_1})] \leq \mathbf{E}[\max(U_1,U_2,...,U_{N_2})]$ and as a result the expected service time of order sizes that are geometrically distributed are smaller than those of deterministic order sizes.

2. Geometric and Poisson
Let $N_1 \sim Geo(\frac{1}{c})$ and $N_2 \sim Poisson(c-1)+1$. They have the following probability generating functions:

$$P_{N_1}(z) = \frac{z}{c-z(c-1)} \text{ and } P_{N_2}(z) = \sum_{n=0}^{\infty} \frac{(c-1)^{n-1}e^{-(c-1)}}{(n-1)!} z^n = ze^{1-c+z(c-1)}$$

Now check if $P_{N_2}(z) \leq P_{N_1}(z)$, then the following must hold:

$$\begin{aligned} P_{N_2}(z) \leq P_{N_1}(z) &\iff ze^{1-c+z(c-1)} \leq \frac{z}{c-z(c-1)} \\ &\iff e^{1-c+z(c-1)} \leq \frac{1}{c-z(c-1)} \\ &\iff (c-z(c-1))e^{1-c+z(c-1)} - 1 \leq 0 \end{aligned}$$

The derivative of $(c-z(c-1))e^{1-c+z(c-1)} - 1$ with respect to z is:

$$\begin{aligned} &-(c-1)e^{1-c+z(c-1)} + (c-z(c-1))(c-1)e^{1-c+z(c-1)} \\ =&(c-1-z(c-1))(c-1)e^{1-c+z(c-1)} \\ =&((c-1)(1-z))(c-1)e^{1-c+z(c-1)} \\ =&(c-1)^2(1-z)e^{1-c+z(c-1)} \end{aligned}$$

Which has one root in $z = 1$. Also this derivative is strictly positive for $0 < z < 1$. Therefore the maximum of $(c-z(c-1))e^{1-c+z(c-1)} - 1$ on $z \in [0,1]$ is in $z = 1$ which is 0. So indeed $P_{N_2}(z) \leq P_{N_1}(z)$ for $0 < z < 1$ and as a result of that $\mathbf{E}[\max(U_1,U_2,...,U_{N_1})] \leq \mathbf{E}[\max(U_1,U_2,...,U_{N_2})]$. So the expected service time of order sizes that are geometrically distributed are smaller than order sizes that are Poisson distributed.

3. Deterministic and Poisson
Let $N_1 = c$ and $N_2 \sim Poisson(c-1)+1$. They have the following probability generating functions:

$$P_{N_1}(z) = z^c, P_{N_2}(z) = ze^{1-c+z(c-1)}$$

Check if $P_{N_1}(z) \leq P_{N_2}(z)$, then the following expression must hold:

$$\begin{aligned} z^c &\leq ze^{1-c+z(c-1)} \Leftrightarrow \\ z^{c-1} &\leq e^{1-c+z(c-1)} \Leftrightarrow \\ z^{c-1} &\leq e^{(z-1)(c-1)} \Leftrightarrow \\ z &\leq e^{z-1} \end{aligned}$$

This last expression is indeed the case for $z \in [0,1]$. Therefore $P_{N_1}(z) \leq P_{N_2}(z)$. Because of that $\mathbf{E}[\max(U_1,U_2,...,U_{N_2})] \leq \mathbf{E}[\max(U_1,U_2,...,U_{N_1})]$. So the expected service time of order sizes that are Poisson distributed are smaller than order sizes that are deterministic.

4. Deterministic and discrete uniform
Let $N_1 = c$ and $N_2 \sim discunif(c-a, c+a)$. They have the following probability generating functions:

$$P_{N_1}(z) = z^c, P_{N_2}(z) = \sum_{n=-a}^{a} \frac{1}{2a+1} z^{c+n}$$

$P_{N_2}(z) - P_{N_1}(z)$ is then equal to:

$$\begin{aligned} \sum_{n=-a}^{a} \frac{1}{2a+1} z^{c+n} - z^c &= \frac{1}{2a+1}(z^{c-a} + z^{c-a+1} + ... + z^{c+a-1} + z^{c+a}) - z^c \\ &= \left(\frac{1}{2a+1}(1 + z + ... + z^{2a-1} + z^{2a}) - z^a\right) z^{c-a} \\ &= \frac{z^{c-a}}{2a+1}(1 + z + ... + z^{a-1} - 2az^a + z^{a+1} + ... + z^{2a-1} + z^{2a}) \\ &= \frac{z^{c-a}}{2a+1}(1-z)(1 + 2z + 3z^2 + ... + az^{a-1} - az^a + ... - 2z^{2a-2} - z^{2a-1}) \\ &= \frac{z^{c-a}}{2a+1}(1-z)((1 - z^{2a-1}) + (2z - 2z^{2a-2}) + ... + (az^{a-1} - az^a)) \end{aligned}$$

All these terms are positive since $z \in [0,1]$. Therefore $P_{N_2}(z) \geq P_{N_1}(z)$ and as a result of that $\mathbf{E}[\max(U_1,U_2,...,U_{N_2})] \leq \mathbf{E}[\max(U_1,U_2,...,U_{N_1})]$. The expected service time of order sizes that are discrete uniform distributed are smaller than order sizes that are deterministic.

5. Deterministic and Binomial

Let $N_1 = c$ and $N_2 \sim Bin(n, \frac{c-1}{n}) + 1$. They have the following probability generating functions:

$$P_{N_1}(z) = z^c, P_{N_2}(z) = z\left(1 + \frac{(c-1)(z-1)}{n}\right)^n$$

First take $n = c - 1$. Then

$$\begin{aligned}P_{N_2}(z) &= z\left(1 + \frac{(c-1)(z-1)}{c-1}\right)^{c-1} \\ &= z(1 + (z-1))^{c-1} \\ &= zz^{c-1} \\ &= z^c = P_{N_1}(z)\end{aligned}$$

For $n > c - 1$: Since $(1 + \frac{x}{n})^n$ is an increasing function in terms of n(proof later on) and $z \in [0, 1]$ it becomes clear that:

$$P_{N_1}(z) \leq P_{N_2}(z)$$

Therefore $\mathbf{E}[\max(U_1, U_2, \ldots, U_{N_2})] \leq \mathbf{E}[\max(U_1, U_2, \ldots, U_{N_1})]$ and the expected service time of order sizes that are Binomial distributed are smaller than order sizes that are deterministic.

Lemma: $(1 + \frac{x}{n})^n$ is increasing in n.

Proof: Let $a_n = (1 + \frac{x}{n})^n$, then:

$$\begin{aligned}\frac{a_{n+1}}{a_n} &= \frac{(1 + \frac{x}{n+1})^{n+1}}{(1 + \frac{x}{n})^n} \\ &= \frac{(1 + \frac{x}{n+1})^{n+1}}{(1 + \frac{x}{n})^{n+1}}\left(1 + \frac{x}{n}\right) \\ &= \left(1 - \frac{x}{(n+1)(n+x)}\right)^{n+1}\left(1 + \frac{x}{n}\right) \\ &\geq \left(1 - \frac{x(n+1)}{(n+1)(n+x)}\right)\left(1 + \frac{x}{n}\right) \\ &= \left(1 - \frac{x}{(n+x)}\right)\left(1 + \frac{x}{n}\right) \\ &= 1\end{aligned}$$

6. Poisson and Binomial

Let $N_1 \sim Poisson(c-1)+1$ and $N_2 \sim Bin(n, \frac{c-1}{n})+1$. They have the following probability generating functions:

$$P_{N_1}(z) = ze^{1-c+z(c-1)}, P_{N_2}(z) = z\left(\frac{n+(c-1)(z-1)}{n}\right)^n = z\left(1+\frac{(c-1)(z-1)}{n}\right)^n$$

The function $(1+\frac{x}{n})^n$ is an increasing function in terms of n and it converges to e^x. Use this and the fact that $z \in [0,1]$ to find that:

$$\begin{aligned} P_{N_2}(z) &= z\left(1+\frac{(c-1)(z-1)}{n}\right)^n \\ &\leq ze^{(c-1)(z-1)} \\ &= ze^{1-c+z(c-1)} = P_{N_1}(z) \end{aligned}$$

So indeed $P_{N_2}(z) \leq P_{N_1}(z)$ for $0 < z < 1$ and as a result of that $\mathbf{E}[\max(U_1,U_2,...,U_{N_1})] \leq \mathbf{E}[\max(U_1,U_2,...,U_{N_2})]$. So the expected service time of order sizes that are Poisson distributed are smaller than order sizes that are Binomial distributed.

7. Geometric and discrete uniform

Let $N_1 =\sim Geo(\frac{1}{c})$ and $N_2 \sim discunif(c-a, c+a)$. They have the following probability generating functions:

$$P_{N_1}(z) = \frac{z}{c-z(c-1)}, P_{N_2}(z) = \frac{z^{c-a}-z^{c+a+1}}{(c+a-(c-a)+1)(1-z)}$$

No mathematical proof is obtained here but there are numerical results which indicate that:

$$P_{N_2}(z) \leq P_{N_1}(z)$$

For example:

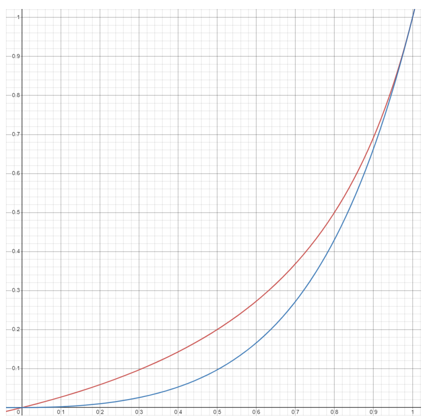


Figure 2.2: Geometric and discrete uniform pgf's.

Here c = 4 and a = 2, and we find that for $z \in [0,1]$ the blue line (discrete uniform) is always smaller than the red line (geometric). Other choices of c and a will give that the pgf of the discrete uniform distribution is lower than that of the geometric distribution.
From this result it can be conjectured that $\mathbf{E}[\max(U_1,U_2,...,U_{N_1})] \leq \mathbf{E}[\max(U_1,U_2,...,U_{N_2})]$ and therefore the expected service time of order sizes that are geometric distributed are smaller than order sizes that are discrete uniform distributed.

8. Geometric and Binomial
Let $N_1 =\sim Geo(\frac{1}{c})$ and $N_2 \sim Bin(n, \frac{c-1}{n}) + 1$. They have the following probability generating functions:

$$P_{N_1}(z) = \frac{z}{c - z(c-1)}, P_{N_2}(z) = z\left(1 + \frac{(c-1)(z-1)}{n}\right)^n$$

Also here no proof can be given but once again a graph can give an insight that:

$$P_{N_2}(z) \leq P_{N_1}(z)$$

With:

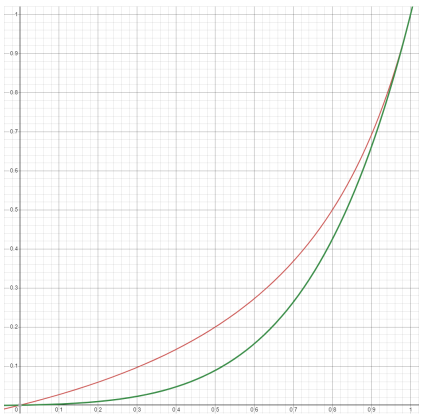


Figure 2.3: Geometric and binomial pgf's.

Here c = 4 and n = 6, the red line is the pgf of the geometric distribution and the green line is the pgf of the binomial distribution.
From this results it once again can be conjectured that $\mathbf{E}[\max(U_1,U_2,...,U_{N_1})] \leq \mathbf{E}[\max(U_1,U_2,...,U_{N_2})]$ and therefore the expected service time of order sizes that are geometric distributed are smaller than order sizes that are binomial.

9. Poisson and discrete uniform
Let $N_1 \sim Poisson(c-1)+1$ and $N_2 \sim discunif(c-a, c+a)$. They have the following probability generating functions:

$$P_{N_1}(z) = ze^{1-c+z(c-1)}, P_{N_2}(z) = \frac{z^{c-a} - z^{c+a+1}}{(c+a-(c-a)+1)(1-z)}$$

Also for this last case, there is no mathematical proof but the picture indicates that:

$$P_{N_2}(z) \leq P_{N_1}(z)$$

And:

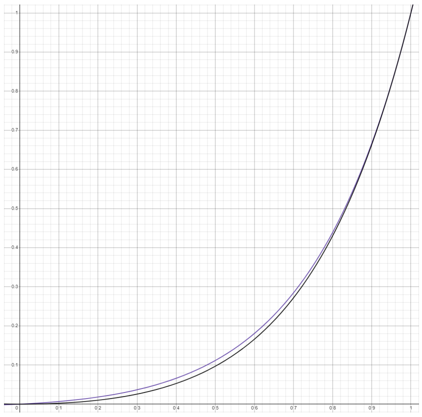


Figure 2.4: Poisson and discrete uniform pgf's.

Here c = 4 and a = 2, where the blue line is the pgf of the Poisson distribution and the black line is the pgf of the discrete uniform distribution. From this results it can be conjectured that $\mathbf{E}[\max(U_1, U_2, ..., U_{N_1})] \leq \mathbf{E}[\max(U_1, U_2, ..., U_{N_2})]$ and therefore the expected service time of order sizes that are Poisson distributed are smaller than order sizes that are discrete uniform distributed.

From these results it can be concluded that geometric distributed order sizes lead to the smallest mean service times in comparison to the deterministic, Poisson, discrete uniform and binomial distribution. The Poisson distribution is better than the deterministic, discrete uniform, binomial distribution. For some parameters, the discrete uniform gives better results than the binomial distribution but for different parameters it is the other way around. Deterministic order sizes always results in the worst possible mean service times.

2.2.2 Mean sojourn time

In the previous part, the mean service time was analysed and some exact results were found. The next step is to analyse the mean sojourn time.
For a general M/G/1 queue, the expected sojourn time is given by the following formula (See Adan et al [1]):

$$\mathbf{E}[S] = \mathbf{E}[B] + \frac{\rho}{1-\rho}\frac{\mathbf{E}[B^2]}{2\mathbf{E}[B]}$$

For the single aisle model, this expression can be rewritten as follows:

$$\begin{aligned}
\mathbf{E}[S] =& \mathbf{E}[B] + \frac{\rho}{1-\rho}\frac{\mathbf{E}[B^2]}{2\mathbf{E}[B]} \\
=& \mathbf{E}\left[max(U_1, ..., U_N)\right] + \mathbf{E}\left[V\right]\mathbf{E}\left[N\right] \\
&+ \frac{\rho}{1-\rho}\frac{\mathbf{E}[B^2]}{2(\mathbf{E}\left[max(U_1, ..., U_N)\right] + \mathbf{E}\left[V\right]\mathbf{E}\left[N\right])} \\
=& \mathbf{E}\left[max(U_1, ..., U_N)\right] + \mathbf{E}\left[V\right]\mathbf{E}\left[N\right] \\
&+ \frac{\rho}{1-\rho}\frac{Var(B) + \mathbf{E}[B]^2}{2(\mathbf{E}\left[max(U_1, ..., U_N)\right] + \mathbf{E}\left[V\right]\mathbf{E}\left[N\right])}
\end{aligned}$$

The variance of the service time can be written as:

$$Var(max(U_1, ..., U_N)) + Var((V_1 + ... + V_N)) + 2Cov(max(U_1, ..., U_N), (V_1 + ... + V_N))$$

Then again, this covariance can be written as:

$$\begin{aligned}
&\mathbf{E}[max(U_1, ..., U_N) * (V_1 + ... + V_N)] - \mathbf{E}[max(U_1, ..., U_N)]\mathbf{E}[V_1 + ... + V_N] \\
=& \sum_{n=0}^{\infty}\mathbf{P}(N=n)\mathbf{E}[max(U_1, ..., U_n) * (V_1 + ... + V_n) \mid N = n] \\
&- \mathbf{E}[max(U_1, ..., U_N)]\mathbf{E}[V_1 + ... + V_N] \\
=& \sum_{n=0}^{\infty}\mathbf{P}(N=n)\frac{n^2}{n+1}\mathbf{E}[V] - \mathbf{E}[max(U_1, ..., U_N)]\mathbf{E}[V]\mathbf{E}[N] \\
=& \mathbf{E}\left[\frac{N^2}{N+1}\right]\mathbf{E}[V] - \mathbf{E}[max(U_1, ..., U_N)]\mathbf{E}[V]\mathbf{E}[N]
\end{aligned}$$

Furthermore:

$$\mathbf{E}\left[\frac{N^2}{N+1}\right] = \mathbf{E}[N] - 1 + \mathbf{E}\left[\frac{1}{N+1}\right]$$

There is no general way to elaborate this expression. But for a specific distribution of the order size it can be elaborated further. From now on we assume that the order sizes are geometrically with parameter θ:

$$\begin{aligned}
\mathbf{E}\left[\frac{N^2}{N+1}\right] =& \mathbf{E}[N] - 1 + \mathbf{E}\left[\frac{1}{N+1}\right] \\
=& \mathbf{E}[N] - 1 + \sum_{i=1}^{\infty} \frac{1}{1+i} \theta (1-\theta)^{i-1} \\
=& \mathbf{E}[N] - 1 + \frac{\theta}{(1-\theta)^2} \sum_{i=1}^{\infty} \frac{1}{1+i} (1-\theta)^{i+1} \\
=& \mathbf{E}[N] - 1 + \frac{\theta}{(1-\theta)^2} \sum_{j=2}^{\infty} \frac{(1-\theta)^j}{j} \\
=& \mathbf{E}[N] - 1 + \frac{\theta}{(1-\theta)^2} \sum_{j=2}^{\infty} \frac{(-1)^j (\theta - 1)^j}{j} \\
=& \mathbf{E}[N] - 1 + \frac{-\theta}{(1-\theta)^2} \sum_{j=2}^{\infty} \frac{(-1)^{j-1} (\theta - 1)^j}{j} \\
=& \mathbf{E}[N] - 1 + \frac{-\theta (ln(\theta) - (\theta - 1))}{(1-\theta)^2}
\end{aligned}$$

Since the series representation of $ln(x+1)$ is equal to:

$$-\sum_{k=1}^{\infty} \frac{(-1)^k x^k}{k}$$

The only thing that is still missing to get exact results is the expectation and variance of $max(U_1, ..., U_N)$. The next part is also explained in Hao and Godbole [3]. Let $Y = max(U_1, ..., U_N)$. Then:

$$\mathbf{P}(Y \leq y | N = n) = F(y)^n$$

and the probability density function of Y is

$$g(y | N = n) = nF(y)^{n-1} f(y)$$

where $f(y)$ is the probability density function and $F(y)$ is the cumulative distribution function of U_i. Consequently, the marginal probability density function of Y is given by:

$$g(y) = \sum_{n=1}^{\infty} g(y|N=n)\mathbf{P}(N=n) = f(y) \sum_{n=1}^{\infty} nF(y)^{n-1}\mathbf{P}(N=n)$$

In our case U_i is *Unif(0,1)*-distributed:

$$g(y) = \sum_{n=1}^{\infty} ny^{n-1}\mathbf{P}(N = n)$$

In the previous part the assumption was made that the order sizes are from a geometric distribution with parameter θ. The probability density function becomes:

$$\begin{aligned}
g(y) =& \sum_{n=1}^{\infty} ny^{n-1}\theta(1-\theta)^{n-1} \\
=&\theta \sum_{n=1}^{\infty} n(y(1-\theta))^{n-1} \\
=&\theta \sum_{n=1}^{\infty} nx^{n-1}|_{x=y(1-\theta)} \\
=&\theta \sum_{n=1}^{\infty} \frac{d}{dx}x^{n}|_{x=y(1-\theta)} \\
=&\theta \frac{d}{dx}\sum_{n=1}^{\infty} x^{n}|_{x=y(1-\theta)} \\
=&\theta \frac{d}{dx}\frac{1}{1-x}|_{x=y(1-\theta)} \\
=&\frac{\theta}{(1-(y(1-\theta)))^2}
\end{aligned}$$

With this probability density function the moments of Y can be calculated by:

$$\begin{aligned}
\mathbf{E}[Y^k] =& \int_0^1 y^k \frac{\theta}{(1-(y(1-\theta)))^2}dy \\
=& \int_1^{\theta} \left(\frac{1-u}{1-\theta}\right)^k \frac{\theta}{u^2}\left(\frac{-1}{1-\theta}\right) du, \text{by substituting } y = \frac{1-u}{1-\theta}. \\
=&\frac{\theta}{(1-\theta)^{k+1}} \int_{\theta}^1 \frac{(1-u)^k}{u^2}du
\end{aligned}$$

$$=\frac{\theta}{(1-\theta)^{k+1}}\int_{\theta}^{1}\frac{\sum_{j=0}^{k}\binom{k}{j}(-u)^j}{u^2}du, \text{by Newton's binomial formula.}$$
$$=\frac{\theta}{(1-\theta)^{k+1}}\sum_{j=0}^{k}\binom{k}{j}\int_{\theta}^{1}(-u)^{j-2}du$$

From this the expectation and variance of $max(U_1, ..., U_N)$ can be obtained by filling in $k = 1$ and $k = 2$.

$$\begin{aligned}
\mathbf{E}[max(U_1, ..., U_N)] =& \mathbf{E}[Y] \\
=& \frac{\theta}{(1-\theta)^2}\sum_{j=0}^{1}\binom{1}{j}\int_{\theta}^{1}(-u)^{j-2}du \\
=& \frac{\theta}{(1-\theta)^2}\left(\int_{\theta}^{1}\frac{1}{(-u)^2}du+\int_{\theta}^{1}\frac{1}{-u}du\right) \\
=& \frac{\theta}{(1-\theta)^2}\left(\frac{1}{\theta}-1+ln(\theta)-ln(1)\right) \\
=& \frac{1-\theta+\theta ln(\theta)}{(1-\theta)^2}
\end{aligned}$$

The second moment is as follows:

$$\begin{aligned}
\mathbf{E}[Y^2] =& \frac{\theta}{(1-\theta)^3}\sum_{j=0}^{2}\binom{2}{j}\int_{\theta}^{1}(-u)^{j-2}du \\
=& \frac{\theta}{(1-\theta)^3}\left(\binom{2}{0}\int_{\theta}^{1}(-u)^{-2}du+\binom{2}{1}\int_{\theta}^{1}(-u)^{-1}du+\binom{2}{2}\int_{\theta}^{1}1du\right) \\
=& \frac{\theta}{(1-\theta)^3}\left(\frac{1}{\theta}-1+2(-ln(1)+ln(\theta))+1-\theta\right) \\
=& \frac{1-\theta^2+2\theta ln(\theta)}{(1-\theta)^3}
\end{aligned}$$

The variance of $max(U_1, ..., U_N)$ is then:

$$\begin{aligned}
Var(max(U_1, ..., U_N)) =& Var(Y) \\
=& \mathbf{E}[Y^2] - \mathbf{E}[Y]^2 \\
=& \frac{1 - \theta^2 + 2\theta ln(\theta)}{(1-\theta)^3} - \\
& \frac{1 - \theta + \theta ln(\theta) - \theta + \theta^2 - \theta^2 ln(\theta) + \theta ln(\theta) - \theta^2 ln(\theta) + \theta^2 (ln(\theta))^2}{(1-\theta)^4} \\
=& \frac{1 - \theta^2 + 2\theta ln(\theta) - \theta + \theta^3 - 2\theta^2 ln(\theta)}{(1-\theta)^4} - \\
& \frac{1 - \theta + \theta ln(\theta) - \theta + \theta^2 - \theta^2 ln(\theta) + \theta ln(\theta) - \theta^2 ln(\theta) + \theta^2 (ln(\theta))^2}{(1-\theta)^4} \\
=& \frac{\theta^3 - 2\theta^2 - \theta^2 ln^2(\theta) + \theta}{(1-\theta)^4}
\end{aligned}$$

So in the end we find:

$$\begin{aligned}
\mathbf{E}[max(U_1, ..., U_N)] =& \frac{1 - \theta + \theta ln(\theta)}{(1-\theta)^2}, \text{ and} \\
Var(max(U_1, ..., U_N)) =& \frac{\theta^3 - 2\theta^2 - \theta^2 ln^2(\theta) + \theta}{(1-\theta)^4}
\end{aligned}$$

With these expressions, exact results for the maximum of the walktimes can be found for geometric order sizes. For the total service time the variance of the picktimes is:

$$Var((V_1 + ... + V_N)) = \mathbf{E}[N]Var(V) + \mathbf{E}[V]^2 Var(N)$$

In the next part the results of the previous calculations will be used to get exact values for the expectation and variance of the service time and the mean sojourn time for geometric distributed order sizes.

2.2.3 Results for geometric order size

Earlier it turned out that for geometric distributed order sizes the mean service time and the variance of the service time have the following expressions:

$$\begin{aligned}
\mathbf{E}[B] &= \mathbf{E}\left[max(U_1, U_2, ..., U_N)\right] + \mathbf{E}\left[V\right]\mathbf{E}\left[N\right] \\
Var(B) &= \frac{\theta^3 - 2\theta^2 - \theta^2 ln^2(\theta) + \theta}{(1-\theta)^4} + \mathbf{E}[N]Var(V) + \mathbf{E}[V]^2 Var(N) \\
&+ \left(\mathbf{E}[N] - 1 + \frac{-\theta(ln(\theta) - (\theta - 1))}{(1-\theta)^2}\right)\mathbf{E}[V] - \mathbf{E}[max(U_1, ..., U_N)]\mathbf{E}[V]\mathbf{E}[N]
\end{aligned}$$

For the results the assumption is made that the picktimes are $Norm(\frac{1}{6},\frac{1}{24})$-distributed. Then for mean order sizes ranging from 1 to 7, the following results are obtained:

Mean order size	Mean service time	Variance	Standard deviation	Coefficient of variation
1	0.66666	0.08507	0.291668	0.437506
2	0.94704	0.19018	0.436096	0.460484
3	1.17604	0.34158	0.584448	0.496963
4	1.38387	0.54310	0.736953	0.532531
5	1.58038	0.79634	0.892379	0.564661
6	1.76998	1.01222	1.006090	0.568420
7	1.95496	1.46131	1.208850	0.618348

As expected, the mean service time increases as the mean order size increases. Another observation is that the variance also increases. Even so much that the coefficient of variation increases when the mean order sizes increase. This indicates that the results are more volatile as the mean order sizes increase. From these results, the mean sojourn time can be calculated which is defined by the expression given before:

$$\begin{aligned}\mathbf{E}[S] =& \mathbf{E}\left[max(U_1,...,U_N)\right] + \mathbf{E}\left[V\right]\mathbf{E}\left[N\right]\\ &+\frac{\rho}{1-\rho}\frac{Var(B)+\mathbf{E}[B]^2}{2(\mathbf{E}\left[max(U_1,...,U_N)\right] + \mathbf{E}\left[V\right]\mathbf{E}\left[N\right])}\end{aligned}$$

For these calculations the parameter for the Poisson process is equal to $\frac{1}{2}$.

Mean order size	Mean sojourn time
1	0.86522
2	1.46323
3	2.22260
4	3.37874
5	5.50529
6	10.7802
7	60.6048

Again as expected, the mean sojourn time increases as there are bigger orders. Also the mean sojourn time becomes very large when the expected order size is equal to 7. That is because the stability condition: $\rho = \lambda\mathbf{E}[B] = 0.997$, which is close to 1.

2.3 Simulation

In the previous parts it became clear that certain distributions for the order sizes are better that others in terms of the duration of the service time and sojourn time of an order. Besides the analytical view to the problem, it is also good to verify the conclusion made so far via other approaches. Therefore this problem is also simulated in RStudio. The code and a short explanation about the code is given in Appendix A.

2.3.1 Results

Different values for the mean order sizes are taken in the simulation of the service times and sojourn times in order to analyse which distribution of the order size works better.

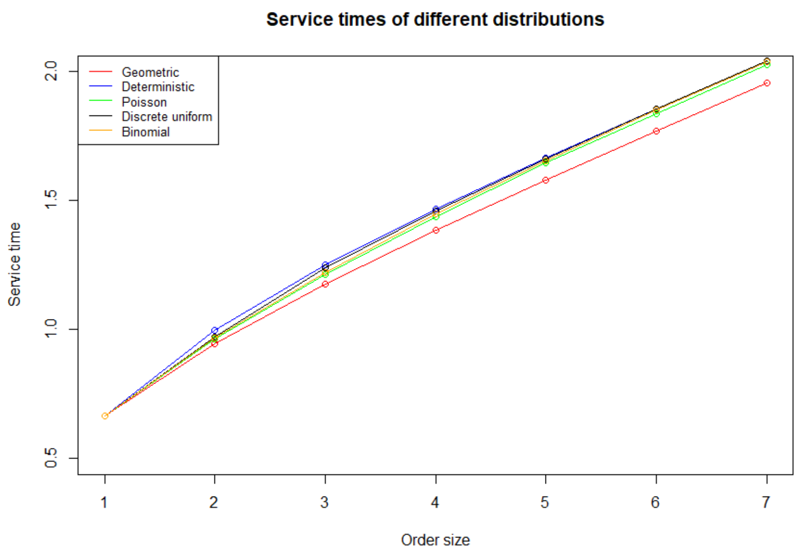


Figure 2.5: Mean service times for different distributions and different order sizes.

From Figure 2.5 it becomes clear that the geometric distribution for the order sizes results in the lowest service times. On the other hand, a deterministic value for the order sizes results in one of the worst service times. These claims were also made after the analytical view on this problem and those claims are now supported by these simulations. Also the values that were obtained for geometric distributed order sizes in the previous part come along nicely with the simulations:

Mean order size	Mean service time (Analytic)	Mean service time (Simulation)
1	0.667	0.666
2	0.947	0.947
3	1.176	1.176
4	1.384	1.384
5	1.580	1.580
6	1.770	1.770
7	1.955	1.956

With 95% confidence intervals:

Mean order size	Mean service time (Simulation)	95% Confidence interval
1	0.666	(0.658, 0.675)
2	0.947	(0.935, 0.960)
3	1.176	(1.159, 1.192)
4	1.384	(1.363, 1.405)
5	1.580	(1.554, 1.606)
6	1.770	(1.740, 1.800)
7	1.956	(1.921, 1.991)

The simulation of the sojourn times give the following results:

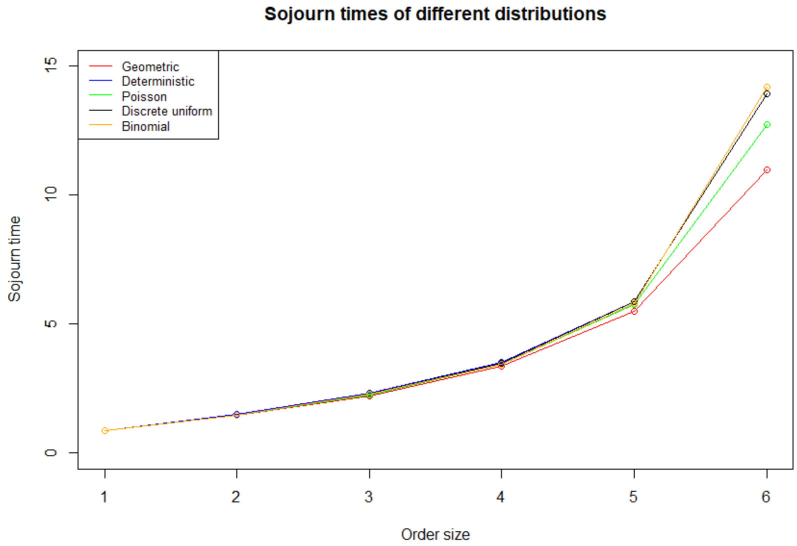


Figure 2.6: Mean sojourn times for different distributions and different order sizes.

Once again, the simulation results conclude that the geometric distribution ends up having the lowest sojourn time. For low order sizes the differences between sojourn times are around as big as the differences for the service times. But for higher order sizes, bigger differences in terms of sojourn times occur. This indicates that the point when a system becomes unstable is different for these distributions. It seems to be that binomial order sizes results in unstable system earlier than any other distribution we have looked at so far. On the other hand, the geometric distribution is doing the best in terms of the stability of the system. When the results from the geometric distribution in the previous chapter are compared to those from the simulation it once again can be concluded that they come along nicely.

Mean order size	Mean sojourn time (Analytic)	Mean sojourn time (Simulation)
1	0.865	0.865
2	1.463	1.466
3	2.223	2.221
4	3.379	3.378
5	5.505	5.464
6	10.780	10.844
7	60.605	48.205

With 95% confidence intervals:

Mean order size	Mean sojourn time (Simulation)	95% Confidence interval
1	0.865	(0.842, 0.888)
2	1.466	(1.402, 1.530)
3	2.221	(2.070, 2.372)
4	3.378	(3.043, 3.712)
5	5.464	(4.518, 6.411)
6	10.844	(7.079, 14.609)
7	48.205	(-7.381, 103.792)

2.4 Conclusion

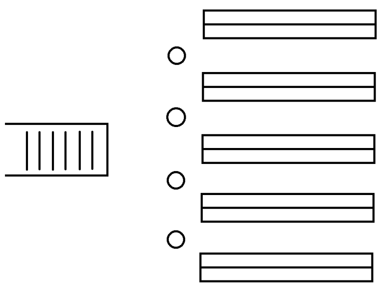
This chapter the focus was on the single aisle model. The aim was to see how the order size influences the system. For that the mean service time and the mean sojourn time were analysed in both an analytic approach and by simulations.
For the analytical approach, a few different distributions for the order sizes were taken and compared in terms of service times. The result was that the geometric distribution performed the best in comparison to a deterministic, Poisson, discrete uniform and binomial distribution. The Poisson distribution was second best, followed by the discrete uniform and binomial distribution. Lastly the deterministic distribution performed the worst. For the sojourn time, an analytical approach was difficult but eventually an expression for the geometric distribution was shown on the basis of the work of Hao and Godbole [3]. Unfortunately we could not compare it to the other distributions.
For the simulation, the same results for the mean service time was found. The geometric distribution once again performed the best followed by the Poisson distribution and then the others. For the sojourn times, the exact same order was found where the geometric distribution performed the best.

Chapter 3

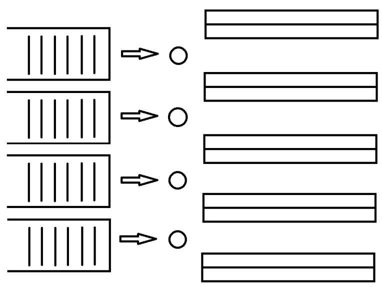
Extended models

3.1 Model descriptions

In the previous part, the basic model was discussed. In that model there was only 1 aisle from which the items can be picked. Because of that, there was only 1 employee that was picking these orders. In the next part 2 different models are analysed. Both of these models have multiple aisles and for every aisle there is one employee to pick the items of the orders. The models look as follows:



(a) Extended model 1



(b) Extended model 2

In Extended model 1, there is 1 queue in which the orders have to wait. The 'First come first serve' discipline is used here. An order is served when the previous order is completed by all the employees. This means that even when an employee is doing nothing, that employee has to wait until the other employees are ready with the current order. Once all the employee are ready to go, they will walk through their aisle to get the items for the next order. An order is full-filled when all the employees are back from their walk and all the items of the order are collected.

In Extended model 2, there is a queue for all the different aisles. When an order arrives in the system, it is known in which aisle the specific items of the order are. Then the orders are divided in 'suborders' for every queue. Furthermore the employees no longer have to wait until all employees are ready. They can just continue with the next suborder of their queue. An order is full-filled when the last suborder of an order is collected.
The basic model already showed that geometric distributed order size gave the best results in both the service times and the sojourn times of an order. In the next part the focus is not only on the distribution of the order size of an order but also on the policy in which aisle the items of an order are. 4 different policies will be analysed:

- *Random* policy. With this policy the items of an order are randomly distributed over the aisles.
- *1 aisle* policy. For this policy all the items of an order are in 1 aisle. In which aisle that is, is random.
- *Equally divided* policy. For this policy all the items will be equally divided over all aisles. There can be a difference of 1 item between the aisles if the order size is not a multiple of the amount of aisles.
- *Close aisles* policy. In this policy the items lay in aisles that are close to each other. Think about an order of mostly electronics. Probably those items are in close proximity of each other in the warehouse.

In the next part these policies are analysed for extended model 1 and extended model 2.

3.2 Extended model 1

First, let us take a look at the first extended model. Exact calculations for this model are more difficult than for the basic model. For each policy that is introduced in the previous part, the difficult points are discussed.
In this model an order is picked if all the aisles are idle. So the service time of a job can then be expressed as:

$$B = max(B_1, B_2, B_3, ..., B_n),$$

where B_i is the service time of the i-th aisle.
The mean service time for this model is then given by:

$$\mathbf{E}[B] = \mathbf{E}[max(B_1, B_2, B_3, ..., B_n)]$$

First, the cumulative distribution function of B should be calculated:

$$\begin{aligned} F_B(x) &= \mathbf{P}(B \leq x) \\ &= \mathbf{P}(max(B_1, B_2, B_3, ..., B_n) \leq x) \\ &= \mathbf{P}(B_1 \leq x, B_2 \leq x, B_3 \leq x, ..., B_n \leq x) \end{aligned}$$

If this expression is obtained, the probability density function is given by:

$$f_B(x) = \frac{d}{dx} F_B(x)$$

From this the expectation of B can be calculated.

But the expression of the cumulative distribution function of B already gives some problems. First, the distribution of the individual B_i's are unknown. This was also a problem in the basic model. This problem was tackled by taking a specific distribution for the order sizes instead of a general distribution. This can also be done in the extended model. But there is another problem. The B_i's are not independent. A short explanation for each policy is given:

- For the *random* policy: it also depends on the distribution of the order size. For example if the order size is deterministic and equal to 5, then when 1 aisle has 4 orders, the other aisles have at most 1 order.

- For the *1 aisle* policy: if one aisle has orders, the other aisles are empty.

- For the *Equally divided* policy: once again, if one aisle has many orders, the other aisles have many orders as well since they are equally divided.

- For the *Close aisles* policy: if one aisle has orders, the aisles that are more than one aisle away have no orders.

Only for the *1 aisle* policy the problem can be simplified. Since only one aisle is occupied at a time and all the servers of the aisle work evenly fast, you can say that this is actually just an $M/G/1$ queue like the basic model. So the service time should be the same as in the basic model when the expected order size are equal for both models.

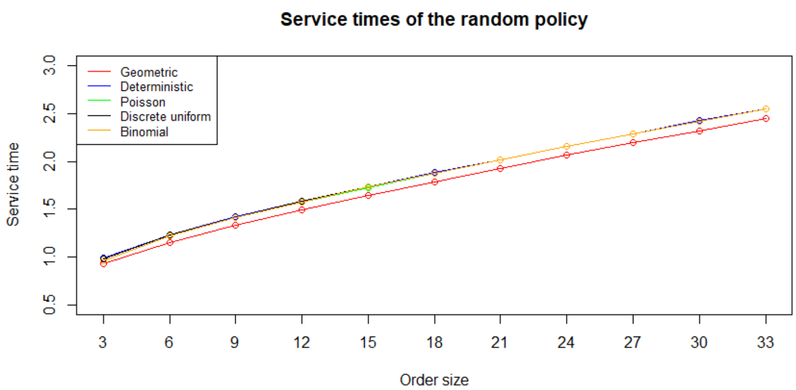
In the next part the mean service and mean sojourn time of the different policies will be compared at the hand of simulation. Also the claim that for the *1 aisle* policy the mean service time will be the same as in the basic model will be verified.

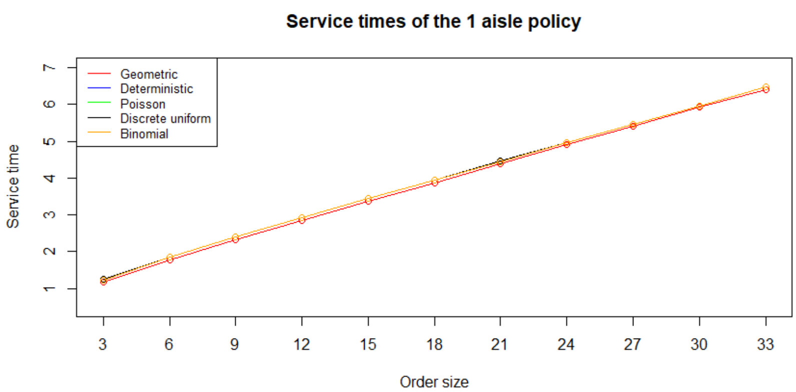
3.2.1 Simulation

Now simulation will be used to see if this claim is correct. The code in *RStudio* an a short explanation can be found in Appendix B.

Results

Now we take a look at the results. First the individual policies will be discussed for different order size distributions. There are some parameters that are going to be fixed in order to do the simulation. The arrival rate is equal to $\frac{1}{2}$. For all policies the expected order size will vary. Also, the number of aisles is equal to 5. The picktimes are still $Norm(\frac{1}{6}, \frac{1}{24})$-distributed. Running the simulation with these values gives the following graphs:





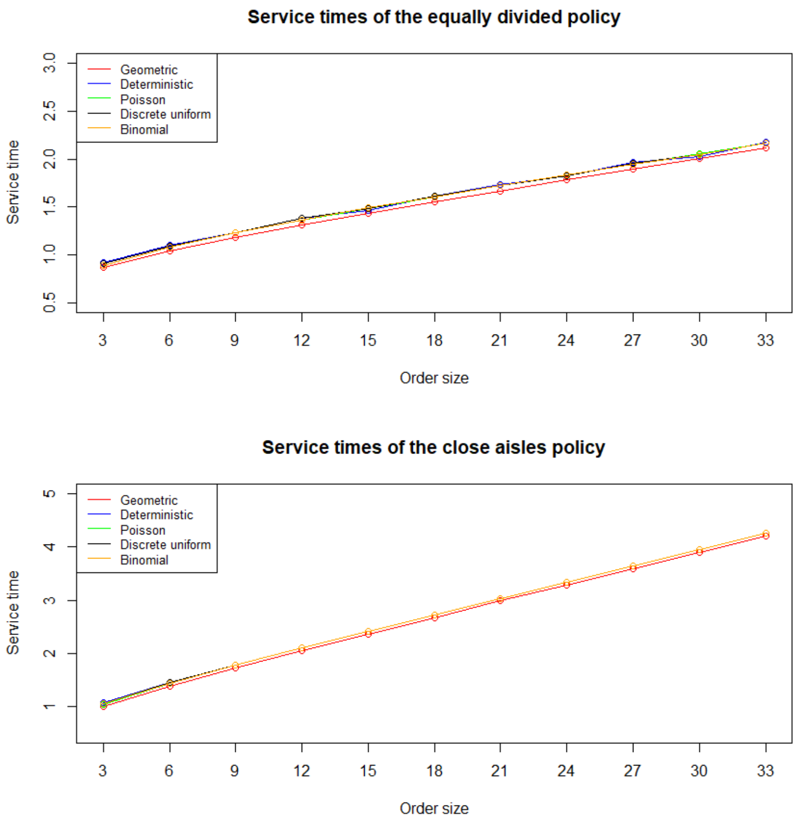


Figure 3.3: Mean service times for different policies.

And for the sojourn times:

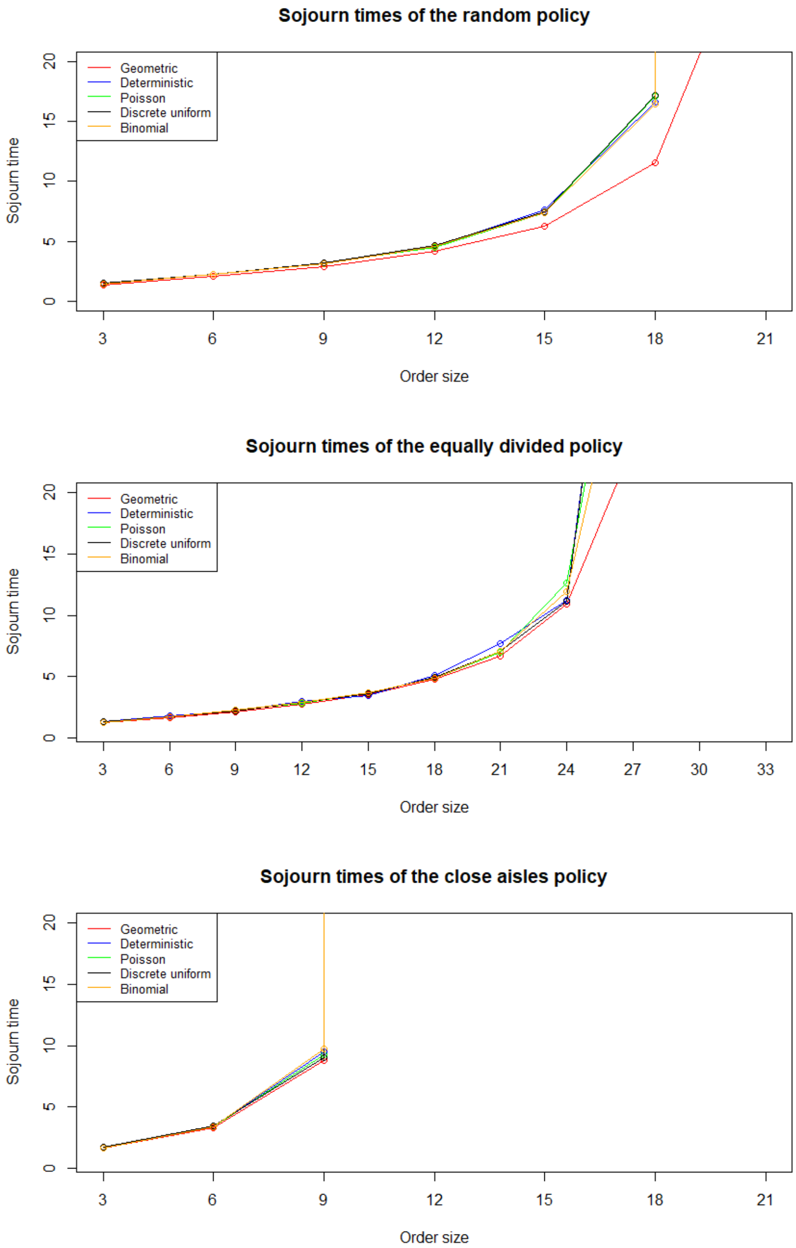


Figure 3.4: Mean sojourn times for different policies.

For the sojourn times, the graph of the *1 aisle* policy is not shown since this will be discussed separately.
For all the different policies it becomes clear that geometrically distributed

order sizes give the smallest service times. This is in line with the results that were found for the basic model with only 1 aisle. So it seems that the amount of aisles and the policy does not influence which distribution for the order sizes is the best. Next, the different policies will be discussed. For this the order sizes are again geometrically distributed just like in the basic model. The following graphs are obtained:

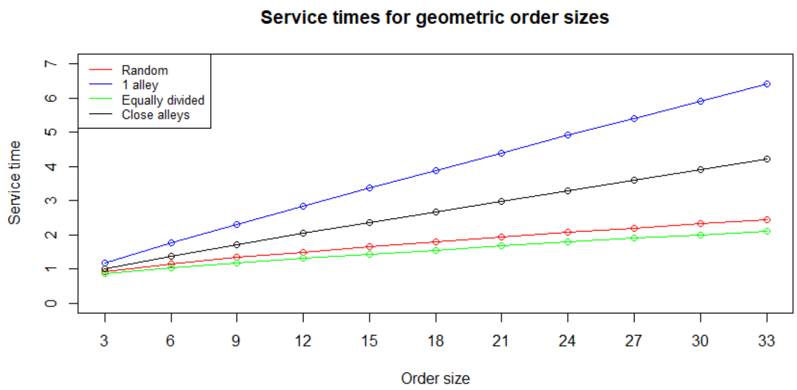


Figure 3.5: Mean service times for different policies and order sizes.

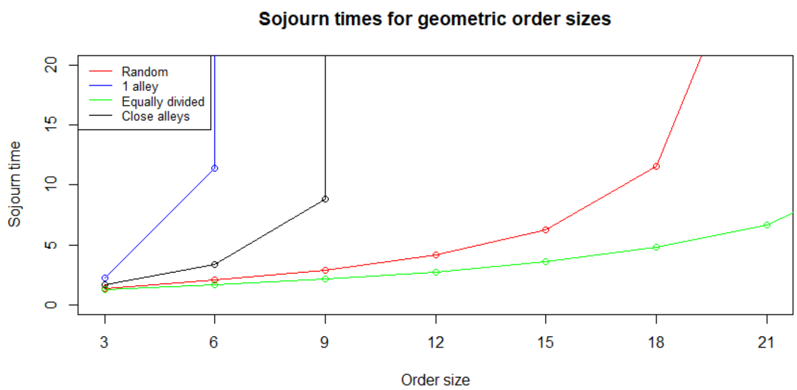


Figure 3.6: Mean sojourn times for different policies and order sizes.

In Figures 3.5 and 3.6 you can see that the 1 aisle policy is the worst policy in terms of both the mean service time and mean sojourn time of the orders. This is also what you expect because for the 1 aisle policy the service time is the highest since every item is in 1 aisle. Since the next order can only be started when the current order is completely finished, the sojourn time is then also the highest. In general it can be stated that policies with a lower service time than another policy, will also have a lower sojourn time. Another observation is that the lines for the mean service time curve for lower order sizes and for higher order sizes it is linear. This can be explained when we look at the expression for the mean service time:

$$\mathbf{E}[B] = \mathbf{E}[max(U_1, U_2, ..., U_N)] + \mathbf{E}[V]\,\mathbf{E}[N]$$

the first expression is the maximum of all the *walk times* which is not linear. However, it is always smaller than 1. The second expression is just linear. So for increasing values of the order size, the first expression will get close to 1 but will grow very slowly while the second expression grows linearly. Therefore you see for lower order sizes that the line curves while for higher order sizes it does not.
So for this model it is best to have the items equally divided over all the aisle so that the service time is as low as possible.

1 aisle **policy**

Earlier on the claim that for the *1 aisle* policy the mean service time will be the same as in the basic model was made. The results of the simulation of the *1 aisle* policy will now be compared with the analytic results and the results of the simulation of the basic model.
For the service time the results are as follows:

Mean order size	E[B] basic analytic	E[B] basic simulation	E[B] extended simulation
1	0.667	0.666	0.667
2	0.947	0.947	0.947
3	1.176	1.176	1.176
4	1.384	1.384	1.384
5	1.580	1.580	1.581
6	1.770	1.770	1.769
7	1.955	1.956	1.955

The 95% confidence intervals for the expected service time of the extended model are:

Mean order size	E[B] extended simulation	95% Confidence interval
1	0.667	(0.658, 0.675)
2	0.947	(0.935, 0.959)
3	1.176	(1.160, 1.192)
4	1.384	(1.364, 1.404)
5	1.581	(1.556, 1.606)
6	1.769	(1.740, 1.799)
7	1.955	(1.920, 1.990)

The results of both models do come along quite nicely. The claim that for the *1 aisle* policy the mean service will be the same as in the basic model is according to these results correct.

3.2.2 Conclusion

When you take a look at the graphs of the mean service time in Figure 3.3, it becomes clear that once again geometrically distributed order sizes give the lowest outcome. This is in line with the results from the basic model. Furthermore, the same observation was done for the mean sojourn time in Figure 3.4. When comparing the different policies with one another for geometrically distributed order sizes, it becomes clear that the *1 aisle* policy is performing really bad in terms of mean service and mean sojourn time. The more equally you divide the jobs over the aisles, the better results you get.
Lastly, the connection between the *1 aisle* policy and the basic model was made. According to the simulations the service times of both models are equal.

3.3 Extended model 2

In this part the second extended model will be analysed. Also here, the mean service and mean sojourn time will be analysed for different policies. The biggest difference with extended model 1 is that the employees of the different aisles do not have to wait on each other but they can just work on the queue of their own aisle until this queue is empty. First, the system with a very low arrival rate for the orders will briefly be discussed and after that the focus will be on the system with higher arrival rates.

3.3.1 Low arrival rate

What is meant by a very low arrival rate is actually that the system is almost always empty when an order arrives in the system. So there will be no waiting time for this order. What you would expect is that the sojourn time for the orders in this extended model will be more or less the same as the service time in extended model 1.
But we actually want to know what happens when there is a higher arrival rate and orders have to wait before they are executed.

3.3.2 Higher arrival rates

In this part the arrival rate will not be very low. So the waiting time will not be close to zero anymore and it changes the values for the sojourn time. Like discussed in the first extended model, it is expected that the 1 aisle policy will once again be the worst choice in terms of sojourn times and we get a similar kind of graph as in Figure 3.6. A mathematical approach to the sojourn time will be very difficult for the same reasons discussed for extended model 1, so this model is mostly discussed with the help of simulations. However, for this model we can say that the sojourn time is given by:

$$S = max(S_1, S_2, S_3, ..., S_n)$$

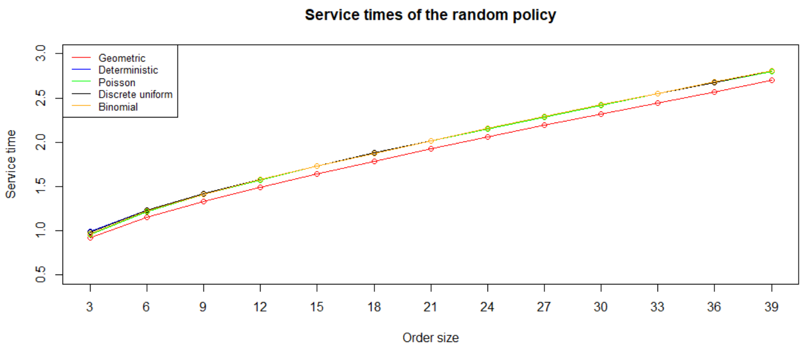
That is because all the aisles have a different queue and therefore this sojourn time is the maximum of sojourn time of these individual queues. Just like in the first extended model this will not give very much insight but once again you can think that for the *1 aisle* policy, the results will be similar to those from the basic model. The only thing that you have to take into account is that the arrival rate should be proportional to the number of aisles in the system.
In the next part simulation is used to check if this idea is correct.

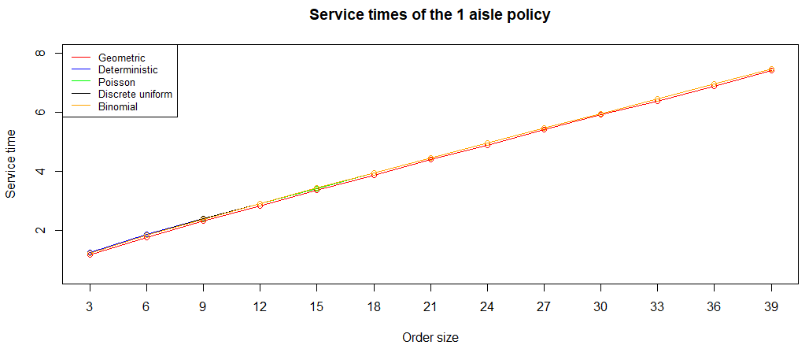
3.3.3 Simulation

The code that is used together with a brief explanation can be found in Appendix C. Now the results will be discussed.

Results

For the simulations the mean order size will vary. The arrival rate for these simulations is once again equal to $\frac{1}{2}$. The following results are obtained:





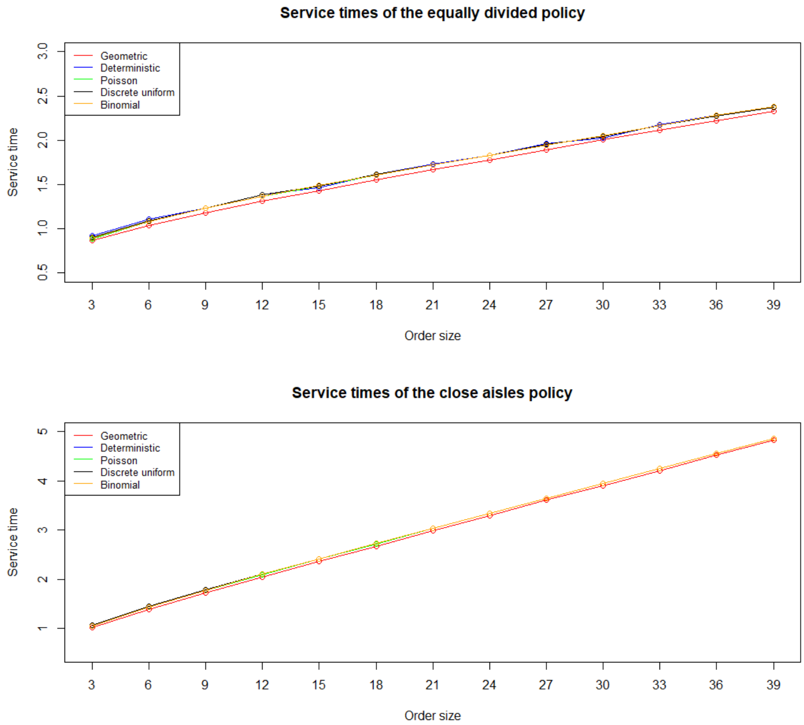
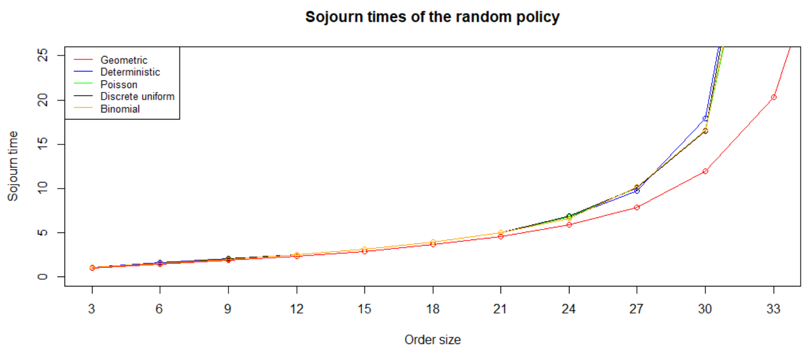


Figure 3.8: Mean service times for different policies.

From these graphs we see that once again the geometric distribution turns out to be the best distribution for the order sizes in terms of mean service time.
For the sojourn times:



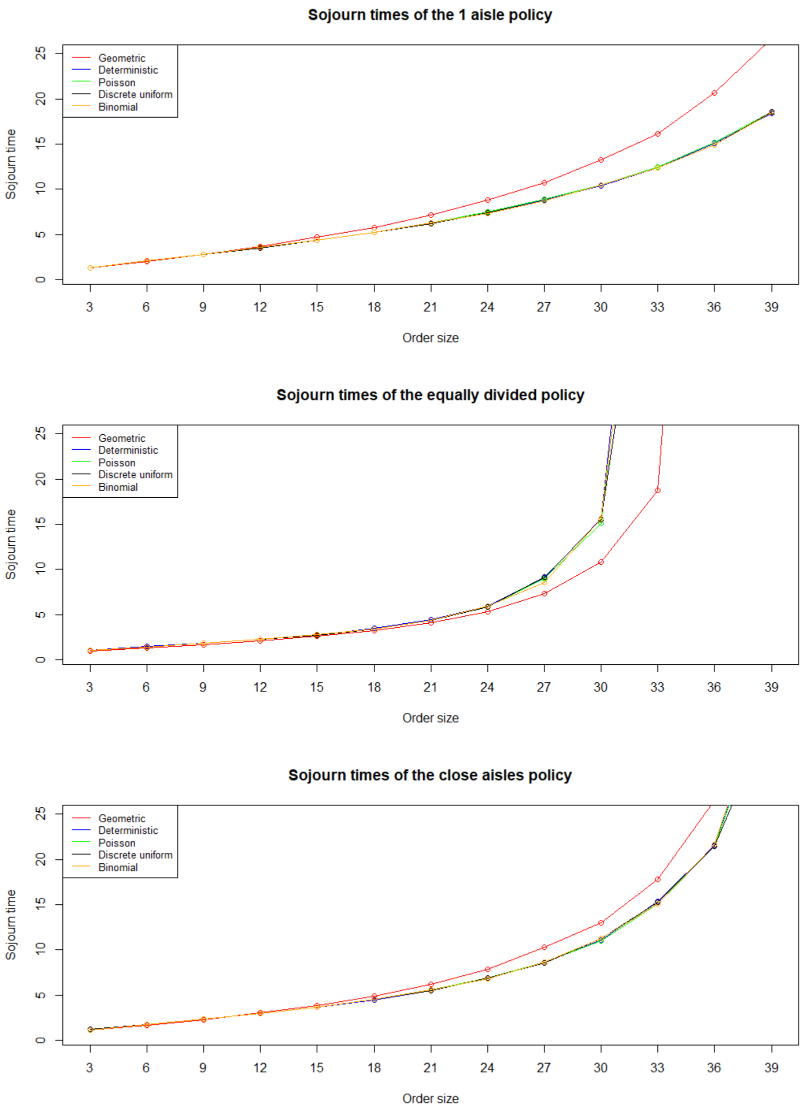


Figure 3.10: Mean sojourn times for different policies.

Also for the different policies and geometrically distributed order sizes, the results are as follows:

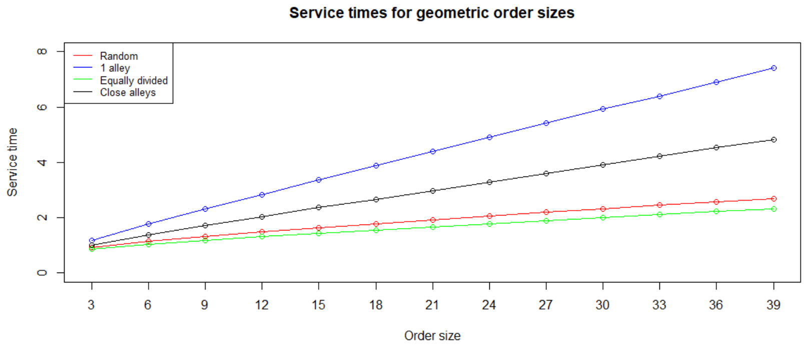


Figure 3.11: Mean service times for different policies and order sizes.

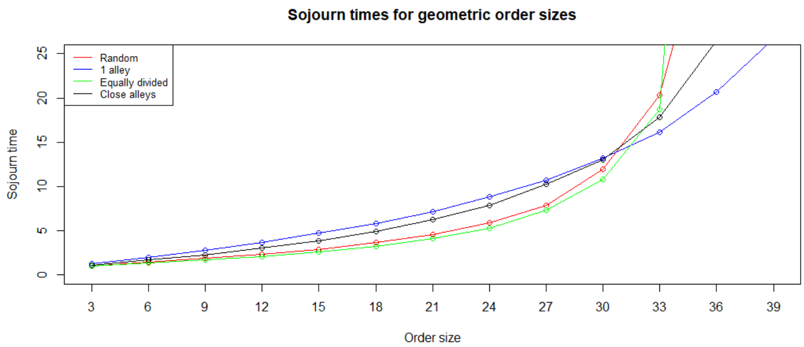


Figure 3.12: Mean sojourn times for different policies and order sizes.

For the mean service times, the results are exactly like you would expect. The geometrically distributed order sizes give the best results like it has done this entire report. But for the sojourn time, there are some interesting things that happened.

- As the mean order sizes get higher, the *1 aisle* policy is becoming better in comparison with the other policies and even becomes the best for really higher values of the order sizes. Furthermore the *equally divided* policy becomes the worst for higher values. So the expectation that this figure would look a lot like that of extended model 1 is not right. Is this outcome logical? If you think of a very busy system, the employees of the different aisles have more and more work to do. When it gets 'too' busy, it is not likely that all the aisles get unstable but probably only 1 aisle will have too much work. In this case the policy that puts all the orders in 1 aisle has a chance to completely avoid this busy aisle and the downside that 1 employee has to pick all the items of that order outweighs the downside of have items lined up in a long queue. You see that the *close aisles* is also doing quite good for higher mean order sizes because it can also avoid the busy aisle.

- Another interesting result is that for the *1-aisle* policy and *close aisles* policy the geometric distribution give the worst results in terms of mean sojourn time while for the other policies it is still doing better than the other distributions. It can not be explained why this is actually the case.

1 aisle policy

Earlier on the claim that for the *1 aisle* policy the mean sojourn time will be the same as in the basic model was made. However, you have to take into account that the arrivals of orders should be increased proportional to the amount of aisles. The results of the simulation of the *1 aisle* policy will now be compared with the analytic results and the results of the simulation of the basic model. The arrival rate is $\frac{5}{2}$ in order to compare it to the results of the basic model where an arrival rate of $\frac{1}{2}$ was used.
For the sojourn time the results are as follows:

Mean order size	E[S] basic analytic	E[S] basic simulation	E[S] extended simulation
1	0.865	0.865	0.865
2	1.463	1.466	1.462
3	2.223	2.221	2.222
4	3.379	3.378	3.373
5	5.505	5.464	5.455
6	10.780	10.844	10.605
7	60.605	48.205	30.701

The 95% confidence intervals for the expected sojourn time of the extended model are:

Mean order size	E[S] extended simulation	95% Confidence interval
1	0.865	(0.842, 0.888)
2	1.462	(1.400, 1.524)
3	2.222	(2.082, 2.361)
4	3.373	(3.021, 3.725)
5	5.455	(4.534, 6.376)
6	10.605	(7.104, 14.106)
7	30.701	(14.758, 46.643)

For lower mean order sizes, it results come along nicely. But for higher values of the mean order size, the expected sojourn time seem to differ more and more. So the claim that the sojourn time of the basic model is the same as that of extended model 2 with *1 aisle* policy, can not be said with complete certainty.

3.3.4 Conclusion

For extended model 2, the service time was once again the best for all policies when the order sizes are geometric distributed. This result is in line with the results from the basic model and extended model 1. But the results for the mean sojourn time gave more interesting results. It turns out that the geometric distribution is not the best for all policies when it comes to the mean sojourn time. A clear explanation for this is difficult to give. Furthermore, the *1 aisle* policy is giving the best results in terms of mean sojourn time when the stability condition is getting closer to 1. It turns out that it is better for an order when it completely avoid a busy aisle and have a higher service time than having a lower service time but some jobs have to wait longer in the queue.
Lastly the connection with the basic model was made for the *1 aisle* policy. The claim was that the sojourn time should be the same when the arrival rate is proportional to the amount of aisles. But this does not seem to be completely true. When the system becomes more unstable, the values for the mean sojourn time differ more and more.

Chapter 4

Conclusion

The aim of this report was to see how the order size influences the model. There were 3 different models: the basic model, extended model 1 and extended model 2. For each of them, the mean service time and mean sojourn time were analysed. For the basic model an analytic approach and simulations were used in order to get the results while for the extended models only the simulations were used. That is because an analytic approach turned out the be way more difficult when multiple aisles were in play.

For the basic model the results from the simulations came along nicely with the results that we found by an analytical approach. After that, different distributions were compared with each other and it turned out to be that the geometric distribution was the best in terms of mean service and mean sojourn time. At least, when the basic and extended model 1 were used. For extended model 2 this was not always the case. It heavily depended on the policy that was used to say anything about the best distribution for order size when you compare the mean sojourn time. An explanation for that was not found.

Furthermore it was also interesting to see which policy performs the best. For extended model 1 it was clear that the order should be distributed evenly over the aisles in order to get the lowest mean service time and mean sojourn time. For extended model 2 this was not the case. When the system becomes busier, it is better to put all the orders in 1 aisle in the off chance to avoid very busy aisles.

Lastly the connection between the extended models and the basic model was made. The idea was that the service time of extended model 1 with the *1 aisle* policy should be the same as that of the basic model. This turned

out to be correct according to the results of the simulations.
Another idea was that the sojourn time of extended model 2 with the *1 aisle* policy should also be the same as the mean sojourn time of the basic model, as long as the arrival rate is proportional to the number of aisles in extended model 2. For lower order sizes this seems to be true but when the system becomes more and more unstable, the results differ pretty significantly. So it could not be said with complete certainty that this claim was correct.

Chapter 5

Further research

Like mentioned before, the results from extended model 2 were not in line with the expectations and ideas that were made beforehand. A clear and understandable reason for that was not found in this report. For further research this would be an interesting topic. Are the results that we found here really that different from the results we found in the basic model? And if so, what are the reasons for that?
Another topic for further research is to dive further in the characteristics of the different policies that were used in this report. Our focus was mainly on influence of the distribution of the order sizes on the system but you could also put the focus on the policies that were used. An interesting result for extended model 2 was found which told us that the performance of the policy depends on how busy the system is. For further research it can be investigated if there are other performances measures of the system which influence the performance of a policy.

Bibliography

[1] Ivo Adan, Céline Comte, Jacques Resing, and Rik Timmerman. *Queueing Systems*. Lecture notes, Eindhoven University of Technology, 2021.

[2] Giorgia Casella, Andrea Volpi, Roberto Montanari, Letizia Tebaldi, and Eleonora Bottani. "Trends in order picking: a 2007–2022 review of the literature". In: *http://mc.manuscriptcentral.com/tpmr* 11.1 (Dec. 2023). ISSN: 2169-3277. DOI: 10.1080/21693277.2023.2191115. URL: https://www.tandfonline.com/doi/abs/10.1080/21693277.2023.2191115.

[3] Jie Hao and Anant Godbole. "Distribution of the maximum and minimum of a random number of bounded random variables". In: *Open Journal of Statistics* 06.02 (2016), pp. 274–285. ISSN: 2161-718X. DOI: 10.4236/OJS.2016.62023.

[4] René de Koster, Tho Le-Duc, and Kees Jan Roodbergen. "Design and control of warehouse order picking: A literature review". In: *European Journal of Operational Research* 182.2 (Oct. 2007), pp. 481–501. ISSN: 0377-2217. DOI: 10.1016/J.EJOR.2006.07.009.

[5] Moshe Shaked and J. George. Shanthikumar. *Stochastic Orders*. Springer, 2007, p. 473. ISBN: 0387346759. URL: https://books.google.com/books/about/Stochastic_Orders.html?hl=nl&id=rPiToBK2rwwC.

Appendix A

Basic model

```
SimBasicModel<-function(N, ordersizes, dist){
  interarrivals<-rexp(N, lambda)
  arrivals<-cumsum(interarrivals)
  walktimes = matrix(numeric(), nrow = N, ncol = max(ordersizes))
  picktimes = matrix(numeric(), nrow = N, ncol = max(ordersizes))
  bedieningstijd<-rep(0,N)
  maxen<-rep(0,N)
  packers<-rep(0,N)
  wachttijd<-rep(0,N)
  verblijftijd<-rep(0,N)
  b2<-rep(0,N)
  for (i in 1:N) {
    for (j in 1:ordersizes[i]) {
      walktimes[i,j]<-runif(1,0,1)
      picktimes[i,j]<-rnorm(1,mu,sigma)
    }
    bedieningstijd[i]<-max(na.omit(walktimes[i,]))
        +sum(na.omit(picktimes[i,]))
    maxen[i]<-max(na.omit(walktimes[i,]))
    packers[i]<-sum(na.omit(picktimes[i,]))
    b2[i]<-bedieningstijd[i]*bedieningstijd[i]
    if(i>1){
      wachttijd[i]<-max(wachttijd[i-1]+bedieningstijd[i-1]
        -interarrivals[i],0)
    }
    verblijftijd[i]<-bedieningstijd[i]+wachttijd[i]
  }
  return(list(dist, mean(bedieningstijd),
```

```
    mean(verblijftijd)))
}
```

Explanation

- The input of the function SimBasicModel is the amount of orders N, the size of each of those orders, and the name of the distribution that is used. So beforehand you already make the orders by some distribution. The next thing that happens is generating the arrivals which is a Poisson process with arrival rate $\frac{1}{2}$.
- Then a matrix for both the walk times and pick times is initialised where the amount of rows is equal to N and the amount of columns is equal to the maximum of the order sizes.
- Next, the service and sojourn times are initialised, along with some other quantities. After that, for each order the walk times and pick times are simulated for every job in the order. With this, the service time is calculated for the order. Eventually the sojourn time can be calculated with the waiting time and the service time of the order.
- Lastly the function returns a few quantities, for example the mean service time and mean sojourn time.

Appendix B

Extended model 1

```
  SimExtModel1<-function(N, ordersizes, dist){
interarrivals<-rexp(N, lambda)
arrivals<-cumsum(interarrivals)
bedieningstijd<-rep(0,N)
wachttijd<-rep(0,N)
verblijftijd<-rep(0,N)
for (i in 1:N) {
  gangperitem<-sample.int(NrRows,ordersizes[i],
      replace = TRUE, useHash = FALSE)
  walktimespergang = matrix(numeric(), nrow = NrRows,
      ncol = ordersizes[i])
  picktimespergang = matrix(numeric(), nrow = NrRows,
      ncol = ordersizes[i])
  bedieningstijdpergang <- rep(0,NrRows)
  for (j in 1:NrRows) {
    for (k in 1:sum(gangperitem == j)) {
      if (sum(gangperitem == j)>0) {
        walktimespergang[j,k]<-runif(1,0,1)
        picktimespergang[j,k]<-rnorm(1,mu,sigma)
      }
    }
    bedieningstijdpergang[j]<-
      max(c(na.omit(walktimespergang[j,]),0))+
      sum(na.omit(picktimespergang[j,]))
  }
  bedieningstijd[i]<-max(bedieningstijdpergang)
  b2[i]<-bedieningstijd[i]*bedieningstijd[i]
  if(i>1){
```

```
      wachttijd[i]<-max(wachttijd[i-1]+
        bedieningstijd[i-1]-interarrivals[i],0)
    }
    verblijftijd[i]<-bedieningstijd[i]+wachttijd[i]
  }
  return(list(dist, mean(bedieningstijd), mean(b2),
    var(bedieningstijd), mean(verblijftijd), mean(ordersizes),
    mean(ordersizes*ordersizes/(ordersizes+1))/6))
}

SimExtModel11Row<-function(N, ordersizes, dist){
  interarrivals<-rexp(N, lambda)
  arrivals<-cumsum(interarrivals)
  bedieningstijd<-rep(0,N)
  wachttijd<-rep(0,N)
  verblijftijd<-rep(0,N)
  for (i in 1:N) {
    gangperitem<-rep(sample.int(NrRows,1,
        replace = TRUE,useHash = FALSE),ordersizes[i])
    walktimespergang = matrix(numeric(),
        nrow = NrRows, ncol = ordersizes[i])
    picktimespergang = matrix(numeric(),
        nrow = NrRows, ncol = ordersizes[i])
    bedieningstijdpergang <- rep(0,NrRows)
    for (j in 1:NrRows) {
      for (k in 1:sum(gangperitem == j)) {
        if (sum(gangperitem == j)>0) {
          walktimespergang[j,k]<-runif(1,0,1)
          picktimespergang[j,k]<-rnorm(1,mu,sigma)
        }
      }
      bedieningstijdpergang[j]<-
        max(c(na.omit(walktimespergang[j,]),0))+
        sum(na.omit(picktimespergang[j,]))
    }
    bedieningstijd[i]<-max(bedieningstijdpergang)
    b2[i]<-bedieningstijd[i]*bedieningstijd[i]
    if(i>1){
      wachttijd[i]<-max(wachttijd[i-1]+
        bedieningstijd[i-1]-interarrivals[i],0)
    }
```

```
    verblijftijd[i]<-bedieningstijd[i]+wachttijd[i]
  }
  return(list(dist, mean(bedieningstijd), mean(b2),
    var(bedieningstijd), mean(verblijftijd), mean(ordersizes),
    mean(ordersizes*ordersizes/(ordersizes+1))/6))
}

SimExtModel1AllRow<-function(N, ordersizes, dist){
  interarrivals<-rexp(N, lambda)
  arrivals<-cumsum(interarrivals)
  bedieningstijd<-rep(0,N)
  wachttijd<-rep(0,N)
  verblijftijd<-rep(0,N)
  for (i in 1:N) {
    gangperitem<-rep(sample.int(NrRows,1,
        replace = TRUE,useHash = FALSE),ordersizes[i])
    for (j in 1:length(gangperitem)) {
      gangperitem[j]<-((gangperitem[j]+j-1)%%NrRows)+1
    }
    walktimespergang = matrix(numeric(),
        nrow = NrRows, ncol = ordersizes[i])
    picktimespergang = matrix(numeric(),
        nrow = NrRows, ncol = ordersizes[i])
    bedieningstijdpergang <- rep(0,NrRows)
    for (j in 1:NrRows) {
      for (k in 1:sum(gangperitem == j)) {
        if (sum(gangperitem == j)>0) {
          walktimespergang[j,k]<-runif(1,0,1)
          picktimespergang[j,k]<-rnorm(1,mu,sigma)
        }
      }
      bedieningstijdpergang[j]<-
        max(c(na.omit(walktimespergang[j,]),0))+
        sum(na.omit(picktimespergang[j,]))
    }
    bedieningstijd[i]<-max(bedieningstijdpergang)
    b2[i]<-bedieningstijd[i]*bedieningstijd[i]
    if(i>1){
      wachttijd[i]<-max(wachttijd[i-1]+
        bedieningstijd[i-1]-interarrivals[i],0)
    }
```

```
    verblijftijd[i]<-bedieningstijd[i]+wachttijd[i]
  }
  return(list(dist, mean(bedieningstijd), mean(b2),
    var(bedieningstijd), mean(verblijftijd), mean(ordersizes),
    mean(ordersizes*ordersizes/(ordersizes+1))/6))
}

SimExtModel1CloseRow<-function(N, ordersizes, dist){
  interarrivals<-rexp(N, lambda)
  arrivals<-cumsum(interarrivals)
  bedieningstijd<-rep(0,N)
  wachttijd<-rep(0,N)
  verblijftijd<-rep(0,N)
  for (i in 1:N) {
    peiler<-sample.int(NrRows,1,
        replace=TRUE,useHash = FALSE)
    myProb<-rep(0,NrRows)
    if (peiler==1){
      myProb[1]<-0.6
      myProb[2]<-0.2
      myProb[NrRows]<-0.2
    }
    if (peiler==NrRows){
      myProb[NrRows]<-0.6
      myProb[1]<-0.2
      myProb[(NrRows-1)]<-0.2
    }
    if (peiler!=1 & peiler!=NrRows){
      myProb[(peiler-1)]<-0.2
      myProb[(peiler)]<-0.6
      myProb[(peiler+1)]<-0.2
    }
    gangperitem<-sample.int(NrRows, ordersizes[i],
        replace = TRUE,prob = myProb,useHash = FALSE)
    walktimespergang = matrix(numeric(),
        nrow = NrRows, ncol = ordersizes[i])
    picktimespergang = matrix(numeric(),
        nrow = NrRows, ncol = ordersizes[i])
    bedieningstijdpergang <- rep(0,NrRows)
    for (j in 1:NrRows) {
      for (k in 1:sum(gangperitem == j)) {
```

```
        if (sum(gangperitem == j)>0) {
          walktimespergang[j,k]<-runif(1,0,1)
          picktimespergang[j,k]<-rnorm(1,mu,sigma)
        }
      }
      bedieningstijdpergang[j]<-
        max(c(na.omit(walktimespergang[j,]),0))+
        sum(na.omit(picktimespergang[j,]))
    }
    bedieningstijd[i]<-max(bedieningstijdpergang)
    b2[i]<-bedieningstijd[i]*bedieningstijd[i]
    if(i>1){
      wachttijd[i]<-max(wachttijd[i-1]+
        bedieningstijd[i-1]-interarrivals[i],0)
    }
    verblijftijd[i]<-bedieningstijd[i]+wachttijd[i]
  }
  return(list(dist, mean(bedieningstijd), mean(b2),
    var(bedieningstijd), mean(verblijftijd), mean(ordersizes),
    mean(ordersizes*ordersizes/(ordersizes+1))/6))
}
```

Explanation

First of all, there is a different function for every different policy. The function is very similar to the basic model but there are some differences. For example the walk and pick times are now generated for every single order and not for the entire model at once. That is done because the walk and pick times is still a matrix but now for the amount of aisles and the amount of jobs of the order. Then the jobs need to be allocated to an aisle. This is done differently for every policy.

- *random* policy: for every job, a random number between 1 and the number of aisles is generated.
- *1 aisle* policy: for every order, a random number between 1 and the number of aisles is generated and all jobs of that order are allocated to that aisle.
- *Equally divided* policy: for every order, a random number between 1 and the number of aisles is generated and then the first job is allocated in that aisle, the second job goes to the next aisle etc.

- *Close aisles* policy: for every order, a random number between 1 and the number of aisles is generated. Then for each job it has a 60% chance of going to that aisle and a 20% chance of be allocated to an aisle that is next to aisle that is generated.

Eventually the service time is now calculated by taking the maximum of the service time of every single aisle.

Appendix C

Extended model 2

```
SimExtModel2<-function(N, ordersizes, dist){
  interarrivals<-rexp(N, lambda)
  arrivals<-cumsum(interarrivals)
  bedieningstijd<-rep(0,N)
  wachttijd<-rep(0,N)
  verblijftijd<-rep(0,N)
  wachttijdpergang = matrix(numeric(),
    nrow = N, ncol = NrRows)
  bedieningstijdpergang = matrix(numeric(),
    nrow = N, ncol = NrRows)
  for (j in 1:NrRows) {
    wachttijdpergang[1,j]<-0
  }
  for (i in 1:N) {
    gangperitem<-sample.int(NrRows,ordersizes[i],
        replace = TRUE,useHash = FALSE)
    walktimespergang = matrix(numeric(),
        nrow = NrRows, ncol = ordersizes[i])
    picktimespergang = matrix(numeric(),
        nrow = NrRows, ncol = ordersizes[i])
    for (j in 1:NrRows) {
      for (k in 1:sum(gangperitem == j)) {
        if (sum(gangperitem == j)>0) {
          walktimespergang[j,k]<-runif(1,0,1)
          picktimespergang[j,k]<-rnorm(1,mu,sigma)
        }
      }
      bedieningstijdpergang[i,j]<-
```

```
        max(c(na.omit(walktimespergang[j,]),0))+
        sum(na.omit(picktimespergang[j,]))
      if (i>1){
        wachttijdpergang[i,j]<-max(wachttijdpergang[i-1,j]+
            bedieningstijdpergang[i-1,j]-interarrivals[i],0)
      }
    }
    wachttijdpergangextra<-wachttijdpergang[i,]
    for (j in 1:NrRows) {
      if (sum(gangperitem == j) == 0){
        wachttijdpergangextra[j]<-0
      }
    }
    bedieningstijd[i]<-max(bedieningstijdpergang[i,])
    b2[i]<-bedieningstijd[i]*bedieningstijd[i]
    if(i>1){
      wachttijd[i]<-max(wachttijdpergangextra)
    }
    verblijftijd[i]<-max(bedieningstijdpergang[i,]+
        wachttijdpergangextra)
  }
  return(list(dist, mean(ordersizes),
    mean(bedieningstijd),mean(verblijftijd)))
}

SimExtModel21Row<-function(N, ordersizes, dist){
  interarrivals<-rexp(N, lambda)
  arrivals<-cumsum(interarrivals)
  bedieningstijd<-rep(0,N)
  wachttijd<-rep(0,N)
  verblijftijd<-rep(0,N)
  wachttijdpergang = matrix(numeric(),
    nrow = N, ncol = NrRows)
  bedieningstijdpergang = matrix(numeric(),
    nrow = N, ncol = NrRows)
  for (j in 1:NrRows) {
    wachttijdpergang[1,j]<-0
  }
  for (i in 1:N) {
    gangperitem<-rep(sample.int(NrRows,1,
        replace = TRUE,useHash = FALSE),ordersizes[i])
```

```
    walktimespergang = matrix(numeric(),
        nrow = NrRows, ncol = ordersizes[i])
    picktimespergang = matrix(numeric(),
        nrow = NrRows, ncol = ordersizes[i])
    for (j in 1:NrRows) {
      for (k in 1:sum(gangperitem == j)) {
        if (sum(gangperitem == j)>0) {
          walktimespergang[j,k]<-runif(1,0,1)
          picktimespergang[j,k]<-rnorm(1,mu,sigma)
        }
      }
      bedieningstijdpergang[i,j]<-
        max(c(na.omit(walktimespergang[j,]),0))+
        sum(na.omit(picktimespergang[j,]))
      if (i>1){
        wachttijdpergang[i,j]<-max(wachttijdpergang[i-1,j]+
            bedieningstijdpergang[i-1,j]-interarrivals[i],0)
      }
    }
    wachttijdpergangextra<-wachttijdpergang[i,]
    for (j in 1:NrRows) {
      if (sum(gangperitem == j) == 0){
        wachttijdpergangextra[j]<-0
      }
    }
    bedieningstijd[i]<-max(bedieningstijdpergang[i,])
    b2[i]<-bedieningstijd[i]*bedieningstijd[i]
    if(i>1){
      wachttijd[i]<-max(wachttijdpergangextra)
    }
    verblijftijd[i]<-max(bedieningstijdpergang[i,]+
        wachttijdpergangextra)
  }
  return(list(dist, mean(ordersizes),
    mean(bedieningstijd), mean(verblijftijd)))
}

SimExtModel2AllRow<-function(N, ordersizes, dist){
  interarrivals<-rexp(N, lambda)
  arrivals<-cumsum(interarrivals)
  bedieningstijd<-rep(0,N)
```

```
wachttijd<-rep(0,N)
verblijftijd<-rep(0,N)
wachttijdpergang = matrix(numeric(),
  nrow = N, ncol = NrRows)
bedieningstijdpergang = matrix(numeric(),
  nrow = N, ncol = NrRows)
for (j in 1:NrRows) {
  wachttijdpergang[1,j]<-0
}
for (i in 1:N) {
  gangperitem<-rep(sample.int(NrRows,1,
      replace = TRUE,useHash = FALSE),ordersizes[i])
  for (j in 1:length(gangperitem)) {
    gangperitem[j]<-((gangperitem[j]+j-1)%%NrRows)+1
  }
  walktimespergang = matrix(numeric(),
      nrow = NrRows, ncol = ordersizes[i])
  picktimespergang = matrix(numeric(),
      nrow = NrRows, ncol = ordersizes[i])
  for (j in 1:NrRows) {
    for (k in 1:sum(gangperitem == j)) {
      if (sum(gangperitem == j)>0) {
        walktimespergang[j,k]<-runif(1,0,1)
        picktimespergang[j,k]<-rnorm(1,mu,sigma)
      }
    }
    bedieningstijdpergang[i,j]<-
      max(c(na.omit(walktimespergang[j,]),0))+
      sum(na.omit(picktimespergang[j,]))
    if (i>1){
      wachttijdpergang[i,j]<-max(wachttijdpergang[i-1,j]+
          bedieningstijdpergang[i-1,j]-interarrivals[i],0)
    }
  }
  wachttijdpergangextra<-wachttijdpergang[i,]
  for (j in 1:NrRows) {
    if (sum(gangperitem == j) == 0){
      wachttijdpergangextra[j]<-0
    }
  }
  bedieningstijd[i]<-max(bedieningstijdpergang[i,])
```

```
    b2[i]<-bedieningstijd[i]*bedieningstijd[i]
    if(i>1){
      wachttijd[i]<-max(wachttijdpergangextra)
    }
    verblijftijd[i]<-max(bedieningstijdpergang[i,]+
        wachttijdpergangextra)
  }
  return(list(dist, mean(ordersizes),
    mean(bedieningstijd), mean(verblijftijd)))
}

SimExtModel2CloseRow<-function(N, ordersizes, dist){
  interarrivals<-rexp(N, lambda)
  arrivals<-cumsum(interarrivals)
  bedieningstijd<-rep(0,N)
  wachttijd<-rep(0,N)
  verblijftijd<-rep(0,N)
  wachttijdpergang = matrix(numeric(),
    nrow = N, ncol = NrRows)
  bedieningstijdpergang = matrix(numeric(),
    nrow = N, ncol = NrRows)
  for (j in 1:NrRows) {
    wachttijdpergang[1,j]<-0
  }
  for (i in 1:N) {
    peiler<-sample.int(NrRows,1,
        replace=TRUE,useHash = FALSE)
    peiler
    myProb<-rep(0,NrRows) #De gangen vormen een "cirkel"
    if (peiler==1){
      myProb[1]<-0.6
      myProb[2]<-0.2
      myProb[NrRows]<-0.2
    }
    if (peiler==NrRows){
      myProb[NrRows]<-0.6
      myProb[1]<-0.2
      myProb[(NrRows-1)]<-0.2
    }
    if (peiler!=1 & peiler!=NrRows){
      myProb[(peiler-1)]<-0.2
```

```
      myProb[(peiler)]<-0.6
      myProb[(peiler+1)]<-0.2
    }
    gangperitem<-sample.int(NrRows,ordersizes[i],
        replace = TRUE,prob = myProb,useHash = FALSE)
    walktimespergang = matrix(numeric(),
        nrow = NrRows, ncol = ordersizes[i])
    picktimespergang = matrix(numeric(),
        nrow = NrRows, ncol = ordersizes[i])
    for (j in 1:NrRows) {
      for (k in 1:sum(gangperitem == j)) {
        if (sum(gangperitem == j)>0) {
          walktimespergang[j,k]<-runif(1,0,1)
          picktimespergang[j,k]<-rnorm(1,mu,sigma)
        }
      }
      bedieningstijdpergang[i,j]<-
        max(c(na.omit(walktimespergang[j,]),0))+
        sum(na.omit(picktimespergang[j,]))
      if (i>1){
        wachttijdpergang[i,j]<-max(wachttijdpergang[i-1,j]+
            bedieningstijdpergang[i-1,j]-interarrivals[i],0)
      }
    }
    wachttijdpergangextra<-wachttijdpergang[i,]
    for (j in 1:NrRows) {
      if (sum(gangperitem == j) == 0){
        wachttijdpergangextra[j]<-0
      }
    }
    bedieningstijd[i]<-max(bedieningstijdpergang[i,])
    b2[i]<-bedieningstijd[i]*bedieningstijd[i]
    if(i>1){
      wachttijd[i]<-max(wachttijdpergangextra)
    }
    verblijftijd[i]<-max(bedieningstijdpergang[i,]+
        wachttijdpergangextra)
  }
  return(list(dist, mean(ordersizes),
    mean(bedieningstijd), mean(verblijftijd)))
}
```

Explanation

The functions that are used for extended model 2 are quite similar to those of extended model 1. The only difference is that every aisle has its own waiting time and service time. For that:

- Two matrices are made with the number of rows equal to the amount of orders in the system and the number of columns equal to the number of rows.
- Next the jobs of an order are once again allocated to an aisle in the same manner that is done in extended model 1.
- The service time is once again calculated for every different aisle but the waiting time is only calculated for an aisle if the order did have jobs in that aisle.
- Eventually the sojourn time is calculated by taking the maximum of the sojourn times of all the aisles where the order did have jobs.