Final Project

Master Applied Mathematics

Storage requirement of a dry bulk terminal

Simone van den Heuvel

S1558048

Supervisor
TBA
A.J. Dijkstra MSc.

Supervisor
University of Twente
Dr. J.C.W. van Ommeren

Graduation committee:

Prof.dr. N.M. van Dijk

Dr. J.C.W. van Ommeren

A.J. Dijkstra MSc.

Dr. B. Manthey

August 30, 2019

Abstract

A dry bulk terminal can be seen as the buffer between the product flows on the landside and the waterside. These flows depend on multiple stochastic distributions, such as the arrival times of trucks and vessels and their load sizes. This causes large differences in the storage level of the terminal over time. In this research the required storage level of a dry bulk terminal is investigated. With different queueing models of a terminal, the steady state probability distribution of the storage level is determined. Besides these models, two different simulations of a terminal are made, one in Python, the other in Trafalquar, the simulation tool of TBA. With a case study of an export terminal with five commodities, the results of the simulations and the queueing models are compared to each other to find the required storage capacity.

Contents

Chapter 1

Introduction

The world gets smaller every year. For work we have contacts all over the world, we go on holidays to see total different cultures, we eat food that could never grow in our own country, and a lot of our daily used products are made on the other side of the world. For all those things a lot of transportation is needed, mostly over sea. Massive vessels sail the oceans to spread the products over multiple continents. In the ports they unload their products, where trucks, trains or smaller vessels will transport it to the final destination. Because those different phases of transport do not connect perfectly on each other, some buffer is needed to store the products for a while; a terminal.

There are many different kinds of terminals in the world. They differ in size and in the way of transportation to and from the terminal. Goods can be brought and picked up by trucks, trains, deep sea vessels and barges. Another big difference is the kind of products they can handle. Some products are packed in containers, others will be transported as dry or liquid bulk. This research is focused on terminals that handle dry bulk. The major commodities of those terminals are iron ore, grain, coal, phosphate rock and bauxite ore (UNCTAD, 1978).

Another division of terminals can be made in import and export. Export terminals are located in producing countries and can be used to store the products until it is sold for a good price. Those terminals mostly have only one commodity in stock. Import terminals are situated in countries with demand for raw products and will often handle multiple commodities (Van Vianen, Ottjes, and Lodewijks, 2011).

To get the products in and out the terminal, different equipment is needed. An example of a dry bulk export terminal is schematically shown in figure 1.1. On the landside trucks and trains arrive that get unloaded in their stations. From these stations the product is transported with conveyor belts to the stacker that puts it in the storage yard. When a vessel arrives at the quay the reclaimer

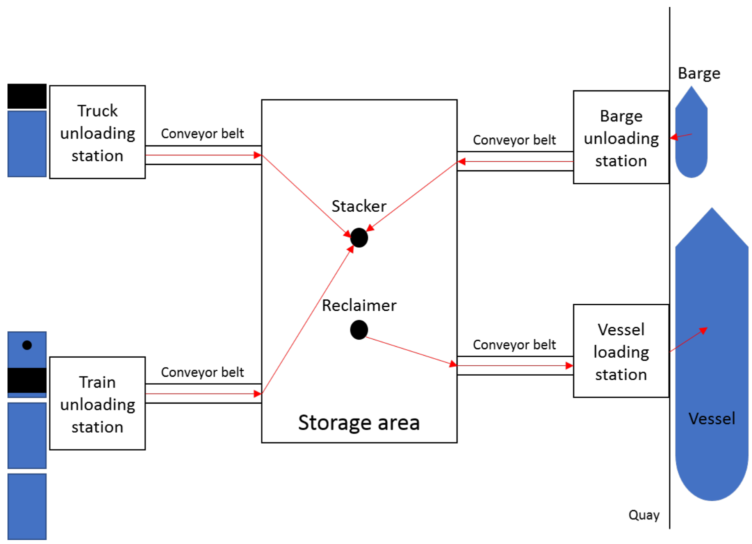

Figure 1.1: Schematic overview of an export terminal

will place the commodity back on the conveyor belts, which transport it to the loading station. Here the product is loaded into the vessel and leaving the terminal. On the waterside it is also possible that barges arrive that deliver goods to the terminal.

For an import terminal the idea is the same, but now the vessels deliver products, while the trucks, trains and barges will pick it up. Some terminals are a combination of import and export, then all transport equipment arriving at the terminal could be loaded and unloaded. It is also possible that the terminal has some bypass of conveyor belts, then the product can be transported from land- to waterside without being stored in the storage area.

An interesting question for terminals is the needed storage capacity. The terminal is a buffer between two transportation flows, but which size does this buffer have to be? This stockyard size depends on the arrival distribution of the ships, the shipload size, the hinterland transport distribution and the loading and unloading rates of the different equipment (UNCTAD, 1978). Because of the stochastic nature of those systems, it is not easy to find the required storage size.

The main research question of this report is:

> What is the required storage capacity of a dry bulk terminal?

Some subquestions are:

- What is the difference in the required storage capacity of a terminal with multiple commodities compared to only one commodity?

- Which characteristics of the terminal have a large effect on the required storage capacity?

This research is conducted at the simulation department of TBA Delft. TBA provides integrated software solutions to simplify the operations of ports, terminals and warehouses all over the world. At the simulation department they model the terminals of the customer to find the optimal design and strategy for their operations. They answer questions like, what types of transport and (un)load equipment is needed, how much equipment is needed and what is the required capacity of the terminal.

This report starts with a literature background of terminals and mathematical models that can be used. Then the problem description is stated with all the assumptions that are made in the models together with a queueing model description. In the next chapter the departure process of the unloading station at a terminal is investigated, followed by the determination of the steady state probability distribution of the queue length at the loading station. Chapter 6 explains models that are combinations of these unloading and loading stations. After all those mathematical models the structure of the simulation in Python and an explanation of the simulation tool of TBA is given. In the end a case study is conducted with the two simulation tools and the effects of changes in the input are investigated. The same input of the case study is used in the different queueing models and those results are compared with the simulation results. At the end the conclusion of the research is given, together with discussion points and ideas for further research.

Chapter 2

Literature

In the current literature there is not yet much published about the required storage capacity of a terminal. In general the amount of literature about container terminals is larger than about dry bulk terminals. To determine the required storage capacity for those terminals different methods are used. For container terminals it is calculated with the mean dwell time of the containers (Mulinas, 2012). For dry bulk terminals the rule of thumb is to have 10 percent of the annual throughput as storage area (Schott and Lodewijks, 2007) or two to four times the largest shipload per commodity (Agerschou, 2004, chapter 8).

Products in the storage area of a terminal are waiting to be picked up and transported further to their final destination. This could be seen as a queue of products, waiting on their service. The modes of transport also could queue before the loading and unloading stations when the machines are busy with other vehicles. Therefore, one way to model a terminal is as a network of queues.

There are already papers published that use queuing theory for (bulk) terminals, but mostly to determine the waiting time of the vessels and the best berth allocation. Mulinas (2012) mentions often used queueing systems for the arriving vessels. One of those systems is applied by El-Naggar (2010) to a container terminal in Egypt. Jagerman and Altiok (2003) also apply a queueing system at the quay, but for bulk terminals. Bugaric et al. (2011) take the ship unloading cranes as queues and Robenek et al. (2014) use a MIP to allocate the vessels to the berths and assign the commodities to the yard.

For the design of the yard there is also research conducted. So do Lodewijks et al. (2007) use a discrete event simulation to find the best locations of the stackers and reclaimers. Binkowski and McCarragher (1999) look for the best ratio of the number and size of stockpiles in a storage area of a certain size. And Van Vianen, Ottjes, and Lodewijks (2014) use a simulation to find the optimal stockyard size.

In other transportation applications queueing is also already used. Schwarz and Epp (2016) looked at the circulating vertical conveyor system which can be seen as a $G^X|G^{0,C}|1|K$ queue and determined the queue length distribution, waiting time and inter-departure times. Alfa (1982) studied the queue length distribution of a discrete time bulk server queue with time-inhomogeneous compound Poisson input, which can be used for bus stops. Medhi (1975) obtains the distribution with its moments of the queue length for a $M|M^{a,b}|1$ queue. Krishnamoorthy and Ushakumari (2000) use forward Kolmogorov equations to determine the queue length probabilities and busy period distribution for a $M|M^{1,d}|1$ queue with single departures. These systems can be used for elevators. Gupta and Sikdar (2004) looked at a $M|G^{a,b}|1$ queue with single vacations. They use partial differential equations to determine the steady state distribution of the queue length through an imbedded Markov chain approach.

The arrival process of products into the terminal can be seen as batch arrivals. Every arriving mode of transport delivers a variable amount of product. Chaudhry and Templeton (1983) give a general overview for bulk queues. They show some common used techniques, such as the imbedded Markov chain technique, the supplementary variable technique, basic renewal theory and Laplace-Stieltjes transforms. Kabak (1970) finds the blocking and delay probabilities for a $M^X|M|1$ queue. Maraghi et al. (2009) show a clear derivation of the probability generating function of the queue length for a $M^X|G|1$ queue with random breakdowns and Bernoulli vacations. Henderson and Taylor (1990) proved that for queueing networks with batch arrivals and services a product form equilibrium distribution exists. Suhasini et al. (2013) looked at a system with two parallel queues with bulk arrivals and a single queue in series. With differential equations the generating function of the joint probability of the three queue lengths is determined.

When the unloading stations are defined as queues, the arrival process to the storage area is actually the departure process of those unloading stations. Whitt (1984) gives some methods to approximate the departure process from a single-server queue. Hu (1996) determines the MacLaurin series of the moments and covariances of the departure process of the $G|G|1$ queue. Daley (1976) looks at the output process of a general $G|G|s|N$ queue with finite and infinite N and gives for some arrival and service distributions an equation for the distribution of the departure process. More interesting for the bulk arrival queue is the work of Kempa (2008), who determines the generating function of the Laplace transform of the departure process of a $G^X|G|1$ queue. Stanford and Fischer (1989) describe the Laplace-Stieltjes transform of the interdeparture time distribution function of a $M|G|1$ queue where multiple types of customers arrive. This is interesting when the terminal handles multiple commodities.

The loading station can be modelled as a double-ended queue with arrivals of product and arrivals of transportation equipment. Kim et al. (2010) show a simulation model for a double-ended queue with extensions such as bulk arrivals, batch size service, general distributions and non-zero processing times.

Chapter 3

Problem description

A terminal is a complex system of a lot of different service points and movements. To model this system, first some assumptions are made to explain and simplify the problem.

3.1 Assumptions

A terminal has a lot of different equipment to transport the goods through the terminal. In this research one of the main assumptions is that all transport between the loading and unloading stations, such as conveyor belts, stackers and reclaimers, have enough capacity to handle the flows of product, without being a bottleneck. Another main assumption is that the terminal is either importing or exporting goods, and can not do both. In both cases the unloading and loading stations are strictly separated. So it is not possible to first unload a vessel and then load another one.

In this way the terminal can be modelled as a system with two handling stations; loading and unloading. See figure 3.1 for a schematic overview. For an import terminal the unloading station is located at the waterside and unloads the vessels at the quay, while at the loading station the products are loaded into the trucks and trains on the landside. For an export terminal the unloading station is on the landside and at the quay the vessels are loaded.

At both the stations different types of *vehicles* arrive. With *vehicles* is meant all types of equipment that can deliver or pick up goods. At the landside these are trucks and trains with different lengths and volumes. At the waterside there are different types of vessels that moor at the berth. All those different types of *vehicles* have their own load size distribution. When a load size distribution is continuous, every value taken out of this distribution will be rounded to a

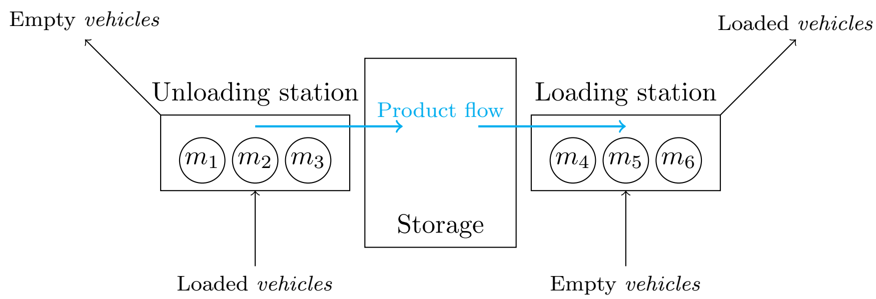

Figure 3.1: Model of a terminal

discrete load size, for example in whole tons. In this way all processes in the terminal can be defined per unit of product.

The two stations at the terminal have multiple machines that can handle the goods. Those machines (un)load the *vehicles* per unit of product. It is assumed that all machines in a station have the same service distribution to handle one unit of product independent of the type of *vehicle* and the commodity they are working on. Every type of *vehicle* has its own maximum number of machines it can be served by. A truck for example can only be handled by one machine, while a big vessel could be handled by five machines.

The terminal can store multiple commodities. Different *vehicles* with different commodities can be in service at the same time. So we assume that the terminal has enough routes from the stations to the storage yard to simultaneously handle commodity A from *vehicle* 1 and commodity B from *vehicle* 2. All *vehicles* arriving at the terminal have exactly one type of commodity loaded.

Vehicles arriving at the unloading station will be served first in, first out (FIFO). When a machine has finished its current *vehicle*, it will check if all other *vehicles* in service already have their maximum number of machines assigned to it. If some *vehicle* has not reached its maximum number of machines yet, the idle machine is assigned to this *vehicle* and starts unloading it. If all *vehicles* in service have enough machines working on it, the first *vehicle* waiting at the station goes into service and the idle machine starts working on this *vehicle*. We assume that a machine changing of *vehicle* and therefore possibly also of commodity, does not take any time.

At the loading station the handling order of the *vehicles* is a bit different. A *vehicle* can only go into service when a certain percentage of its load is already in the storage area. When the first *vehicle* wants a commodity that is not in stock, while the second *vehicle* in the queue wants another commodity which is in the storage area, this second *vehicle* will be served first. When this percentage is set to 0 percent, the *vehicles* are served FIFO.

In a real terminal the quay has besides a limited amount of machines, also a maximum number of berths. Lets take an example of vessels which have a maximum of three machines working on it and a quay with five machines. The first vessel arrives and will be served by the first three machines. Then the second one arrives and in this model the service will start with the remaining two machines. In this research is assumed that there is always enough space at the quay to moor all ships assigned to the machines, so there is no maximum number of berths. It is also possible that a machine in the middle of the quay becomes idle and gets assigned to a vessel moored at the end of the quay, because that vessel has not reached its maximum number of machines yet. It could be that this machine actually can not reach this vessel, because it then has to move along two other cranes. This is, however, more a problem of the berth schedule than of the storage level, and therefore ignored as well. So we only take into account the number of machines in the (un)loading stations and not how those machines can reach the *vehicles* and where those *vehicles* are.

The terminal has only space for a maximum number of (waiting) *vehicles* at both the stations. If a *vehicle* arrives at the terminal and sees the maximum number of *vehicles* at the station, it will not join the queue, but leaves without being handled and will not come back. This maximum number of *vehicles* is independent of the types and commodities of the present *vehicles* and could also be infinite.

The terminal has also a maximum capacity of storage area. When the storage yard is full, all machines in the unloading station will stop working. All *vehicles* at this station, in service and waiting, will stay and wait. If in the loading station product is handled into a *vehicle* and removed from the storage yard, the unloading station can start handling product again. All machines start with a new handling of a unit of product, there is no remaining handling time from before the stop. Because we are interested in the required storage capacity of the terminal, this maximum storage capacity will in the models be set to infinity.

If in the loading station a *vehicle* is in service with a wanted commodity that is not in stock, the machines working on this *vehicle* will stop and the *vehicle* waits in service. When new product of this commodity arrives in the stockyard, the machines can start a new handling of one unit of product into this *vehicle*. The machines will not be assigned to other *vehicles* that pick up other commodities which are still in stock.

If a *vehicle* is almost completely served or a certain commodity is almost out of stock, some machines will stop working earlier than the point it is completely empty. For example, there are only two units of product A in stock and a third machine will start loading this commodity into a *vehicle*. In a realistic case, this machine will not start the handling of a new unit of product, because the two units are already handled by two other machines. In this model we assume that we do not know what other machines are doing. So all machines are working on the last unit of product and when the first machine finishes its

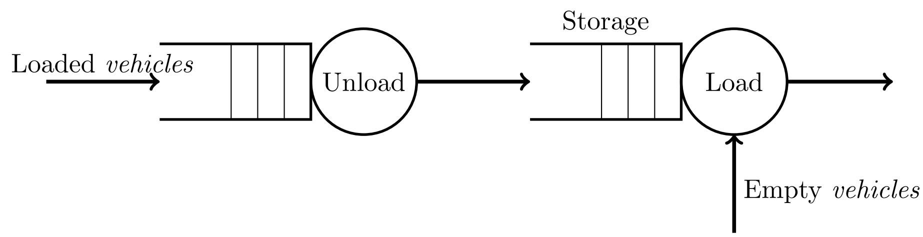

Figure 3.2: Queueing system terminal

handling time, all machines will stop working. One of the disadvantages of this assumption is that in the current handling time, a new unit of product could be unloaded and arrived in the storage yard. The start of the loading process of this unit then actually already started before it was in the storage yard. But this disadvantage will not have a large impact on the storage level, since the terminal handles large amounts of product, so these one or two units will not make the difference.

All combinations of *vehicle* type and commodity type have their own arrival distribution independent of each other. The load size distributions of the *vehicles* are independent of the commodity.

3.2 Queueing theory

As mentioned in chapter 2 one way to look at a terminal is as a network of queues. In figure 3.2 a model of two queues is shown which represents a terminal. The first queue is the unloading station where *vehicles* arrive with a batch of product. This first server has s_1 machines that handle the batches of product per unit. After this server the units of product go into the waiting line of the second server, which can be seen as the storage yard. The second server is the loading station with s_2 machines. When a *vehicle* arrives here the machines start loading units of product from the storage queue into this *vehicle*. This continues until the load size of this *vehicle* is loaded, and the *vehicle* will leave the system.

In this system we are interested in the queue length distribution of the second queue. First the two queues will be explored separately, so the departure process of the first queue and the queue length distribution of the second queue. Then the two queues will be combined in one model.

Chapter 4

Departure process of the unloading station

The arrival process of units of product into the storage yard is equal to the departure process of units of product out of the unloading station. When the *vehicles* at this station arrive in a Poisson process bringing one unit of product at a time and the s_1 machines in the station have all the same exponential service distribution, this station is a $M|M|s_1$ queue. That system has a departure distribution equal to the arrival distribution (Ross, 2006, corollary 6.6).

4.1 $M^X|M|1$ queue

When the *vehicles* do not bring only one unit of product, but deliver batches of product into the station, and there is only one machine in this station, the system changes to an $M^X|M|1$ queue. For this queue the departure process of units of product from the station into the storage yard can be divided in two cases. The first case is when after a departure there are still unloaded *vehicles* in the station. The second case is when the queue is empty after an end of handling of one unit of product.

For the first case the time until a new departure is equal to the time of a new end of service, since a new unit of product immediately goes into service. Let $F_B(t)$ be the probability that the service time is less or equal than t. For an exponential service with rate μ_1 this is $F_B(t) = 1 - \exp(-\mu_1 t)$.

When the queue is empty after an end of handling, the new departure first has to wait for a new arrival and then has to unload a first unit of product. Let $F_A(t)$ be the probability that the inter-arrival time between two batches of product is less or equal than t and let $F_{A+B}(t)$ be the probability that the inter-arrival time plus the service time is less or equal than t. Assume that the batches arrive in a Poisson process with rate λ, then $F_A(t) = 1-\exp(-\lambda t)$. The joint probability for $F_{A+B}(t)$ now can be determined, using the fact that the arrival process is memoryless.

$$\begin{aligned}
F_{A+B}(t) = Pr(A+B \leq t) &= \int_{y=0}^{t} Pr(B \leq t-y)Pr(A=y)dy \\
&= \int_{y=0}^{t} F_B(t-y)dF_A(y) = \int_{y=0}^{t} \big(1-\exp(-\mu_1(t-y))\big)\lambda \exp(-\lambda y)dy \\
&= 1 - \frac{\mu_1}{\mu_1-\lambda}\exp(-\lambda t) + \frac{\lambda}{\mu_1-\lambda}\exp(-\mu_1 t).
\end{aligned}$$

Now the cumulative distribution function (CDF) for the inter-departure times ($F_D(t)$) is a combination of $F_B(t)$ and $F_{A+B}(t)$. Let p_0 be the probability that there are no *vehicles* at the station, so the machine is working $1-p_0$ fraction of time. Let p_1 be the probability there is only one unit of remaining work in the station, so after the handling of that unit of product the station is empty. The fraction of times that the queue is empty after a departure is equal to $\frac{p_1}{1-p_0}$. The CDF for the inter-departure times is

$$\begin{aligned}
F_D(t) &= \left(1-\frac{p_1}{1-p_0}\right)F_B(t) + \frac{p_1}{1-p_0}F_{A+B}(t) \\
&= \left(1-\frac{p_1}{1-p_0}\frac{\mu_1}{\mu_1-\lambda}\right)(1-\exp(-\mu_1 t)) + \frac{p_1}{1-p_0}\frac{\mu_1}{\mu_1-\lambda}(1-\exp(-\lambda t)),
\end{aligned}$$

which is a hyperexponential distribution.

When the batch arrivals are just single arrivals, $p_0 = 1-\frac{\lambda}{\mu_1}$ and $p_1 = \frac{\lambda}{\mu_1}p_0$, so $\frac{p_1}{1-p_0} = \frac{\mu_1-\lambda}{\mu_1}$. This gives a departure distribution of $F_D(t) = 1-\exp(-\lambda t)$, which is a Poisson process with rate λ. In this case the queue is just $M|M|1$, for which was already mentioned that the departure process is Poisson.

When multiple commodities arrive to the terminal, but all types of *vehicles* arrive in independent Poisson processes, the departure process per commodity is also according to the given hyperexponential distribution. Since the sum of Poisson processes is still a Poisson process with a rate equal to the sum of the independent rates and the service times per commodity are all the same, the queue is still $M^X|M|1$. So the departure process $F_D(t)$ of all commodities is the hyperexponential distribution, but now $\lambda = \sum_c \lambda_c$ where λ_c is the arrival rate of *vehicles* with commodity c.

For different types of *vehicles* it works as well, but now p_0 and p_1 will be different because of the different batch sizes of arrivals.

4.2 $M^X|G|1$ queue

When the service times of the unloading machine are not exponentially distributed, but according to another, general, distribution, the queue becomes $M^X|G|1$. For this system the departure distribution can be determined on a same way as for the $M^X|M|1$ queue, but now with another service distribution.

After a departure there are again two possibilities, an empty system and a system with waiting *vehicles*. For the second case a new unit of product goes in service right away, so the time until the new departure has the same distribution as the service time, $F_B(t)$.

When the station is empty after a departure, the server has to wait for a new arrival. Since the *vehicles* arrive according to a Poisson process, the time until a new arrival is independent of the time of the departure. After the arrival of the first *vehicle* the server starts unloading. When the first unit of product is unloaded, the new departure occurs. So again the distribution $F_{A+B}(t)$ has to be determined.

$$\begin{aligned} F_{A+B}(t) = Pr(A+B \le t) &= \int_{y=0}^{t} Pr(B \le t-y|A=y)Pr(A=y)dy \\ &= \int_{y=0}^{t} F_B(t-y)f_A(y)dy = (F_B * f_A)(t), \end{aligned}$$

where $*$ is the convolution of the two functions and $f_A(t) = \lambda \exp(-\lambda t)$ because of the Poisson process of arrivals.

The ratio of $F_B(t)$ and $F_{A+B}(t)$ is again $\frac{p_1}{1-p_0}$, so

$$F_D(t) = \left(1 - \frac{p_1}{1-p_0}\right) F_B(t) + \frac{p_1}{1-p_0}(F_B * f_A)(t).$$

This equation works again for multiple commodities and multiple types of *vehicles* arriving at the loading station, as long as λ is adjusted to the right sum of all individual independent arrival processes and p_0 and p_1 are set to the right probabilities. The important assumption in this model is that all arrivals are independent Poisson processes, because only then the arrival processes are memoryless.

The departure distributions of these models are not easy to use in the models of the loading station, since the inter-departure times are not independent of each other. For example, when the queue of *vehicles* is empty, the first inter-departure time takes the time of a new arrival of a *vehicle* and the service of a first unit of product. The second inter-departure time will take only a service time, as long as the batches are larger than one unit of product. So after a relative long inter-departure time of an arrival and service, always a shorter, only service, inter-departure time will follow. Because dependent inter-departure times are harder to model than independent ones, the models in the next chapter will assume Poisson or Poisson batch arrivals into the storage yard. So these models ignore the service times of the unloading station.

Chapter 5

Queue length distribution of the loading station

The storage level of the terminal is in the queueing model equal to the queue length of products before the loading station. Therefore we are interested in the probability distribution of this queue. The system of this second queue is actually a double queue where both products and *vehicles* arrive and then have to be matched to each other. In this section we will look for the stationary probability distribution of the queue length. Let p_n be the probability that after an infinite time the storage area stores n units of product.

In this chapter all models assume only one commodity. In the last section 5.6 an explanation is given how these models can be expanded to more commodities. Other main assumptions are Poisson arrivals of *vehicles* and exponential service times of the machines in this station.

When we set the incoming flow of *vehicles* at the loading station such that it brings more workload than the inflow of products can supply, the queue of *vehicles* will grow to infinity. In this way it can be assumed that the machines in the loading station are always working as long as the storage area is not empty. This simplifies the system to a more general queue with only one flow instead of the double queue.

When the *vehicles* at the unloading station arrive in a Poisson process with load sizes of one unit, the arrivals of units of products to the loading station are also according to a Poisson process, see the introduction of chapter 4. If the s_2 machines in the loading station have all exponential service times with rate μ_2, this loading queue can be modelled as an $M|M|s_2$ queue, which has queue length probabilities

$$p_0 = \left(\sum_{i=0}^{s_2-1} \frac{(s_2\rho)^i}{i!} + \frac{(s_2\rho)^{s_2}}{s_2!} \frac{1}{1-\rho} \right)^{-1},$$
$$p_n = \begin{cases} \frac{(s_2\rho)^n}{n!} p_0, & \text{for } n = 1, 2, \dots, s_2; \\ \frac{s_2^{s_2}\rho^n}{s_2!} p_0, & \text{for } n = s_2, s_2+1, \dots, \end{cases} \tag{5.1}$$

with occupation rate $\rho = \frac{\lambda}{s_2\mu_2} < 1$. (Ross, 2006, section 8.9.2)

5.1 $M^G|M|1$ queue

If you assume that there is no unloading station, but that the goods are delivered in batches into the storage yard, and that the loading station has only one machine, the second queue could be seen as an $M^G|M|1$ queue. Here batches arrive according to a compound Poisson process with rate λ and batch sizes according to distribution G. Let g_k be the probability of an arrival of k units and assume $g_0 = 0$. The machine in the loading station has exponential service times with rate μ_2. Again an infinite queue of *vehicles* at the loading station is assumed, such that the machines will always work as long the storage area is not empty. The balance equations for this system are

$$\lambda p_0 = \mu_2 p_1,$$
$$(\lambda + \mu_2)p_n = \lambda \sum_{k=1}^{n} g_k p_{n-k} + \mu_2 p_{n+1}, \qquad \text{for } n > 0.$$

The generating function for this probability distribution is $\hat{P}(z) = \sum_{n=0}^{\infty} p_n z^n$ and for the batch size distribution $\hat{G}(z) = \sum_{k=1}^{\infty} g_k z^k$. By multiplying the balance equations with z^n and summing over all n we get

$$\lambda \sum_{n=0}^{\infty} p_n z^n + \mu_2 \sum_{n=1}^{\infty} p_n z^n = \lambda \sum_{n=1}^{\infty} \sum_{k=1}^{n} g_k p_{n-k} z^n + \mu_2 \sum_{n=0}^{\infty} p_{n+1} z^n,$$
$$\lambda \hat{P}(z) + \mu_2 \left(\hat{P}(z) - p_0 \right) = \lambda \sum_{k=1}^{\infty} g_k z^k \sum_{n=k}^{\infty} p_{n-k} z^{n-k} + \mu_2 z^{-1} \sum_{m=1}^{\infty} p_m z^m,$$
$$(\lambda + \mu_2)\hat{P}(z) - \mu_2 p_0 = \lambda \hat{G}(z) \sum_{j=0}^{\infty} p_j z^j + \mu_2 z^{-1} \left(\hat{P}(z) - p_0 \right),$$
$$\hat{P}(z) = \frac{\mu_2(1 - z^{-1})}{\lambda \left(1 - \hat{G}(z) \right) + \mu_2(1 - z^{-1})} p_0,$$

which is already stated in Chaudhry and Templeton (1983, section 3.1).

Using the properties of generating functions $\hat{P}(1) = 1$ and $\hat{P}'(1) = \mathbb{E}[P]$ and by applying L'Hôpital's rule, the value of p_0 can be determined.

$$
\begin{aligned}
1 = \lim_{z\to 1} \hat{P}(z) &= \lim_{z\to 1} \frac{\mu_2(1-z^{-1})}{\lambda\left(1-\hat{G}(z)\right)+\mu_2(1-z^{-1})} p_0 \\
&= \lim_{z\to 1} \frac{\mu_2 z^{-2}}{-\lambda \hat{G}'(z)+\mu_2 z^{-2}} p_0 = \frac{\mu_2}{-\lambda\,\mathbb{E}[G]+\mu_2} p_0, \\
p_0 = \frac{\mu_2 - \lambda\,\mathbb{E}[G]}{\mu_2} &= 1-\rho\,\mathbb{E}[G],
\end{aligned}
$$

with $\rho = \frac{\lambda}{\mu_2}$ and $\mathbb{E}[G]$ as the mean batch size.

By solving the balance equations directly the other steady state probabilities can be found.

Proposition 1. *For an $M^G|M|1$ queue with g_k as probability of a batch arrival of k units and $g_0 = 0$, the steady state probabilities of the queue length are*

$$
p_n = p_0\Big[\rho(\rho+1)^{n-1} + \sum_{\ell=1}^{n}(-\rho)^\ell \sum_{k=1}^{n-\ell}(\rho+1)^{n-k-\ell-1} \left(\frac{(n-k)!}{(n-k-\ell)!\ell!}\rho + \frac{(n-k-1)!}{(n-k-\ell)!(\ell-1)!}\right) C_g(\ell,k)\Big], \quad (5.2)
$$

for $n > 0$, with

$$
C_g(\ell,k) = \sum_{\substack{\bar{m}\in\mathbb{N}^\ell \\ m_1+\ldots+m_\ell=k}} \prod_{i=1}^{\ell} g_{m_i}.^{1}
$$

Proof. This proposition is proven by induction and using the balance equations for $i > 1$

$$
p_i = (\rho+1)p_{i-1} - \rho\sum_{m=1}^{i-2} g_m p_{i-1-m} - \rho g_{i-1} p_0.
$$

For $i = 1$ the proposition is true since $p_1 = \rho p_0$, which can be determined by equation 5.2 and by the balance equation of p_0.

For $i = 2$ the balance equation is $p_2 = (\rho+1)p_1 - \rho g_1 p_0 = p_0\left(\rho(\rho+1) - \rho g_1\right)$. This is equal to the proposition, since its sums have only positive terms for $\ell = 1$ and $k = 1$, so

$$
p_2 = p_0\left(\rho(\rho+1) + (-\rho)(\rho+1)^{-1}(\rho+1)\,C_g(1,1)\right) = p_0\left(\rho(\rho+1) - \rho g_1\right).
$$

[1] For example $C_g(3,5) = g_1g_1g_3 + g_1g_2g_2 + g_1g_3g_1 + g_2g_1g_2 + g_2g_2g_1 + g_3g_1g_1 = 3g_1^2g_3 + 3g_1g_2^2$.

Now assume that for all $i < n$ the proposition is true. The balance equation for $i = n$ can then be written as

$$\frac{p_n}{p_0} = -\rho g_{n-1} + \rho(\rho+1)^{n-1} + \sum_{\ell=1}^{n-1}(-\rho)^{\ell}\sum_{k=1}^{n-1-\ell}(\rho+1)^{n-k-\ell-1}$$
$$\left(\frac{(n-1-k)!}{(n-1-k-\ell)!\ell!}\rho + \frac{(n-k-2)!}{(n-1-k-\ell)!(\ell-1)!}\right) C_g(\ell,k)$$
$$+\sum_{r=2}^{n-1} g_{n-r}\Big[-\rho^2(\rho+1)^{r-2} + \sum_{\ell=1}^{r-1}(-\rho)^{\ell+1}\sum_{k=1}^{r-1-\ell}(\rho+1)^{r-k-\ell-2}$$
$$\left(\frac{(r-1-k)!}{(r-1-k-\ell)!\ell!}\rho + \frac{(r-k-2)!}{(r-1-k-\ell)!(\ell-1)!}\right) C_g(\ell,k)\Big].$$

Rearranging the terms and using $C_g(1,k) = g_k$, gives

$$\frac{p_n}{p_0} = \rho(\rho+1)^{n-1} - \sum_{k=1}^{n-2}\rho^2(\rho+1)^{n-k-2}C_g(1,k) - \rho C_g(1,n-1)$$
$$+\sum_{\ell=1}^{n-1}(-\rho)^{\ell}\sum_{k=1}^{n-1-\ell}(\rho+1)^{n-k-\ell-1}C_g(\ell,k)$$
$$\left(\frac{(n-1-k)!}{(n-1-k-\ell)!\ell!}\rho + \frac{(n-k-2)!}{(n-1-k-\ell)!(\ell-1)!}\right)$$
$$+\sum_{\ell=1}^{n-2}(-\rho)^{\ell+1}\sum_{s=2}^{n-\ell-1}(\rho+1)^{n-s-\ell-2}\sum_{r=n-s+1}^{n-1} g_{n-r}C_g(\ell, s-n+r)$$
$$\left(\frac{(n-1-s)!}{(n-1-s-\ell)!\ell!}\rho + \frac{(n-s-2)!}{(n-1-s-\ell)!(\ell-1)!}\right).$$

The sum $\sum_{r=n-s+1}^{n-1} g_{n-r}C_g(\ell, s-n+r)$ is equal to all combinations of $\ell+1$ g_i's with the sum of i's equal to s, so $C_g(\ell+1, s)$.

$$\frac{p_n}{p_0} = \rho(\rho+1)^{n-1} - \sum_{k=1}^{n-2}\rho^2(\rho+1)^{n-k-2}C_g(1,k) - \rho C_g(1,n-1)$$
$$+\sum_{\ell=1}^{n-1}(-\rho)^{\ell}\sum_{k=1}^{n-1-\ell}(\rho+1)^{n-k-\ell-1}C_g(\ell,k)$$
$$\left(\frac{(n-1-k)!}{(n-1-k-\ell)!\ell!}\rho + \frac{(n-k-2)!}{(n-1-k-\ell)!(\ell-1)!}\right)$$
$$+\sum_{\ell=2}^{n-1}(-\rho)^{\ell}\sum_{k=2}^{n-\ell}(\rho+1)^{n-k-\ell-1}C_g(\ell,k)$$
$$\left(\frac{(n-1-k)!}{(n-k-\ell)!(\ell-1)!}\rho + \frac{(n-k-2)!}{(n-k-\ell)!(\ell-2)!}\right).$$

Because $C_g(\ell, 1) = 0$ for $\ell > 1$, terms with these values can be added without changing the sum. This gives

$$\begin{aligned}
\frac{p_n}{p_0} &= \rho(\rho+1)^{n-1} - \sum_{k=1}^{n-2} \rho^2 (\rho+1)^{n-k-2} C_g(1,k) - \rho C_g(1, n-1) \\
&\quad + \sum_{l=2}^{n-1} (-\rho)^{\ell} C_g(\ell, n-\ell) - \rho \sum_{k=1}^{n-2} (\rho+1)^{n-k-2} \left((n-k-1)\rho + 1\right) C_g(1,k) \\
&\quad + \sum_{\ell=2}^{n-1} (-\rho)^{\ell} \sum_{k=1}^{n-1-\ell} (\rho+1)^{n-k-\ell-1} C_g(\ell,k) \\
&\quad \left(\frac{(n-1-k)!}{(n-1-k-\ell)!\ell!}\rho + \frac{(n-k-2)!}{(n-1-k-\ell)!(\ell-1)!} \right) \\
&\quad + \sum_{\ell=2}^{n-1} (-\rho)^{\ell} \sum_{k=1}^{n-l-1} (\rho+1)^{n-k-\ell-1} C_g(\ell,k) \\
&\quad \left(\frac{(n-1-k)!}{(n-k-\ell)!(\ell-1)!}\rho + \frac{(n-k-2)!}{(n-k-\ell)!(\ell-2)!} \right).
\end{aligned}$$

Rearranging the terms again leads to

$$\begin{aligned}
\frac{p_n}{p_0} &= \rho(\rho+1)^{n-1} + \sum_{\ell=2}^{n-1} (-\rho)^{\ell} \sum_{k=1}^{n-1-\ell} (\rho+1)^{n-k-\ell-1} \\
&\quad \left(\frac{(n-k)!}{(n-k-\ell)!\ell!}\rho + \frac{(n-k-1)!}{(n-k-\ell)!(\ell-1)!} \right) C_g(\ell,k) \\
&\quad + \sum_{\ell=1}^{n-1} (-\rho)^{\ell} C_g(\ell, n-\ell) - \rho \sum_{k=1}^{n-2} (\rho+1)^{n-k-2} \left((n-k)\rho + 1\right) C_g(1,k).
\end{aligned}$$

The last term of this equation is the second term for $\ell = 1$ and the second-last term is equal to this second term for $k = n - \ell$. This concludes to

$$\begin{aligned}
\frac{p_n}{p_0} &= \rho(\rho+1)^{n-1} + \sum_{\ell=1}^{n-1} (-\rho)^{\ell} \sum_{k=1}^{n-\ell} (\rho+1)^{n-k-\ell-1} \\
&\quad \left(\frac{(n-k)!}{(n-k-\ell)!\ell!}\rho + \frac{(n-k-1)!}{(n-k-\ell)!(\ell-1)!} \right) C_g(\ell,k).
\end{aligned}$$

The only term missing in this equation compared to equation 5.2 is for $\ell = n$, but then the sum of k goes from 1 to $n - n = 0$. This term is therefore not existing so can be added without changing the output. In this way it is proven that the outcome of the balance equation for $i = n$ is equal to the proposition. By induction can now be concluded that the steady state probabilities are as stated as in equation 5.2 for all $n > 0$. □

5.2 $M + M^r|M|1$ queue

In the previous section we assumed an infinite queue of *vehicles* at the loading station, which implies that the machines can always load product into some *vehicle*. But the system is actually a double queue with both arriving products and arriving *vehicles*. To take this fact into the model, a second variable will be added to the state of the model. Suppose the following system.

Units of product arrive in a Poisson process with rate λ. Empty *vehicles* arrive in a Poisson process with rate γ having a constant load size of r units of product. There is one machine with exponential service times with rate μ_2. These assumptions lead to an $M + M^r|M|1$ queue. Take as state of this queue (n, v), where n is the number of units of product in the storage area and v the remaining work of the present *vehicles* in units of product. This system has the following transitions.

- With rate λ a new unit of product arrives. $(n, v) \to (n + 1, v)$
- With rate γ a new *vehicle* with workload r arrives. $(n, v) \to (n, v + r)$
- With rate μ_2 one unit of product is loaded into the *vehicle*. $(n, v) \to (n - 1, v - 1)$

Take $p_{i,j} = Pr(n = i, v = j)$. The balance equations are

$$\begin{aligned}
(\lambda+\gamma)p_{0,j} &= \mu_2 p_{1,j+1}, && \text{for } j < r,\\
(\lambda+\gamma)p_{0,j} &= \mu_2 p_{1,j+1} + \gamma p_{0,j-r}, && \text{for } j \geq r,\\
(\lambda+\gamma)p_{i,0} &= \mu_2 p_{i+1,1} + \lambda p_{i-1,0}, && \text{for } i > 0,\\
(\lambda+\gamma+\mu_2)p_{i,j} &= \mu_2 p_{i+1,j+1} + \lambda p_{i-1,j}, && \text{for } i > 0 \text{ and } 0 < j < r,\\
(\lambda+\gamma+\mu_2)p_{i,j} &= \mu_2 p_{i+1,j+1} + \lambda p_{i-1,j} + \gamma p_{i,j-r}, && \text{for } i > 0 \text{ and } j \geq r.
\end{aligned}$$

Now take the generating function $\hat{P}(y,z) = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} p_{i,j} y^i z^j$. Summing over all balance equations and multiplying them with $y^i z^j$ gives

$$\begin{aligned}
&(\lambda+\gamma)\hat{P}(y,z) + \mu_2 \sum_{i=1}^{\infty}\sum_{j=1}^{\infty} p_{i,j} y^i z^j\\
&\quad = \mu_2 \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} p_{i+1,j+1} y^i z^j + \lambda \sum_{i=1}^{\infty}\sum_{j=0}^{\infty} p_{i-1,j} y^i z^j + \gamma \sum_{i=0}^{\infty}\sum_{j=r}^{\infty} p_{i,j-r} y^i z^j,
\end{aligned}$$

$$\begin{aligned}
&(\lambda(1-y) + \gamma(1-z^r) + \mu_2(1-(yz)^{-1}))\hat{P}(y,z)\\
&\qquad\qquad = \mu_2\left(1-(yz)^{-1}\right)\left(\sum_{i=1}^{\infty} p_{i,0} y^i + \sum_{j=0}^{\infty} p_{0,j} z^j\right).
\end{aligned}$$

This last sum between the brackets is the probability that the machine is not working, lets define $\hat{p}_{idle}(y,z) = \sum_{i=1}^{\infty} p_{i,0} y^i + \sum_{j=0}^{\infty} p_{0,j} z^j$. Then the generating function becomes

$$\hat{P}(y,z) = \frac{\mu_2 \left(1-(yz)^{-1}\right)}{\lambda(1-y)+\gamma(1-z^r)+\mu_2\left(1-(yz)^{-1}\right)} \hat{p}_{idle}(y,z).$$

Instead of looking to the probabilities of these two-dimensional states, it is also possible to only look at the probability of the storage level. Therefore let p_i be the probability of i units of product in storage, independent of the remaining workload in the system. So $p_i = \sum_{j=0}^{\infty} p_{i,j}$. Using the balance equations for $n = 0$ we get

$$\begin{aligned}
(\lambda+\gamma)\sum_{j=0}^{\infty} p_{0,j} &= \mu_2 \sum_{j=0}^{\infty} p_{1,j+1} + \gamma \sum_{j=r}^{\infty} p_{0,j-r}, \\
(\lambda+\gamma) p_0 &= \mu_2 \sum_{j=1}^{\infty} p_{1,j} + \gamma \sum_{j=0}^{\infty} p_{0,j}, \\
\lambda p_0 &= \mu_2 (p_1 - p_{1,0}).
\end{aligned}$$

So $p_1 = \frac{\lambda}{\mu_2} p_0 + p_{1,0}$. In the same way the probabilities for $i > 1$ can be determined.

$$\begin{aligned}
(\lambda+\gamma)\sum_{j=0}^{\infty} p_{1,j} + \mu_2 \sum_{j=1}^{\infty} p_{1,j} &= \mu_2 \sum_{j=0}^{\infty} p_{2,j+1} + \lambda \sum_{j=0}^{\infty} p_{0,j} + \gamma \sum_{j=r}^{\infty} p_{1,j-r}, \\
(\lambda+\gamma+\mu_2) p_1 - \mu_2 p_{1,0} &= \mu_2 (p_2 - p_{2,0}) + \lambda p_0 + \gamma p_1, \\
p_2 &= \frac{\lambda}{\mu_2} p_1 + p_{2,0}. \\
p_i = \frac{\lambda}{\mu_2} p_{i-1} + p_{i,0} &= \left(\frac{\lambda}{\mu_2}\right)^i p_0 + \sum_{k=0}^{i-1} \left(\frac{\lambda}{\mu_2}\right)^k p_{i-k,0}, \qquad \text{for } i > 0.
\end{aligned}$$

The sum over all probabilities has to be one, which gives

$$\begin{aligned}
1 = \sum_{i=0}^{\infty} p_i = p_0 + \sum_{i=1}^{\infty} \left(\rho^i p_0 + \sum_{k=0}^{i-1} \rho^k p_{i-k,0}\right) &= \sum_{i=0}^{\infty} \rho^i p_0 + \sum_{i=1}^{\infty} \sum_{k=0}^{i-1} \rho^k p_{i-k,0} \\
&= \frac{1}{1-\rho} p_0 + \sum_{k=0}^{\infty} \rho^k \sum_{j=1}^{\infty} p_{j,0} = \frac{1}{1-\rho}\left(p_0 + \sum_{j=1}^{\infty} p_{j,0}\right),
\end{aligned}$$

with $\rho = \frac{\lambda}{\mu_2}$. This last sum between the brackets can be seen as p_{idle}.

To make sure this system is stable, the inflow of products has to be equal to the inflow of workload, so $\lambda = \gamma r$. By determining the distribution of this $1-\rho$ over the elements in p_{idle}, all steady state probabilities of the storage level in this model can be found.

5.3 Virtual storage level

Instead of the two dimensional state (n, v), you could also look at the system with a one dimensional state $(n - v)$. So the stock level minus the remaining work already in the system. We now also assume that there is finite maximum of *vehicles* (V) that can be at the station. Therefore the state space has range $[-rV, \infty)$. In this model we will look at the virtual storage level, so when a *vehicle* arrives, it is immediately loaded completely. Another way to see this, is that there are no loading machines, or these loading machines are infinitely fast. The transitions in this system are

- With rate λ a new unit of product arrive. $(m) \rightarrow (m + 1)$
- With rate γ a new *vehicle* with workload r arrives. This *vehicle* only goes into the system if not already V *vehicles* are at the station. $(m) \rightarrow (m - r)$

Again both arrival processes are assumed to be Poisson processes.

This model is only stable if the incoming flow of workload is more than the incoming flow of products, because of the limitation of the queue of *vehicles*. This is the same as $\frac{\lambda}{\gamma r} < 1$.

To simplify the notation, we will shift the states with $-rV$, such that all the states are positive. So $p_i = Pr(n - v = i - rV)$. The balance equations for this system are

$$\begin{aligned} \lambda p_0 &= \gamma p_r, \\ \lambda p_i &= \lambda p_{i-1} + \gamma p_{i+r}, && \text{for } 0 < i < r, \\ (\lambda + \gamma) p_i &= \lambda p_{i-1} + \gamma p_{i+r}, && \text{for } i \geq r. \end{aligned}$$

Lets take as basic solution $p_i = x^{i-r} p_r$ for $i \geq r$, the balance equations for $i > r$ then become

$$\begin{aligned} (\lambda + \gamma) x^{i-r} p_r &= \lambda x^{i-r-1} p_r + \gamma x^i p_r, \\ 0 &= \gamma x^{r+1} - (\lambda + \gamma) x + \lambda. \end{aligned} \tag{5.3}$$

The value of x in the basic solution of p_i has to satisfy this balance equation 5.3, and because the summation of all p_i has to be one, the value of x also has to be in the interval $(0, 1)$.

Proposition 2. *If* $\lambda < \gamma r$*, then the function* $L(x) = \gamma x^{r+1} - (\lambda + \gamma) x + \lambda$ *has one real root in the interval* $(0, 1)$.

Proof. The second derivative of $L(x)$ is $\gamma r(r + 1) x^{r-1}$, which is nonnegative on the interval $[0, 1]$. This implies that the function is convex on this interval. On the borders of the interval the values of L are $L(0) = \lambda > 0$ and $L(1) = 0$. This implies, together with the convexity, that the function is either strictly positive on $(0, 1)$ or has another root in this interval.

The minimum of a convex function is the point where the first derivative is equal to 0. For $L(x)$ this is

$$0 = L'(x_{min}) = \gamma(r+1)x_{min}^r - \lambda - \gamma,$$
$$x_{min} = \sqrt[r]{\frac{\lambda+\gamma}{\gamma(r+1)}}.$$

When $\lambda < \gamma r$, the value of the fraction in x_{min} is

$$0 < \frac{\lambda+\gamma}{\gamma(r+1)} < \frac{\gamma r+\gamma}{\gamma(r+1)} = 1.$$

So the value of x_{min} lies in the interval $(0,1)$. The value of $L(1) = 0$, which has to be above the minimum. This implies that $L(x_{min}) < 0$, so the function is not strictly positive in the interval $(0,1)$. By combing this with the convexity of $L(x)$, is proven that $L(x)$ has one real root in the interval $(0,1)$. $\square$

Because this system is only stable if $\lambda < \gamma r$, the balance equations for $i > r$ give one real solution $\bar{x}$ to use in the basic solution of p_i. The balance equations for $i < r$ give

$$\begin{aligned}
p_0 &= \frac{\gamma}{\lambda}p_r,\\
p_1 &= p_0 + \frac{\gamma}{\lambda}xp_r = \frac{\gamma}{\lambda}(1+x)p_r,\\
p_i &= p_{i-1} + \frac{\gamma}{\lambda}x^i p_r = \frac{\gamma}{\lambda}\sum_{k=0}^{i} x^k p_r, \qquad \text{for } 0 < i < r.
\end{aligned}$$

The value of p_r can now be determined with the normalisation equation.

$$\begin{aligned}
1 &= \sum_{i=0}^{\infty} p_i = \sum_{i=0}^{r-1}\frac{\gamma}{\lambda}\sum_{k=0}^{i} x^k p_r + \sum_{i=r}^{\infty} x^{i-r}p_r,\\
\frac{1}{p_r} &= \frac{\gamma}{\lambda}\sum_{k=0}^{r-1}(r-k)x^k + \frac{1}{1-x} = \frac{\gamma x(x^r-1)+\gamma r(1-x)+\lambda(1-x)}{\lambda(1-x)^2}.
\end{aligned}$$

By filling in $\bar{x}$ and using equation 5.3 this equation is equal to

$$p_r = \frac{\lambda}{\gamma r}(1-\bar{x}),$$

which concludes to

$$p_i = \begin{cases} \frac{1}{r}(1-\bar{x}^{i+1}), & \text{for } i = 0,1,\ldots,r-1;\\ \frac{\lambda}{\gamma r}(1-\bar{x})\bar{x}^{i-r}, & \text{for } i = r, r+1, \ldots. \end{cases}$$

Because of the shifting that was done to get only positive states, the states now can be shifted back. Let $p_m = Pr(n - v = m)$. The probabilities of positive virtual storage levels are

$$p_m = \frac{\lambda}{\gamma r}(1 - \bar{x})\bar{x}^{m+rV-r} \qquad \text{for } m = 0, 1, \ldots. \tag{5.4}$$

5.4 Breakdowns

In the previous sections the second state variable saved the remaining workload of the *vehicles* and could get an infinite amount of different values. This induces a large state space with two infinite variables. To reduce this size of the state space, another second variable could be taken. This variable only has two values; working and not-working. When a *vehicle* arrives at the loading station the machine starts working and the variable will be set to 1. When the *vehicle* leaves the terminal, the machine can not load anything anymore. This can be seen as a breakdown of the machine and the second variable will be set to 0.

Suppose the following system. Units of product arrive to the loading station in a Poisson process with rate λ with single arrivals. In this station is one machine with exponential service times with rate μ_2. This machine is working for an exponentially distributed time with mean $\frac{1}{\theta}$. When the *vehicle* leaves the system or when the storage area is empty, the machine goes in downtime. The time until it starts working again is exponentially distributed with a mean time of $\frac{1}{\gamma}$. This process is an $M|M|1$ queue with breakdowns.

The states of this model will be (n, w) where n is the number of units of product in the queue and w is the state of the machine. If the machine is working $w = 1$ and $w = 0$ if it is in downtime. Take $p_{i,j} = \text{Prob}(n = i, w = j)$. The balance equations are

$$\begin{aligned}
\lambda p_{0,0} &= \mu_2 p_{1,1}, \\
\lambda p_{0,1} &= 0, \\
(\lambda + \gamma) p_{i,0} &= \theta p_{i,1} + \lambda p_{i-1,0}, && \text{for } i \geq 1, \\
(\lambda + \theta + \mu_2) p_{i,1} &= \gamma p_{i,0} + \mu_2 p_{i+1,1} + \lambda p_{i-1,1}, && \text{for } i \geq 1.
\end{aligned}$$

This is a quasi birth-death process with generator

$$Q = \begin{bmatrix} B_0 & A_0 & 0 & \ldots \\ B_1 & A_1 & A_0 & \ddots \\ 0 & A_2 & A_1 & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix},$$

with

$$
B_0 = \begin{bmatrix} -\lambda & 0 \\ 0 & -\lambda \end{bmatrix}, \qquad B_1 = \begin{bmatrix} 0 & 0 \\ \mu_2 & 0 \end{bmatrix}, \qquad A_0 = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix},
$$
$$
A_1 = \begin{bmatrix} -(\lambda+\gamma) & \gamma \\ \theta & -(\lambda+\theta+\mu_2) \end{bmatrix}, \qquad A_2 = \begin{bmatrix} 0 & 0 \\ 0 & \mu_2 \end{bmatrix}.
$$

Take $p_i = [p_{i,0}, p_{i,1}]$ and lets try a basic solution $p_i = p_{i-1}R = p_1 R^{i-1}$. By the balance principle the flow from p_i to p_{i+1} has to be equal to the flow from p_{i+1} to p_i. So $\lambda p_{i,0} + \lambda p_{i,1} = \mu_2 p_{i+1,1}$, or $p_i A_3 = p_{i+1} A_2$, with

$$
A_3 = \begin{bmatrix} 0 & \lambda \\ 0 & \lambda \end{bmatrix}.
$$

The balance equations for $i \geq 1$ can in this way be written as

$$
\begin{aligned}
0 &= p_{i-1}A_0 + p_i A_1 + p_{i+1}A_2, \\
0 &= p_{i-1}A_0 + p_i(A_1 + A_3).
\end{aligned}
$$

Because the determinant of $(A_1 + A_3)$ is $\mu_2(\lambda+\gamma) \neq 0$, matrix $(A_1 + A_3)$ is invertible. This gives $R = -A_0(A_1 + A_3)^{-1}$.

$$
\begin{aligned}
R &= -\begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \left(\begin{bmatrix} -(\lambda+\gamma) & \gamma \\ \theta & -(\lambda+\theta+\mu_2) \end{bmatrix} + \begin{bmatrix} 0 & \lambda \\ 0 & \lambda \end{bmatrix} \right)^{-1} \\
&= \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \begin{bmatrix} \frac{(\theta+\mu_2)}{\mu_2(\lambda+\gamma)} & \frac{1}{\mu_2} \\ \frac{\theta}{\mu_2(\lambda+\gamma)} & \frac{1}{\mu_2} \end{bmatrix} = \frac{\lambda}{\mu_2(\lambda+\gamma)} \begin{bmatrix} \theta+\mu_2 & \lambda+\gamma \\ \theta & \lambda+\gamma \end{bmatrix}.
\end{aligned}
$$

To find the values of p_0 and p_1, the boundary equations together with the normalisation equation are used. The boundary equations are the balance equations for i equal to 0 and 1, which give

$$
\begin{aligned}
0 &= -\lambda p_{0,1}, & &\implies p_{0,1} = 0, \\
0 &= -\lambda p_{0,0} + \mu_2 p_{1,1}, & &\implies p_{1,1} = \frac{\lambda}{\mu_2} p_{0,0}, \\
0 &= \theta p_{1,1} - (\lambda+\gamma)p_{1,0} + \lambda p_{0,0}, & &\implies p_{1,0} = \frac{\lambda(\theta+\mu_2)}{\mu_2(\lambda+\gamma)} p_{0,0}.
\end{aligned}
$$

The normalisation equation gives

$$
1 = \sum_{i=0}^{\infty} p_i e = p_0 e + p_1(I + R + R^2 + \ldots)e = p_0 e + p_1(I - R)^{-1} e,
$$

where I is the identity matrix and e a vector of ones.

Using the determined values of R, p_0 and p_1, this normalisation equation can be written as

$$\begin{aligned}
1 &= \begin{bmatrix} p_{0,0} & p_{0,1} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} p_{1,0} & p_{1,1} \end{bmatrix} \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} \frac{\lambda(\theta+\mu_2)}{\mu_2(\lambda+\gamma)} & \frac{\lambda}{\mu_2} \\ \frac{\lambda\theta}{\mu_2(\lambda+\gamma)} & \frac{\lambda}{\mu_2} \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\
&= p_{0,0} + \frac{p_{0,0}}{\mu_2\gamma - \lambda\theta - \lambda\gamma} \begin{bmatrix} \frac{\lambda(\theta+\mu_2)}{\mu_2(\lambda+\gamma)} & \frac{\lambda}{\mu_2} \end{bmatrix} \begin{bmatrix} (\mu_2-\lambda)(\lambda+\gamma) & \lambda(\lambda+\gamma) \\ \lambda\theta & \mu_2\gamma - \lambda\theta \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\
&= p_{0,0} \frac{\mu_2(\lambda+\gamma)}{\mu_2\gamma - \lambda\theta - \lambda\gamma}, \\
p_{0,0} &= \frac{\mu_2\gamma - \lambda\theta - \lambda\gamma}{\mu_2(\lambda+\gamma)} = 1 - \frac{\lambda(\gamma+\theta+\mu_2)}{\mu_2(\lambda+\gamma)}.
\end{aligned}$$

This queueing system can therefore only be stable if $\frac{\lambda(\gamma+\theta+\mu_2)}{\mu_2(\lambda+\gamma)} < 1$, which is equal to $\frac{\lambda}{\mu_2} < \frac{\gamma}{\theta+\gamma}$.

Now for all $i > 0$ applies

$$p_i = \left(\frac{\lambda}{\mu_2(\lambda+\gamma)} \right)^i \frac{\mu_2\gamma - \lambda\theta - \lambda\gamma}{\mu_2} \begin{bmatrix} \frac{(\theta+\mu_2)}{(\lambda+\gamma)} & 1 \end{bmatrix} \begin{bmatrix} (\theta+\mu_2) & (\lambda+\gamma) \\ \theta & (\lambda+\gamma) \end{bmatrix}^{i-1}. \quad (5.5)$$

To get realistic results for a terminal, the values of the variables λ, μ_2, γ and θ need to be determined. The first three are given as input, since λ is the arrival rate of *vehicles* to the unloading station, μ_2 is the service rate of the loading station and γ is the arrival rate of *vehicles* at the loading station. The value of θ could be seen as the inverse of the mean loading time of a *vehicle* at the loading station. So if these *vehicles* have load size r, θ would be $\frac{\mu_2}{r}$. In this setting the stability condition can be written as

$$\begin{aligned}
\frac{\lambda}{\mu_2} &< \frac{\gamma}{\theta+\gamma} = \frac{\gamma}{\frac{\mu_2}{r}+\gamma} = \frac{\gamma r}{\mu_2 + \gamma r}, \\
\lambda\gamma r &< \mu_2(\gamma r - \lambda).
\end{aligned}$$

For a terminal you would expect that the inflow of products is equal to the inflow of workload, so $\lambda = \gamma r$, but in that case $\lambda\gamma r$ has to be negative, which is impossible.

A reason that he value of θ is not exactly $\frac{\mu_2}{r}$, is because it is possible that a new *vehicle* arrives while another *vehicle* is still being loaded. The time the machines are now working continuously is doubled, which will increase the mean time the machines are not in downtime. So we are interested in the probabilities that a new *vehicle* arrives, before the previous one has left the terminal.

The *vehicles* arrive in a Poisson process with rate γ and lets assume a constant loading time per *vehicle* T, with $T = \frac{r}{\mu_2}$. The probability that a busy period of the loading station only takes one *vehicle* is equal to the probability that the inter-arrival time between two successive *vehicles* is more than T. Let P_i

be the probability that a busy period handles $i > 0$ *vehicles* and A_i the time between the start of a busy period and the arrival time of the i'th *vehicle* after this start. Because the *vehicles* arrive in a Poisson process $P_1 = Pr(A_1 > T) = \exp(-\gamma T)$.

For P_2 the first arrival has to be before T and the second one has to be after $2T$. This gives

$$\begin{aligned}
P_2 &= Pr(A_1 < T, A_2 > 2T) \\
&= \int_{a_1=0}^{T} Pr(A_1 = a_1) Pr(A_2 - A_1 > 2T - a_1 | A_1 = a_1) da_1 \\
&= \int_{a_1=0}^{T} \gamma \exp(-\gamma a_1) \exp(-\gamma(2T - a_1)) da_1 = \gamma T \exp(-2T\gamma).
\end{aligned}$$

For larger i the idea is the same.

$$\begin{aligned}
P_3 &= Pr(A_1 < T, A_2 < 2T, A_3 > 3T) \\
&= \int_{a_1=0}^{T} \gamma \exp(-\gamma a_1) \int_{a_2=a_1}^{2T} \gamma \exp(-\gamma(a_2 - a_1)) \exp(-\gamma(3T - a_2)) da_1 da_2 \\
&= \gamma^2 \exp(-3T\gamma) \int_{a_1=0}^{T} \int_{a_2=a_1}^{2T} da_1 da_2 = \frac{3}{2} T^2 \gamma^2 \exp(-3T\gamma). \\
P_4 &= \gamma^3 \exp(-4T\gamma) \int_{a_1=0}^{T} \int_{a_2=a_1}^{2T} \int_{a_3=a_2}^{3T} da_1 da_2 da_3, = \frac{16}{6} T^3 \gamma^3 \exp(-4T\gamma). \\
P_i &= \frac{i^{i-2}}{(i-1)!} T^{i-1} \gamma^{i-1} \exp(-iT\gamma), \qquad \text{for } i > 0.
\end{aligned}$$

The mean time of a busy period of the loading station is

$$\frac{1}{\theta} = \sum_{i=1}^{\infty} iTP_i = \sum_{i=1}^{\infty} \frac{i^{i-1}}{(i-1)!} T^i \gamma^{i-1} \exp(-iT\gamma). \tag{5.6}$$

5.5 Finite queue of *vehicles*

In the previous section the state of the machine in the loading station was on or off, so two choices. Another value for this second variable can be the number of *vehicles* at the loading station, so the machines are not working if this number of *vehicles* is zero, and working if this variable is nonzero. Lets take as states of the system (n, v) where n is the storage level en v the number of *vehicles* at the terminal. This number of *vehicles* will have an upper limit of V, so there can not be more than V *vehicles* simultaneously at the loading station.

The units of product again arrive in a Poisson process with rate λ in single arrivals and the *vehicles* arrive in a Poisson process with rate γ. The loading machine has exponential service times per unit of product with rate μ_2 and with probability q this is the last unit of the *vehicle*, after which the *vehicle* will leave the system. So the workload of a *vehicle* is geometrically distributed with parameter q.

The balance equations of this system are

$$\begin{aligned}
(\lambda+\gamma)p_{0,0} &= q\mu_2 p_{1,1}, \\
(\lambda+\gamma)p_{0,j} &= \gamma p_{0,j-1} + (1-q)\mu_2 p_{1,j} + q\mu_2 p_{1,j+1}, \qquad \text{for } 0<j<V, \\
\lambda p_{0,V} &= \gamma p_{0,j-1} + (1-q)\mu_2 p_{1,j}, \\
(\lambda+\gamma)p_{i,0} &= \lambda p_{i-1,0} + q\mu_2 p_{i+1,1}, \qquad \text{for } i>0, \\
(\lambda+\gamma+\mu_2)p_{i,j} &= \lambda p_{i-1,j} + \gamma p_{i,j-1} + (1-q)\mu_2 p_{i+1,j} + q\mu_2 p_{i+1,j+1}, \\
&\qquad \text{for } \ i>0 \text{ and } 0<j<V, \\
(\lambda+\mu_2)p_{i,V} &= \lambda p_{i-1,V} + \gamma p_{i,V-1} + (1-q)\mu_2 p_{i+1,V}, \qquad \text{for } i>0.
\end{aligned}$$

Take p_i as vector $[p_{i,0},\dots,p_{i,V}]$. In matrix notation the balance equations for $i>0$ are $p_{i-1}A_0 + p_iA_1 + p_{i+1}A_2 = 0$ with

$$A_0 = \begin{bmatrix} \lambda & 0 & \\ 0 & \ddots & 0 \\ & 0 & \lambda \end{bmatrix}, \qquad A_2 = \begin{bmatrix} 0 & 0 & & \\ q\mu_2 & (1-q)\mu_2 & & \\ & \ddots & \ddots & \\ & & q\mu_2 & (1-q)\mu_2 \end{bmatrix},$$

$$A_1 = \begin{bmatrix} -(\lambda+\gamma) & \gamma & & & \\ & -(\lambda+\gamma+\mu_2) & \gamma & & \\ & & \ddots & \ddots & \\ & & & -(\lambda+\gamma+\mu_2) & \gamma \\ & & & & -(\lambda+\mu_2) \end{bmatrix}.$$

Those three matrices have dimension $(V+1)\times(V+1)$.

The boundary equation for $i=0$ is $p_0B_0 + p_1A_2 = 0$ with

$$B_0 = \begin{bmatrix} -(\lambda+\gamma) & \gamma & & & \\ & -(\lambda+\gamma) & \gamma & & \\ & & \ddots & \ddots & \\ & & & -(\lambda+\gamma) & \gamma \\ & & & & -\lambda \end{bmatrix}.$$

This model is again a quasi birth-death process with generator

$$Q = \begin{bmatrix} B_0 & A_0 & 0 & \dots \\ A_2 & A_1 & A_0 & \ddots \\ 0 & A_2 & A_1 & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}.$$

Lets take as basic solution $p_i = p_0 R^i$ for all i, where R is the minimal nonnegative solution of $A_0 + RA_1 + R^2 A_2 = 0$. This can be iteratively solved through $R_{k+1} = -(A_0 + R_k^2 A_2) A_1^{-1}$ and starting with $R_0 = 0$. After finding R, the vector p_0 can be determined by the boundary equation $0 = p_0 B_0 + p_1 A_2 = p_0(B_0 + RA_2)$ and normalising equation $1 = p_0(I - R)^{-1} e$.

5.6 Multiple commodities

All models in this chapter are about terminals with only one commodity. When a terminal has more than one commodity, the systems can be adjusted in two ways.

The first way is just to see all commodities as one commodity and sum up over all the incoming *vehicles* and the storage level. This is a relatively easy way to make it usable for the models. It, however, will probably give lower storage levels than in the real case. A *vehicle* at the loading station will now be loaded with all commodities instead of only one, so the terminal will have less waiting times and lower storage levels.

Another solution is to give all commodities an individual terminal. When there are, for example, three commodities, three probability functions for the storage levels, $F_c(x) = Pr(n_c \leq x)$ with n_c the storage level of commodity c, can be determined. The probability function of the total storage level is the convolution of these individual probability functions per commodity. For the example with three commodities it will be

$$F(x) = Pr(n_1 + n_2 + n_3 \leq x) = \int_{y=0}^{x} Pr(n_2 + n_3 \leq x - y) dF_1(y)$$
$$= \int_{y=0}^{x} \int_{z=0}^{x-y} F_3(x - y - z) dF_2(z) dF_1(y) = (F_1 * F_2 * F_3)(x).$$

In this way all *vehicles* will get only one commodity, which is an advantage over the first solution. A disadvantage of this solution is that *vehicles* with different commodities can be loaded simultaneously in the virtual different terminals, which is not possible in the real terminal.

Chapter 6

Queue length distribution of the total system

A terminal is a combination of the unloading and loading station. To model this system we can use a similar model as in section 5.5, but now with a three dimensional state space.

6.1 Geometric load sizes

Let the state of the system be (n, v_1, v_2) with n the storage level, v_1 the number of *vehicles* at the unloading station and v_2 the number of *vehicles* at the loading station. Take $p_{i,j,k}$ as the steady state probability of state $(n = i, v_1 = j, v_2 = k)$. To get finite matrices, the number of *vehicles* at the unloading and loading station will be limited at V_1 and V_2 *vehicles* respectively. So when there are V_1 *vehicles* at the unloading station and a new one arrives, it will not go in to the system, but leaves without being unloaded.

We assume that the inter-arrival times of the *vehicles* at the unloading and loading station are exponentially distributed. At both stations there is only one machine with exponential handling times per unit of product with rate μ_1 and μ_2. With probability q_1 the handling of one unit of product was the last unit of product of that *vehicle* at the unloading station, and the *vehicle* will leave. At the loading station this is with probability q_2. In this way the load sizes of the *vehicles* have geometric distributions with a mean of $\frac{1}{q_1}$ and $\frac{1}{q_2}$ units of product.

The transitions of this model are as follows.

- With rate λ *vehicles* arrive at the unloading station as long as there are less than V_1 *vehicles* at this station. $(n, v_1, v_2) \to (n, \min(v_1 + 1, V_1), v_2)$

- With rate γ *vehicles* arrive at the loading station as long as there are less than V_2 *vehicles* at this station. $(n, v_1, v_2) \to (n, v_1, \min(v_2 + 1, V_2))$
- With rate μ_1 one unit of product is unloaded from a *vehicle* into the storage area. With probability q_1 the *vehicle* leaves, with probability $1 - q_1$ the *vehicles* stays in the system.
  - Rate $q_1\mu_1$ $\quad (n, v_1, v_2) \to (n + 1, v_1 - 1, v_2)$
  - Rate $(1 - q_1)\mu_1$ $\quad (n, v_1, v_2) \to (n + 1, v_1, v_2)$
- With rate μ_2 one unit of product is loaded from the storage area into a *vehicle*. With probability q_2 the *vehicle* leaves, with probability $1-q_2$ the *vehicles* stays in the system.
  - Rate $q_2\mu_2$ $\quad (n, v_1, v_2) \to (n - 1, v_1, v_2 - 1)$
  - Rate $(1 - q_2)\mu_2$ $\quad (n, v_1, v_2) \to (n - 1, v_1, v_2)$

The main balance equation for $i, j, k > 0$ and $j < V_1$ and $k < V_2$, is

$$(\lambda + \gamma + \mu_1 + \mu_2)p_{i,j,k} = \lambda p_{i,j-1,k} + \gamma p_{i,j,k-1} + q_1\mu_1 p_{i-1,j+1,k} \\ + (1 - q_1)\mu_1 p_{i-1,j,k} + q_2\mu_2 p_{i+1,j,k+1} + (1 - q_2)\mu_2 p_{i+1,j,k}.$$

This model is a quasi birth-death process, which has solutions of the form $p_i = p_0 R^i$. Here is p_i the vector $[p_{i,0,0}, \ldots, p_{i,V_1,0}, p_{i,0,1}, \ldots, p_{i,V_1,V_2}]$ and R a square matrix of order $(V_1 + 1)(V_2 + 1)$. R is the minimal nonnegative solution of equation $A_0 + RA_1 + R^2A_2 = 0$ with

$$A_0 = \begin{bmatrix} A_0' & & \\ & \ddots & \\ & & A_0' \end{bmatrix}, \quad A_0' = \begin{bmatrix} 0 & 0 & & & \\ q_1\mu_1 & (1-q_1)\mu_1 & & & \\ & \ddots & \ddots & & \\ & & & q_1\mu_1 & (1-q_1)\mu_1 \end{bmatrix},$$

with A_0' having dimension $(V_1 + 1) \times (V_1 + 1)$ and A_0 consists of $(V_2 + 1)$ A_0' sub-matrices.

$$A_1 = \begin{bmatrix} A_1' & A_1'''' & & & \\ & A_1'' & A_1'''' & & \\ & & \ddots & \ddots & \\ & & & A_1'' & A_1'''' \\ & & & & A_1''' \end{bmatrix},$$

$$A_1' = \begin{bmatrix} -(\lambda+\gamma) & \lambda & & & \\ & -(\lambda+\gamma+\mu_1) & \lambda & & \\ & & \ddots & \ddots & \\ & & & -(\lambda+\gamma+\mu_1) & \lambda \\ & & & & -(\gamma+\mu_1) \end{bmatrix},$$

$$A_1'' = \begin{bmatrix} -(\lambda+\gamma+\mu_2) & \lambda & & & \\ & -\Delta & \lambda & & \\ & & \ddots & \ddots & \\ & & & -\Delta & \lambda \\ & & & & -(\gamma+\mu_1+\mu_2) \end{bmatrix},$$

$$A_1''' = \begin{bmatrix} -(\lambda+\mu_2) & \lambda & & & \\ & -(\lambda+\mu_1+\mu_2) & \lambda & & \\ & & \ddots & \ddots & \\ & & & -(\lambda+\mu_1+\mu_2) & \lambda \\ & & & & -(\mu_1+\mu_2) \end{bmatrix},$$

$$A_1'''' = \begin{bmatrix} \gamma & & \\ & \ddots & \\ & & \gamma \end{bmatrix},$$

with $\Delta = \lambda + \gamma + \mu_1 + \mu_2$ and A_1', A_1'', A_1''' and A_1'''', having dimension $(V_1 + 1) \times (V_1 + 1)$. A_1 consists of $(V_2 + 1)$ sub-matrices on the diagonal.

$$A_2 = \begin{bmatrix} \mathbf{0} & & & \\ A_2'' & A_2' & & \\ & \ddots & \ddots & \\ & & A_2'' & A_2' \end{bmatrix},$$

$$A_2' = \begin{bmatrix} (1-q_2)\mu_2 & & \\ & \ddots & \\ & & (1-q_2)\mu_2 \end{bmatrix}, \qquad A_2'' = \begin{bmatrix} q_2\mu_2 & & \\ & \ddots & \\ & & q_2\mu_2 \end{bmatrix},$$

with $\mathbf{0}$ as zero square matrix of order $(V_1 + 1)$, A_2' and A_2'', having dimension $(V_1+1)\times(V_1+1)$ and A_2 consists of (V_2+1) sub-matrices on the diagonal.

R can again be iteratively found by using $R_{k+1} = -(A_0 + R_k^2 A_2)A_1^{-1}$ and starting with $R_0 = 0$. Then p_0 can be determined by the normalising equation $1 = p_0(I - R)^{-1}e$ and the boundary equation $0 = p_0(B_0 + RA_2)$, where B_0 is equal to A_1 without μ_2 in the sums on the diagonal.

6.2 Constant load sizes

Another way to model this total system is by changing the state variables from number of *vehicles* to remaining workload. So again the states are (n, v_1, v_2) with n as the number of units of product in the storage area, but now v_1 and v_2 as the remaining workload at the unloading and loading station in units of product. The last two variables will be limited at r_1V_1 and r_2V_2, where r_1 is the load size of the *vehicles* arriving at the unloading station and r_2 the load size of the *vehicles* arriving at the loading station. These load sizes are assumed to be constant for all arriving *vehicles*.

When the other distributions and parameters stay equal to the previous section, the transitions in this model become

- With rate λ *vehicles* arrive at the unloading station as long as there are less than V_1 *vehicles* at this station. $(n, v_1, v_2) \rightarrow (n, v_1 + r_1, v_2)$
- With rate γ *vehicles* arrive at the loading station as long as there are less than V_2 *vehicles* at this station. $(n, v_1, v_2) \rightarrow (n, v_1, v_2 + r_2)$
- With rate μ_1 one unit of product is unloaded from a *vehicle* into the storage area. $(n, v_1, v_2) \rightarrow (n + 1, v_1 - 1, v_2)$
- With rate μ_2 one unit of product is loaded from the storage area into a *vehicle*. $(n, v_1, v_2) \rightarrow (n - 1, v_1, v_2 - 1)$

So the main balance equation for $i, j, k > 0$ and $j < r_1V_1$ and $k < r_2V_2$, is

$$(\lambda + \gamma + \mu_1 + \mu_2)p_{i,j,k} = \lambda p_{i,j-r_1,k} + \gamma p_{i,j,k-r_2} + \mu_1 p_{i-1,j+1,k} + \mu_2 p_{i+1,j,k+1}.$$

Again we use the solution $p_i = p_0R^i$ with R as the minimal nonnegative solution of equation $A_0 + RA_1 + R^2A_2 = 0$. Matrices A_0, A_1, A_2 and R are all square matrices of order $(r_1V_1 + 1)(r_2V_2 + 1)$.

A_0 has only nonzero elements on the diagonal below the main diagonal. This row consists of a repetition of r_1V_1 times μ_1 followed by one zero.

A_1 has nonzero elements on three diagonal rows. On the main diagonal the first element is $-(\lambda+\gamma)$, then $r_1(V_1-1)$ times $-(\lambda+\gamma+\mu_1)$ and r_1 times $-(\gamma+\mu_1)$. After this there is a $r_2(V_2 - 1)$ times repetition of one time $-(\lambda + \gamma + \mu_2)$, $r_1(V_1 - 1)$ times $-(\lambda + \gamma + \mu_1 + \mu_2)$ and r_1 times $-(\gamma + \mu_1 + \mu_2)$. And this main diagonal ends with r_2 repetitions of one time $-(\lambda + \mu_2)$, $r_1(V_1 - 1)$ times $-(\lambda+\mu_1+\mu_2)$ and r_1 times $-(\mu_1+\mu_2)$. This gives in total $(r_1V_1+1)(r_2V_2+1)$ elements on the diagonal.

The second nonzero diagonal row in A_1 is r_1 columns to the right and consist of λ on the rows that the main diagonal also consists a λ. So there are $r_1(V_1-1)+1$ times λ followed by r_1 zeros, and this all $(r_2V_2 + 1)$ times, where the last repetition will not have the zeros anymore. The last nonzero diagonal row in this matrix is $r_2(r_1V_1+1)$ columns on the right of the main diagonal and consists of all γ.

Matrix A_2 has only one nonzero diagonal, which is r_1V_1+1 rows below the main diagonal. All those elements are equal to μ_2.

To find R and p_0 the same algorithms are used again. So $R_{k+1} = -(A_0 + R_k^2A_2)A_1^{-1}$, normalising equation $1 = p_0(I - R)^{-1}e$ and the boundary equation $0 = p_0(B_0 + RA_2)$ where B_0 is equal to A_1 without μ_2 in the sums on the diagonal.

The order of all the square matrices is $(r_1 V_1 + 1)(r_2 V_2 + 1)$ which grows fast when the load sizes of the *vehicles* increase or when more *vehicles* are allowed at the terminal. These large dimensions give numerical problems by solving the equations to find R.

Chapter 7

Simulation

To compare the queueing models with a more realistic model, a simulation of a dry bulk terminal is made in Python and in Trafalquar. Python is used to get the simulation close to the assumptions that are made in the queueing models. Trafalquar is the simulation tool of TBA which they use to answer the questions of their customers and which is more extensive than the simulation in Python.

7.1 Python

In Python a simulation is made that models the terminal according to the assumptions of section 3.1. The following sets, variables, parameters and distributions are used in this simulation.

Sets

C	Set of all commodities stored in the terminal.
X_{unload}	Set of all arriving *vehicles* at the unloading station.
X_{load}	Set of all arriving *vehicles* at the loading station.
Y_{unload}	Set of all types of *vehicles* arriving at the unloading station.
Y_{load}	Set of all types of *vehicles* arriving at the loading station.
M_{unload}	Set of all machines in the unloading station.
M_{load}	Set of all machines in the loading station.

Lets define $X = X_{unload} \cup X_{load}$, $Y = Y_{unload} \cup Y_{load}$ and $M = M_{unload} \cup M_{load}$.

Variables

- n_c — Number of units of commodity c in the storage area, for $c \in C$.
- l_x — Remaining units of workload of *vehicle* x, for $x \in X$. 0 if x not at the terminal.
- $h_{x,m}$ — Assignment of the machines to the *vehicles*. 1 if machine m serves *vehicle* x, 0 otherwise, for $(x, m) \in (X_{unload}, M_{unload})$ and $(x, m) \in (X_{load}, M_{load})$.

Parameters

- `capacity` — Maximum number of units of product that can be stored at the terminal.
- `maxUnloading` — Maximum number of *vehicles* at the unloading station.
- `maxLoading` — Maximum number of *vehicles* at the loading station.
- $\texttt{commodity}_x$ — Type of commodity in *vehicle* x, for $x \in X$.
- $\texttt{type}_x$ — Type of vehicle x, for $x \in X$.
- $\texttt{load}_x$ — Load size of *vehicle* x, for $x \in X$.
- $\texttt{maxMachines}_y$ — Maximum number of machines working on a *vehicle* of type y, for $y \in Y$.
- `minStorage` — Fraction of load size of *vehicle* that has to be in the storage area before the *vehicle* can be assigned to a machine in the loading station.
- `initStorage` — Initial number of units of product in the terminal at the start of the simulation.

Distributions

- $A_{y,c}$ — Distribution of inter-arrival times of successive *vehicles* of type y with commodity c, for $y \in Y$ and $c \in C$.
- L_y — Load size distribution of *vehicles* of type y, for $y \in Y$.
- B_{unload} — Handling distribution of a machine in the unloading station handling one unit of product.
- B_{load} — Handling distribution of a machine in the loading station handling one unit of product.

When these parameters and distributions are given as input, the most interesting output of this simulation are the probability distributions of n_c, so $p_i = Pr(\sum_c n_c = i)$ and $p_{c,i} = Pr(n_c = i)$.

This model has a state space with elements of the form $(\mathbf{n}, \mathbf{l}, H_{unload}, H_{load})$, with

$$\begin{aligned}
\mathbf{n} &= (n_c \text{ for } c \in C) \in \mathbb{N}_0^{|C|}, \\
\mathbf{l} &= (l_x \text{ for } x \in X) \in \mathbb{N}_0^{|X|}, \\
H_{unload} &= (h_{x,m} \text{ for } x \in X_{unload}, m \in M_{unload}) \in [0,1]^{|X_{unload}| \times |M_{unload}|}, \\
H_{load} &= (h_{x,m} \text{ for } x \in X_{load}, m \in M_{load}) \in [0,1]^{|X_{load}| \times |M_{load}|},
\end{aligned}$$

where $\mathbb{N}_0$ is the set of natural numbers $[0, 1, 2, \ldots]$ and $|Z|$ is the cardinality of set Z.

Further there are some limitations to this state space.

$$\begin{aligned}
\sum_{c \in C} n_c &\leq \texttt{capacity}, & \\
\sum_{m \in M} h_{x,m} &\leq \texttt{maxMachines}_{type_x}, & \forall x \in X, \\
\sum_{x \in X} h_{x,m} &\leq 1, & \forall m \in M, \\
h_{x,m} &\leq l_x, & \forall x \in X, \forall m \in M, \\
\sum_{x \in X_{unload}} \mathbb{1}\{l_x > 0\}, &\leq \texttt{maxUnloading}, & \\
\sum_{x \in X_{load}} \mathbb{1}\{l_x > 0\}, &\leq \texttt{maxLoading}, &
\end{aligned}$$

with $\mathbb{1}\{Z\}$ as the indicator function. So $\mathbb{1}\{Z\} = 1$ if Z is true and 0 otherwise.

The first limitation says that the total amount of units of product in the storage area can not exceed the `capacity`. The second equation limits the number of machines working on *vehicle* x to the maximum number of machines allowed to work on that type of *vehicle*. The third limitation says that a machine can work at most at one *vehicle* at a time. The next limitation makes sure that a machine is only assigned to *vehicles* with positive workload. If the remaining workload is 0, the *vehicle* is not in the system, so no machine can work on it. The last two limitations define the number of *vehicles* at the terminal, which can not exceed the capacity of the stations.

The state of this system changes over time. For all possible events the following changes of the state occur.

Arriving *vehicle* If *vehicle* x arrives, according to distribution $A_{type_x, commodity_x}$, the remaining workload of this *vehicle* gets the value of the load size of this *vehicle*, according to L_{type_x} ($l_x \sim L_{type_x}$). This is only possible if the *vehicle* queue at the station has not reached its maximum queue length yet.

New assignment of machine to a *vehicle* If some machine is not assigned to any *vehicle* yet (if $\exists m$ with $\sum_x h_{x,m} = 0$) and there are *vehicles* waiting or served without maximum capacity, a new assignment is made. If a *vehicle* is already in service, but not having the maximum number of machines assigned to it yet, then the idle machine is assigned to this *vehicle* (if $\exists x$ with $0 < \sum_{m \in M} h_{x,m} < \texttt{maxMachines}_{type_x}$ then $h_{x,m} = 1$). If all *vehicles* in service already have their maximum number of machines working on it, the idle machine will be assigned to the first *vehicle* in the waiting line that has already enough commodity in the storage area (if $\exists x$ with $l_x > 0$, $\sum_{m \in M} h_{x,m} = 0$ and $\texttt{minStorage} \cdot l_x \leq n_{\texttt{commodity}_x}$, then $h_{x,m} = 1$). The last constraint only applies for *vehicles* at the loading station.

Start of unloading one unit of product If an idle unloading machine m is assigned to a *vehicle* x ($h_{x,m} = 1$), a new unit of product starts to get handled. The end time of unloading this unit is determined according to distribution B_{unload}. This can only be done if the total storage level has not reached the capacity yet ($\sum_c n_c < \texttt{capacity}$).

Start of loading one unit of product If an idle loading machine m is assigned to a *vehicle* x ($h_{x,m} = 1$) and the right commodity is in storage ($n_{commodity_x} > 0$), a new unit of product starts to get handled. The end time of loading this unit is determined according to distribution B_{load}.

End of unloading one unit of product If one of the unloading machines m is done with handling one unit of product, the number of units in the storage yard increases by one ($n_{commodity_x} = n_{commodity_x} + 1$). The remaining workload of the *vehicle* it is handling decreases by one ($l_x = l_x - 1$, for x with $h_{x,m} = 1$).

End of loading one unit of product If one of the loading machines m is done with handling one unit of product, the number of units in the storage yard decreases by one ($n_{commodity_x} = n_{commodity_x} - 1$). The remaining workload of the *vehicle* it is handling decreases by one as well ($l_x = l_x - 1$, for x with $h_{x,m} = 1$).

End of service *vehicle* If a vehicle x is totally (un)loaded, the workload will be equal to zero ($l_x = 0$). This automatically implies that the *vehicle* has left the system. The assignments of the machines which were working on this *vehicle* become 0 ($h_{x,m} = 0, \forall m$). For the other machines that were working on this *vehicle*, their end of handling times will be deleted.

Empty storage If the product in stock of a certain commodity c becomes empty ($n_c = 0$), all loading machines serving vehicles with this commodity stop working. So $\forall m \in M_{load}$ with $h_{x,m} = 1$ for some $x \in X_c$ where X_c is the set of vehicles with commodity c ($\texttt{commodity}_x = c$), the end of handling times of these machines m are deleted.

Full storage If the total number of units of product in the storage yard reaches the maximum capacity ($\sum_c n_c = \texttt{capacity}$), all unloading machines will stop. All end of handling times of the machines in the unloading station will be deleted.

With all these transitions, the simulation can start running over time. All the times in the future when events will happen, are saved. These events are chronologically executed, changing the state space and adding new events to the timeline. The storage levels of the different commodities are saved over time and their movements will tell something about the storage requirement.

7.2 Trafalquar

Trafalquar is the name of the model TBA made for simulating mainly the waterside of a terminal, but it is also used to simulate a dry bulk terminal. This model is made in eM-Plant 7.0.

Trafalquar is much more detailed than the Python simulation. Besides the handling times of the machines in the unloading and loading station, in Trafalquar also all movements between those stations and the storage area are simulated. At the waterside the arriving vessels got a much more detailed planning of where to moor and there are berthing times and times to leave the terminal again.

This all give much more realistic and detailed results than the Python model and therefore we will first compare the results of Trafalquar with the results of Python. Because of the complexity of Trafalquar, this model is slower than the simulation in Python.

Chapter 8

Results of simulation

8.1 Case study

To compare the mathematical results with the simulation of a terminal, a case study is executed. The study is about an export terminal with five commodities. This terminal has a throughput of 5 million ton per year. The first commodity has a throughput of 30 percent of the total throughput per year; 1.5 million ton. The other commodities have respectively 25, 20, 15 and 10 percent of the total throughput.

All those commodities are brought to the terminal with one type of truck. Those trucks all bring a constant load of 30 ton and will always bring only one commodity. The number of trucks that arrive in one year is such that it brings the total throughput. So for the first commodity 50 000 trucks arrive in one year. At the terminal there are three dumpers where the trucks can be unloaded. The trucks will get handled first come, first serve and get unloaded by only one dumper. The dumpers have a mean service time of 3.6 seconds per ton, so they can unload 1 000 ton per hour per dumper.

At the waterside four different type of deepsea vessels arrive; Handysize, Handymax, Panamax and Capesize. Those vessel types have respectively mean load sizes of 20 000, 40 000, 65 000 and 90 000 ton. Every arriving vessel will have a load size according to a triangle distribution with a maximum of 20 percent above or below the mean load size of its corresponding vessel type. Into all vessels only one commodity will be loaded. Of those four types of vessels respectively 27, 31, 26 and 17 vessels arrive per year, so 101 in total. In the simulations an arrival scheme for the vessels is used that tells which day which vessel type and commodity will arrive, this scheme can be found in appendix A. The load sizes of the vessels are not uniformly distributed over the year, see table 8.1. In the fourth and fifth month more vessels are scheduled to ar-

Commodity	jan	feb	mar	apr	may	jun
1	20 000	125 000	255 000	150 000	125 000	125 000
2	125 000	130 000	85 000	190 000	175 000	40 000
3	130 000	20 000	105 000	85 000	40 000	175 000
4	65 000	90 000	20 000	40 000	155 000	40 000
5	0	65 000	0	90 000	20 000	65 000
Total	340 000	430 000	465 000	555 000	515 000	445 000

Commodity	jul	aug	sep	oct	nov	dec
1	150 000	190 000	60 000	60 000	110 000	130 000
2	125 000	40 000	110 000	40 000	150 000	40 000
3	105 000	0	105 000	110 000	40 000	85 000
4	20 000	40 000	65 000	110 000	65 000	40 000
5	0	90 000	40 000	65 000	0	65 000
Total	400 000	360 000	380 000	385 000	365 000	360 000

Table 8.1: Overview of vessel arrivals in ton per commodity

rive than in the rest of the year. The vessels do not exactly arrive on their scheduled day, but can arrive earlier or later. Their delays are according to a normal distribution with a mean of 36 hours late and a standard deviation of 48 hours.

At the terminal there is a berth of 300 meters long. All types of vessels can moor here, but only one at a time. At the quay two cranes will load the vessels, so they are always both working on the same vessel. Those two cranes have both mean service times of 3.6 seconds per ton, so every crane can load 1 000 ton per hour into a vessel.

The distributions of the arrivals of the trucks and the handling times of the truck dumpers and vessel cranes differ in Python and Trafalquar, see appendix B. For the handling times Python uses an exponential distribution with a rate of 1 000 ton per hour, while in Trafalquar every truck and vessel get its own productivity factor. For trucks this is between 0.8 and 1.2 and for vessels between 0.9 and 1.1. This factor is then multiplied with the mean handling time of 3.6 seconds and every ton loaded in this vessel or unloaded from this truck will take this amount of time.

For the inter-arrival times of the trucks Python takes an exponential distribution per commodity. For the first commodity 50 000 trucks have to arrive in a year, or around 5.7 per hour. With this rate the trucks of this commodity arrive to the terminal in a Poisson process. The other commodities have their own rate and arrive all in independent Poisson processes. In Trafalquar the arrivals of trucks have a week and day pattern, for example in the weekends and at night less trucks will arrive than on a Wednesday afternoon. At the beginning of every day in the simulation the mean number of trucks of that day is determined according to the week pattern. Then these trucks get a uniform random arrival time on that day with the given day pattern.

8.2 Experiments

With all this input multiple experiments can be run with the Python and Trafalquar simulation. Because the arrival scheme of the vessels is only for one year, all experiments will be done for one year. Another reason for this is that a real terminal will have control over the in- and outflow and therefore can control the storage level. In the simulations this is not possible, so longer simulations can lead to unrealistic results. For example, when the workload of the arriving vessels is a bit less than the amount of product the trucks deliver, after a couple of years the storage level will be increased this difference in inflow and outflow.

In the mathematical models the stationary probability distribution is given. This will not be attained in simulations of only one year. Therefore multiple experiments are run with different initial storage levels. We will look at a terminal starting with an empty storage area, starting with 100 000 ton in stock and starting with 200 000 ton. So the parameter `initStorage` is set to these values. These amounts are the total initial storage levels which will be equally divided over the five commodities.

In the experiments it is possible that a vessel is at the berth to get loaded, while there is not sufficient stock of its commodity yet to completely load the vessel. Other vessels at the terminal now have to wait until this vessel is loaded. This can cause long waiting times for the other vessels. To prevent this, a rule is made that at least 90 percent of the vessel load has to be in the storage area, before a vessel can go to the quay. So in Python parameter `minStorage` is set to 0.9. In this way the long waiting times of the vessels with enough of their commodity in stock will be reduced.

A last experiment is called 'unlimited'. Here the terminal starts with an initial stock equal to the total throughput per commodity. In this way none of the arriving vessels have to wait on their commodity. At the end of the year the storage levels of the simulation will be lowered such that the minimum storage level per commodity over the year is zero. In this way the value of `minStorage` does not matter, since all commodities are always available and the vessels will be loaded FIFO.

All experiments will have 20 replications of one year with 20 days setup time. At the end of an experiment we are interested in the storage levels and the waiting times of the vessels. We will look at the storage levels that the terminal does not exceed in 90%, 95% and 98% of the time. These limits can be seen as a required upper bound of the storage area. For the vessels we will look at the fraction of vessels that can go to the quay immediately and to the fraction

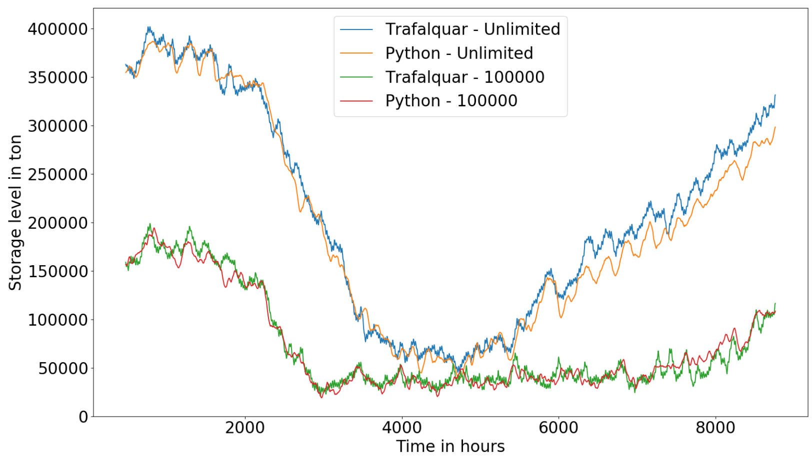

Figure 8.1: Mean storage level over the year for one commodity

of vessels that has to wait more than three days before it gets loaded. For all these results a 95% confidence interval is taken.

All the experiments are done for the terminal with five commodities and for the same terminal where all commodities will be set to the same one. This last case can be considered as a terminal with only one commodity.

8.3 One commodity

For the terminal with one commodity, we first look at the experiment with an initial stock of 100 000 ton. In figure 8.1 the mean storage level over the year is shown and in figure 8.2 the probabilities of a certain waiting time per vessel type can be seen. Both figures show the results of the Trafalquar and the Python simulation.

In this terminal around half of the vessels are loaded immediately and around 20 percent of the vessels have to wait for more than three days before it goes into service. The storage level shows an upper limit around 200 000 ton. In table 8.2 the numerical results are shown for this and the other experiments in Python and Trafalquar.

When the initial stock changes, the limits change with almost the same amount. So the 90% limit is around 70 000 ton above the initial stock, for 95% this is around 90 000 ton and by 98% it is around 108 000 ton more than the initial stock. In figure 8.1 can be seen that the maximum storage level for the experi-

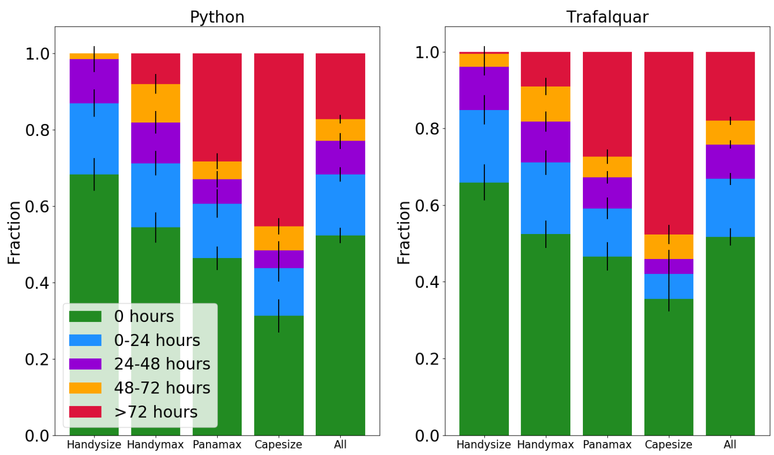

Figure 8.2: Waiting times of vessels at the terminal with one commodity

Initial stock	**Storage level** (×1000 ton)			**Waiting vessels** (%)	
	90 %	95 %	98 %	0 hour	>72 hours
0	83 ± 5	97 ± 7	114 ± 9	42 ± 2	24 ± 1
100 000	167 ± 6	185 ± 6	202 ± 6	52 ± 2	18 ± 2
200 000	272 ± 10	292 ± 10	308 ± 9	65 ± 4	9 ± 2
Unlimited	365 ± 16	383 ± 17	401 ± 18	84 ± 2	0 ± 0

(a) Python

Initial stock	**Storage level** (×1000 ton)			**Waiting vessels** (%)	
	90 %	95 %	98 %	0 hour	>72 hours
0	80 ± 3	93 ± 5	108 ± 7	43 ± 2	25 ± 1
100 000	170 ± 6	189 ± 6	207 ± 7	52 ± 2	18 ± 1
200 000	268 ± 8	289 ± 10	308 ± 10	64 ± 2	9 ± 1
Unlimited	369 ± 12	390 ± 13	409 ± 13	79 ± 2	0 ± 0

(b) Trafalquar

Table 8.2: Results for one commodity

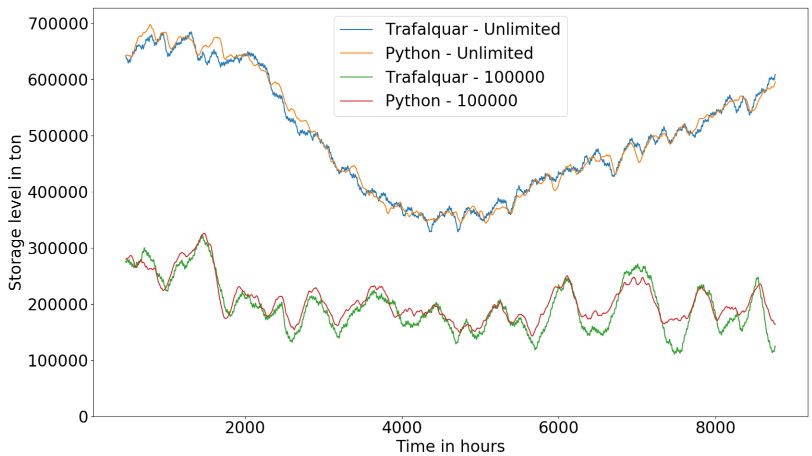

Figure 8.3: Mean storage level over the year for five commodities

ments is reached at the beginning of the year. That could be the reason of the big dependency of the initial stock and the storage level limits. When at the end of the unlimited experiment the actual needed initial stock is determined, the mean is found around 300 000 ton. So for this experiment the same upper limit differences seem to appear.

For the waiting times of the vessels it is clear to see that higher initial stock implies less waiting vessels. The unlimited experiment has the lowest waiting times, with no vessels that have to wait for more than three days. It is not possible to load all vessels immediately, since some vessels arrive before the previous vessel can be completely loaded.

The simulations of Python and Trafalquar give similar results for all experiments.

8.4 Five commodities

Looking at the total terminal with five commodities, again with an initial stock of 100 000 ton in total, the mean storage level over the year in the Python and the Trafalquar experiments can be seen in figure 8.3. The distribution of the waiting times of the vessels is shown in figure 8.4. The initial stock is equally divided over all the commodities, so 20 000 ton per commodity.

The total storage level in this case is larger than for the terminal with only one commodity. The maximum storage level is above 300 000 ton, while for

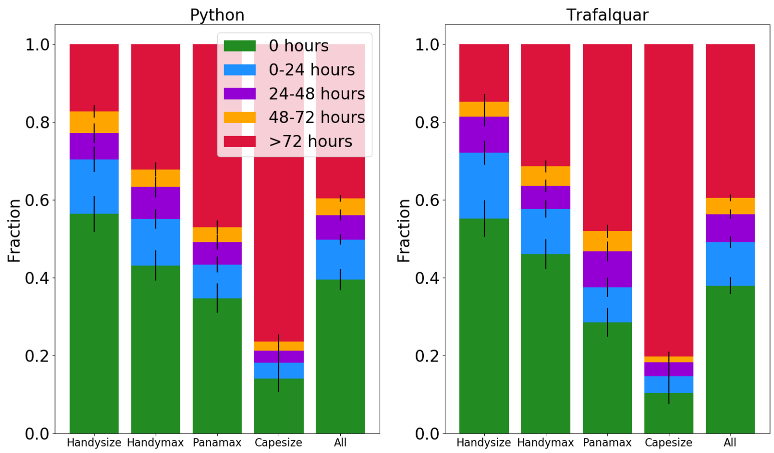

Figure 8.4: Waiting times of vessels at the terminal with five commodities

one commodity this is around 200 000 ton. About 40 percent of the vessels can be loaded immediately, and the same amount of vessels have to wait for more than 72 hours. For a terminal with one commodity this was only 20 percent. Especially the Capesize vessels have to wait more often, because vessels can only go to the quay if 90% of their load is already in stock. For this vessel type this is 81 000 ton, so it is likely that other (smaller) vessels will be loaded first.

In table 8.3 the numerical results of the experiments can be found for the terminal with five commodities. The initial stock is for the total terminal, so every commodity gets a fifth of it.

The storage level limits increase again when the initial stock increases, but this increment is not the same as the increment of the initial stock, like it did for one commodity. For the experiment with no initial stock the 98% limit is around 280 000 ton above this initial stock. For the experiment with 200 000 ton initial stock, the 98% limit is only around 175 000 ton above this starting point.

The waiting times decrease again when the initial stock increases and for the unlimited experiment the waiting times of the vessels are again minimal. The vessels in this experiment are served exactly first in, first out and never have to wait for new products to arrive. The storage level however is a lot higher in this case than in the other experiments, because that is the only way that the vessels can always be served immediately. The initial stock of the terminal in this experiment is different for all five commodities. The means of initial stock are around 140 000, 225 000, 115 000, 75 000 and 25 000 ton. So in total the simulation starts with an initial stock of 580 000 ton, which is already more than the upper limit storage levels of the other experiments. The difference

Initial stock	**Storage level** (×1000 ton)			**Waiting vessels** (%)	
	90 %	95 %	98 %	0 hour	>72 hours
0	241 ± 10	259 ± 11	277 ± 12	29 ± 3	51 ± 3
100 000	275 ± 5	295 ± 6	315 ± 8	39 ± 3	39 ± 3
200 000	326 ± 9	354 ± 8	381 ± 10	47 ± 3	31 ± 3
Unlimited	663 ± 17	684 ± 18	701 ± 18	83 ± 2	0 ± 0

(a) Python

Initial stock	**Storage level** (×1000 ton)			**Waiting vessels** (%)	
	90 %	95 %	98 %	0 hour	>72 hours
0	244 ± 3	265 ± 4	285 ± 5	26 ± 2	52 ± 2
100 000	273 ± 5	296 ± 6	320 ± 9	38 ± 2	40 ± 2
200 000	319 ± 4	347 ± 4	371 ± 5	45 ± 2	32 ± 2
Unlimited	654 ± 10	673 ± 11	691 ± 12	80 ± 2	0 ± 0

(b) Trafalquar

Table 8.3: Results for five commodities

between the initial stock and the 98% limit is around 120 000 ton, which is much less than this difference for the other initial stocks.

Because the experiments of Python and Trafalquar give comparable results and Python is around 6 times faster than Trafalquar, further experiments will only be done in Python.

8.5 Seasonality of truck arrivals

In the case study the arrivals of the trucks are evenly distributed over the year, while the load sizes of the vessels are not, see table 8.1. The previous two subsections show that this induces a large decrease of the storage level in the months with higher demand. It is however very likely that the trucks have a same seasonality in their arrivals as the vessels do. Therefore, a seasonality factor for the trucks is introduced. They still arrive in a Poisson process with averaged over the year the same rate as before, but for every month this rate is multiplied with the seasonality factor of table 8.4. The factors in the row of the total values are used for the experiments with one commodity.

Table 8.5 shows that for one commodity the unlimited experiment requires on average an initial stock less than 100 000 ton, because the storage level limits of this experiment are below the limits of the experiment with 100 000 ton initial stock. Compared to the results of section 8.3 the storage level limits decreased for all experiments and the waiting times of the vessels also became less.

For five commodities figure 8.5 shows that the mean storage level is more stable over the year than without the seasonality, especially for the unlimited experi-

Commodity	jan	feb	mar	apr	may	jun
1	0.16	1	2.04	1.20	1	1
2	1.20	1.25	0.82	1.82	1.68	0.38
3	1.56	0.24	1.26	1.02	0.48	2.1
4	1.04	1.44	0.32	0.64	2.48	0.64
5	0	1.56	0	2.16	0.48	1.56
Total	0.82	1.03	1.12	1.33	1.24	1.07

Commodity	jul	aug	sep	oct	nov	dec
1	1.20	1.52	0.48	0.48	0.88	1.04
2	1.20	0.38	1.06	0.38	1.44	0.38
3	1.26	0	1.26	1.32	0.48	1.02
4	0.32	0.64	1.04	1.76	1.04	0.64
5	0	2.16	0.96	1.56	0	1.65
Total	0.96	0.86	0.91	0.92	0.88	0.86

Table 8.4: Seasonality factors for truck arrivals

ment. The storage level limits of the unlimited experiment are more than 200 000 ton below the limits of section 8.4 and the waiting times decreased as well. This shows that the seasonality of arrivals has a large impact on the storage level limits. The experiment with no initial stock gives worse results than without the seasonality and the 100 000 ton experiment has a bit lower storage level limits and waiting times. The experiment with 200 000 ton initial stock also gives better results with seasonality, where especially the waiting times decreased a lot.

8.6 Other arrival scheme for vessels

Another way to stabilise the storage level over the year is by changing the arrival scheme of the vessels. If the arrival scheme is adjusted such that the arriving workload per month is better divided over the year, see table 8.6, and the trucks arrive again homogeneous over the year, we get the following results.

In table 8.7a the results of one commodity with the new arrival scheme is shown. Compared to the original arrival scheme, this terminal needs less storage capacity. Interesting to see is that the unlimited experiment needs less storage capacity than the experiment that starts with 200 000 ton initial stock. The unlimited experiment starts on average with an initial stock around 150 000 ton, which is below the initial stock of the 200 000 ton experiment. The waiting times of the vessels decrease with this arrival scheme faster when the initial stock increases, than with the original arrival scheme. In figure 8.6 can also be seen that the mean storage level over the year of the unlimited experiment is

Initial stock	**Storage level** (×1000 ton)			**Waiting vessels** (%)	
	90 %	95 %	98 %	0 hour	>72 hours
0	69 ± 2	77 ± 2	85 ± 3	37 ± 4	24 ± 4
100 000	157 ± 10	168 ± 10	182 ± 10	83 ± 1	0 ± 0
200 000	250 ± 13	261 ± 13	274 ± 13	83 ± 1	0 ± 0
Unlimited	141 ± 7	153 ± 7	166 ± 8	85 ± 2	0 ± 0

(a) One commodity

Initial stock	**Storage level** (×1000 ton)			**Waiting vessels** (%)	
	90 %	95 %	98 %	0 hour	>72 hours
0	285 ± 34	343 ± 46	399 ± 57	22 ± 3	59 ± 5
100 000	254 ± 20	278 ± 27	300 ± 36	39 ± 3	36 ± 4
200 000	309 ± 16	324 ± 16	341 ± 19	62 ± 3	15 ± 2
Unlimited	410 ± 9	421 ± 9	434 ± 10	83 ± 2	0 ± 0

(b) Five commodities

Table 8.5: Results for seasonality of truck arrivals

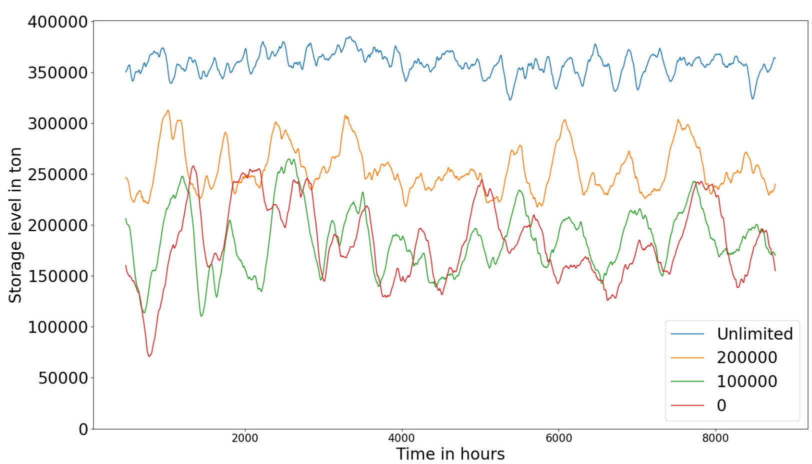

Figure 8.5: Mean storage level over the year for five commodities with seasonality of truck arrivals

Commodity	jan	feb	mar	apr	may	jun
1	150 000	125 000	125 000	130 000	85 000	125 000
2	125 000	90 000	85 000	125 000	110 000	60 000
3	90 000	60 000	85 000	105 000	65 000	85 000
4	65 000	90 000	65 000	60 000	90 000	60 000
5	0	20 000	90 000	0	65 000	65 000
Total	430 000	385 000	450 000	420 000	415 000	395 000
Commodity	**jul**	**aug**	**sep**	**oct**	**nov**	**dec**
1	150 000	125 000	125 000	60 000	150 000	150 000
2	125 000	80 000	110 000	105 000	105 000	130 000
3	80 000	90 000	105 000	90 000	60 000	85 000
4	20 000	40 000	65 000	90 000	65 000	40 000
5	0	90 000	40 000	65 000	0	65 000
Total	375 000	425 000	445 000	410 000	380 000	470 000

Table 8.6: Overview of vessel arrivals in ton per commodity of new arrival scheme

Initial stock	**Storage level** ($\times$1000 ton)			**Waiting vessels** (%)	
	90 %	95 %	98 %	0 hour	>72 hours
0	77 ± 10	86 ± 11	94 ± 12	39 ± 4	24 ± 3
100 000	126 ± 12	140 ± 12	155 ± 12	70 ± 6	6 ± 3
200 000	244 ± 15	259 ± 16	273 ± 16	83 ± 1	0 ± 0
Unlimited	172 ± 12	185 ± 12	200 ± 13	85 ± 2	0 ± 0

(a) One commodity

Initial stock	**Storage level** ($\times$1000 ton)			**Waiting vessels** (%)	
	90 %	95 %	98 %	0 hour	>72 hours
0	256 ± 13	276 ± 14	298 ± 16	32 ± 3	48 ± 3
100 000	286 ± 15	306 ± 15	327 ± 14	42 ± 3	34 ± 3
200 000	324 ± 18	340 ± 19	359 ± 19	55 ± 4	22 ± 3
Unlimited	423 ± 14	436 ± 14	452 ± 16	83 ± 2	0 ± 0

(b) Five commodities

Table 8.7: Results with another arrival scheme for vessels

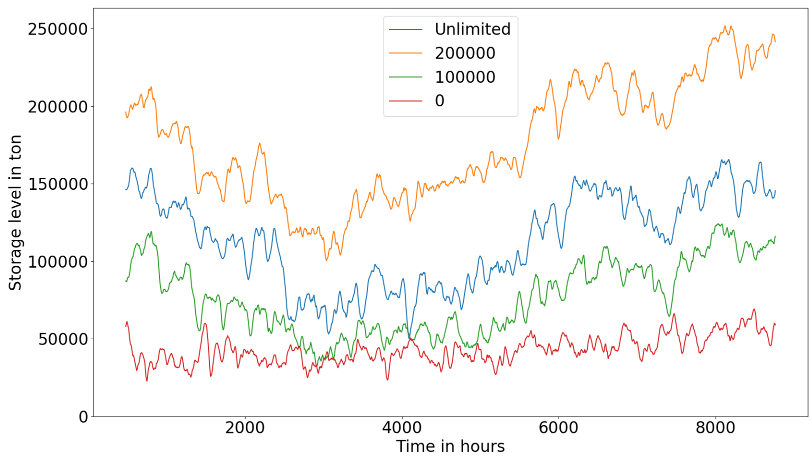

Figure 8.6: Mean storage level over the year for one commodity with another arrival scheme for vessels

below the 200 000 ton experiment and that the storage level is more stable over the year than with the original arrival scheme.

For five commodities the results are shown in table 8.7b. For low initial stocks the storage level limits increased a bit, but for the 200 000 ton and the unlimited experiment it decreased. The waiting times however decreased for all the experiments with this new arrival scheme.

For all experiments the confidence intervals increased. A reason for this could be that the peak of the year is not at the beginning of the year anymore, but more to the end. And points further in time will have larger variations than points at the beginning of the timeline.

8.7 Larger arrival delays

Another way to change the arrival pattern of the vessels is by changing the distribution of the delay times of the arriving vessels. In the original model the vessels arrive before or after the scheduled time with a normal distribution with a mean of 36 hours and standard deviation of 48 hours. In this way all vessels arrive between 3 days before or 7 days after. When the parameters of this distribution is changed to a mean of 48 hours and standard deviation of 72 hours, all vessels will arrive between 5 days before or 10 days after the scheduled time. This change gives the results of table 8.8. Because of the increased variation the experiments are run with 30 replications instead of 20.

Initial stock	Storage level (×1000 ton)			Waiting vessels (%)	
	90 %	95 %	98 %	0 hour	>72 hours
0	88 ± 7	106 ± 9	127 ± 10	43 ± 2	23 ± 2
100 000	182 ± 9	205 ± 8	226 ± 9	51 ± 2	17 ± 2
200 000	284 ± 8	307 ± 8	331 ± 9	65 ± 3	8 ± 2
Unlimited	373 ± 11	396 ± 11	416 ± 12	81 ± 1	0 ± 0

(a) One commodity

Initial stock	Storage level (×1000 ton)			Waiting vessels (%)	
	90 %	95 %	98 %	0 hour	>72 hours
0	246 ± 8	265 ± 8	286 ± 9	29 ± 2	50 ± 2
100 000	288 ± 11	313 ± 11	338 ± 13	39 ± 2	38 ± 2
200 000	345 ± 13	371 ± 13	398 ± 13	47 ± 2	29 ± 2
Unlimited	664 ± 13	689 ± 14	709 ± 14	80 ± 1	0 ± 0

(b) Five commodities

Table 8.8: Results for larger arrival delays

For one commodity the results of table 8.8a differs a maximum of 25 000 ton with the original model and the fractions of waiting vessels are also similar. Table 8.8b shows that the results for five commodities also give similar results as the original model. So this change of the delay distribution does not seem to have a lot of impact on the storage requirement, but the little changes are not advantageous.

8.8 Other values of `minStorage`

In all the previous experiments a vessel only got accepted to go to the quay when at least 90 percent of its load is in the storage area. This ensures that the quay cranes do not get blocked by a vessel that has to wait on a large amount of product that is not available yet, but that vessels with other commodities will be loaded first. To show this effect the experiment with 100 000 ton initial stock is run with other values of parameter `minStorage`.

For a terminal with one commodity table 8.9a shows that the storage level limits do not depend a lot on this parameter, while the waiting times increase when the threshold percentage decreases. The reason of the small change in this system is that all vessels want the same commodity. So the only change in loading order can be that smaller vessels will be loaded before a larger vessel. Looking at the productivity of the vessels, so the mean amount of product that is loaded into the vessel per hour, the increased waiting times are also visible. For the 90% experiment the productivity is on average 1 999 ton per hour, and with the two cranes with loading speeds of 1 000 per hour, this productivity is on its maximum. For the 50% and 0% the productivity drops to 1 606 and

`minStorage`	**Storage level** (×1000 ton)			**Waiting vessels** (%)	
	90 %	95 %	98 %	0 hour	>72 hours
90 %	167 ± 6	185 ± 6	202 ± 6	52 ± 2	18 ± 2
50 %	164 ± 8	186 ± 7	205 ± 6	45 ± 4	21 ± 3
0 %	170 ± 8	190 ± 7	207 ± 7	47 ± 5	34 ± 6

(a) One commodity

`minStorage`	**Storage level** (×1000 ton)			**Waiting vessels** (%)	
	90 %	95 %	98 %	0 hour	>72 hours
90 %	275 ± 5	295 ± 6	315 ± 8	39 ± 3	39 ± 3
50 %	309 ± 11	340 ± 12	371 ± 13	8 ± 2	80 ± 3
0 %	586 ± 35	655 ± 37	699 ± 39	2 ± 1	94 ± 2

(b) Five commodities

Table 8.9: Results for other values of `minStorage`

1 257 ton per hour respectively. So the mean time a vessel will be at the quay increased, and therefore the waiting times of the other vessels at the terminal also increase.

For the terminal with five commodities the impact is bigger, see table 8.9b. When the percentage is set to zero, the vessels will be served first in, first out and almost all vessels have to wait for more than three days. Figure 8.7 shows a large peak in the middle of the year for this experiment, because some vessel with an unavailable commodity demand is blocking the cranes and all other vessels. The productivity of the vessels in the 90% experiment is on average 1 765 ton per hour, while for the other two experiments this is 1 399 and 1 485 ton per hour. This last productivity is better than for the one commodity experiment and 50% experiment. A reason for this could be that in this experiment one vessel will block the system for a long time and gets a low productivity, but in the meantime the other commodities have increased in stock level, such that the next couple of vessels can be loaded relatively fast with a high productivity.

8.9 Exponential arrivals of vessels

In the mathematical models the assumption is made that the *vehicles* at the loading station arrive in a Poisson process. This process has much more variation than the scheduled arrivals of the case study. In Python this arrival process can be changed to a Poisson process. The order of the vessels with its commodities is the same as in the arrival schedule, but now the time between two arrivals is exponentially distributed with a mean of 101 vessels per year. The same experiments are run, but now with 50 replications, because of the increased variation. The results of these experiments are shown in table 8.10.

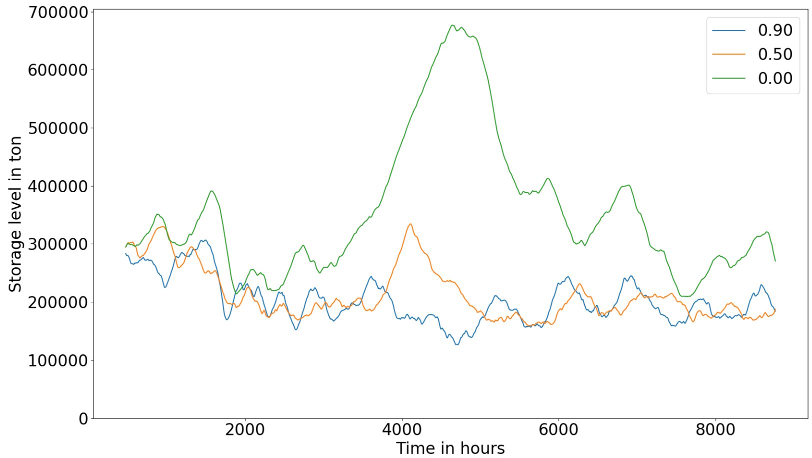

Figure 8.7: Mean storage level over the year for five commodities for different threshold percentages

Initial stock	**Storage level** (×1000 ton)			**Waiting vessels** (%)	
	90 %	95 %	98 %	0 hour	>72 hours
0	229 ± 49	261 ± 51	291 ± 53	46 ± 4	22 ± 4
100 000	294 ± 64	331 ± 63	361 ± 63	51 ± 5	17 ± 4
200 000	427 ± 61	473 ± 61	506 ± 61	58 ± 4	11 ± 3
Unlimited	584 ± 42	614 ± 41	643 ± 38	70 ± 1	0 ± 0

(a) One commodity

Initial stock	**Storage level** (×1000 ton)			**Waiting vessels** (%)	
	90 %	95 %	98 %	0 hour	>72 hours
0	321 ± 37	350 ± 40	377 ± 41	33 ± 4	46 ± 5
100 000	385 ± 42	420 ± 46	447 ± 47	40 ± 4	36 ± 4
200 000	443 ± 48	481 ± 49	515 ± 49	46 ± 4	30 ± 5
Unlimited	768 ± 42	803 ± 43	832 ± 43	71 ± 2	0 ± 0

(b) Five commodities

Table 8.10: Results for exponential arrivals of vessels

Initial stock	**Storage level** (×1000 ton)			**Waiting vessels** (%)	
	90 %	95 %	98 %	0 hour	>72 hours
0	288 ± 74	319 ± 78	342 ± 80	37 ± 8	27 ± 6
100 000	379 ± 76	416 ± 74	443 ± 74	48 ± 8	18 ± 5
200 000	403 ± 73	442 ± 69	472 ± 66	49 ± 7	17 ± 5
Unlimited	608 ± 44	648 ± 42	672 ± 39	72 ± 2	0 ± 0

(a) One commodity

Initial stock	**Storage level** (×1000 ton)			**Waiting vessels** (%)	
	90 %	95 %	98 %	0 hour	>72 hours
0	676 ± 76	708 ± 77	731 ± 76	41 ± 4	37 ± 5
100 000	778 ± 69	813 ± 71	840 ± 71	49 ± 4	28 ± 4
200 000	772 ± 61	807 ± 61	834 ± 60	48 ± 3	28 ± 4
Unlimited	1 071 ± 37	1 095 ± 34	1 116 ± 32	72 ± 2	0 ± 0

(b) Five commodities

Table 8.11: Results for exponential arrivals of one type of vessel

The storage level limits of the experiments with one commodity are almost twice the limits of the experiments with the scheduled arrivals of vessels, while the waiting times of the vessels are similar. For five commodities the differences of the storage level limits are not that big, but still above the original model. The waiting times of the vessels are here less than before.

The unlimited experiment again has the highest storage levels, but also can handle the highest percentage of vessels immediately. The storage level limits are even higher than with the scheduled vessel arrivals and the waiting times are a bit longer, since in a Poisson process two successive vessels can arrive in a relative short time, while for the scheduled arrivals this probability is smaller.

8.10 One vessel type

Besides the exponential arrivals, the queueing models also assume only one type of vessels to arrive. To see if this has impact on the results of the simulation, the experiments are also done for a terminal with only one vessel type. This vessel type has a mean load size of $\frac{5000000}{101} \approx 49\,505$ ton. The arrivals will again be in a Poisson process with as rate the number of vessels divided by the hours per year. For the case with one commodity this is $\frac{101}{365 \cdot 24}$. For the experiments with five commodities, every commodity has its own independent arrival process with the same ratio's as for the total throughput are used, so the vessels of the first commodity have arrival rate $0.3 \cdot \frac{101}{365 \cdot 24}$.

In table 8.11 the results of these experiments can be found. The difference between no initial stock and 100 000 ton initial stock is around 100 000 ton, while the difference between 100 000 and 200 000 ton initial stock is much smaller. The waiting times of the vessels also do not differ a lot. The unlimited experiments need again a much bigger storage area, but then the waiting times drop to their minimum.

Compared with the four types of vessels arriving in a Poisson process, the storage level limits are in this section higher. This difference is even bigger for the five commodity terminal, than for only one commodity. A reason for this could be that the Poisson processes for the vessels lead to less than 101 vessel arrivals per year. In the previous section there is only one Poisson process of arrivals, while here every commodity gets its own process. When all those five commodities have 2 vessel arrivals in a year less than the schedule, the simulation misses 10 vessels of around 50 000 ton at the end of the year. This induces that the storage area has at least 500 000 ton in stock. This large values of the storage level at the end of the year causes increments of the storage level limits.

8.11 Geometric vessel load sizes

In the queueing models of sections 5.5 and 6.1, a vessel leaves after every end of loading one unit of product with probability q. This could be seen as vessel load sizes according to a geometric distribution, since after every end of loading there is a 'succes' with probability q, or a stay with probability $(1-q)$. In figure 8.8 the previous used triangular distribution of the vessel load is shown together with the geometric distribution. Both have a mean load size of $\frac{5000000}{101} \approx 49\,505$ ton and the triangular distribution has a 20 percent margin above and below this mean. In this figure can be seen that the variation of the geometric distribution is much bigger than of the triangular distribution.

To see the impact of this variation, we take a model with one type of vessels with a geometric load size distribution arriving in a Poisson process. All other parameters are the same as before. The Python simulation gives then the storage level limits shown in table 8.12. The confidence intervals are wider, because the variation of the load size vessels is larger here, but also the mean values of the limits are higher than for the same system with triangular load size distribution (section 8.10). The initial stock has for five commodities less impact on the storage level limits, than in the case of one commodity. The waiting times are lower than before, which could be because of the increased probability of a vessel with a small load size.

The assumption of these geometric vessel load sizes is not realistic, because of the large variability. Since the simulation already gives higher storage level limits, it can be expected that the queueing models with this assumption also give higher results than in a real terminal would be needed.

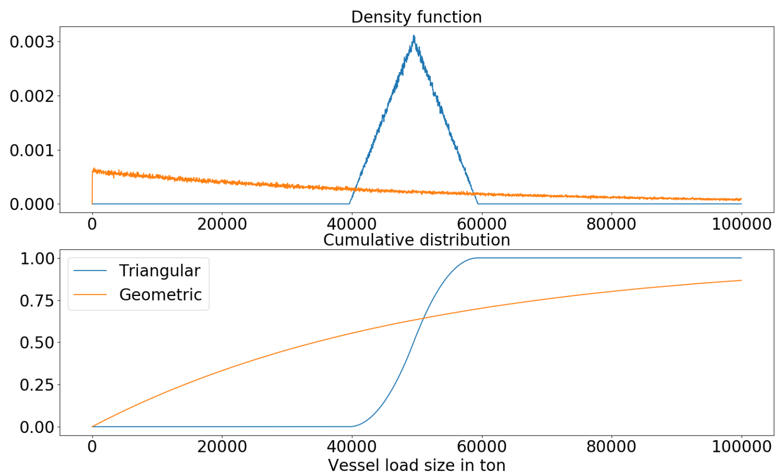

Figure 8.8: Load size distributions of vessels

Initial stock	**Storage level** (×1000 ton)			**Waiting vessels** (%)	
	90 %	95 %	98 %	0 hour	>72 hours
0	411 ± 84	453 ± 86	490 ± 86	62 ± 3	13 ± 3
100 000	533 ± 99	578 ± 100	615 ± 101	65 ± 3	10 ± 2
200 000	600 ± 98	651 ± 99	687 ± 98	66 ± 3	9 ± 2
Unlimited	845 ± 70	890 ± 67	922 ± 64	72 ± 2	2 ± 0

(a) One commodity

Initial stock	**Storage level** (×1000 ton)			**Waiting vessels** (%)	
	90 %	95 %	98 %	0 hour	>72 hours
0	960 ± 88	1 006 ± 88	1 046 ± 87	57 ± 3	24 ± 3
100 000	991 ± 96	1 042 ± 96	1 080 ± 96	58 ± 3	20 ± 3
200 000	1 019 ± 92	1 064 ± 92	1 100 ± 91	59 ± 3	20 ± 3
Unlimited	1 474 ± 64	1 526 ± 62	1 554 ± 59	70 ± 2	4 ± 2

(b) Five commodities

Table 8.12: Results for exponential arrivals of one type of vessel with geometric load size distribution

Initial stock	**Storage level** (×1000 ton)			**Waiting vessels** (%)	
	90 %	95 %	98 %	0 hour	>72 hours
0	197 ± 47	228 ± 50	259 ± 53	54 ± 2	18 ± 2
100 000	240 ± 64	279 ± 67	309 ± 69	55 ± 2	18 ± 2
200 000	261 ± 48	305 ± 53	342 ± 56	56 ± 2	17 ± 2
Unlimited	974 ± 75	1 006 ± 65	1 033 ± 58	66 ± 2	4 ± 0

(a) One commodity

Initial stock	**Storage level** (×1000 ton)			**Waiting vessels** (%)	
	90 %	95 %	98 %	0 hour	>72 hours
0	941 ± 77	1 002 ± 81	1 049 ± 82	59 ± 2	24 ± 2
100 000	901 ± 64	958 ± 66	1 005 ± 66	59 ± 2	23 ± 2
200 000	888 ± 73	937 ± 75	976 ± 75	59 ± 2	22 ± 2
Unlimited	1 576 ± 64	1 613 ± 62	1 638 ± 58	67 ± 2	5 ± 0

(b) Five commodities

Table 8.13: Results for a terminal with a finite waiting queue

8.12 Finite waiting area for vessels

In some of the queueing models the waiting space for the vessels is limited. When we set this value to a maximum of 5 vessels, some arriving vessels will not go into the terminal. Therefore the number of arriving vessels will be increased by 20 percent, so instead of 101 vessels per year, now 121 vessels will come to the terminal. Again we will assume that the vessels have geometric load sizes with a mean of 49 505 ton and arrive in a Poisson process per commodity.

For one commodity table 8.13a shows that the initial stock in these experiments does not have a lot of impact on the storage level limits and the waiting times of the vessels. The storage level stays around a mean value and does not increase or decrease a lot over the year. The unlimited experiment with this input actually does not say anything, because too many vessels arrive at the terminal. There is unlimited stock so all 121 vessels will go into the terminal, which will pick up $20 \cdot 49\,505 = 990\,100$ ton more products than the trucks deliver. So the storage level in this experiment only decreases over the year and the storage level peaks are at the start of the year.

For five commodities, table 8.13b, the unlimited experiment does the same. For the other three cases the storage level increases over the year. Only around 93-99 vessels will be accepted to the terminal per year, which do not have the capacity of the 5 million ton that the trucks deliver. So at the end of the year there is a surplus of product and the storage level limits are touched here. When at the start of the year the initial storage level is higher, more vessels will be accepted, so less surplus at the end, which induces lower storage level limits. The waiting times do not differ a lot between the experiments.

Compared to the same simulations with unlimited waiting space for the vessels (section 8.11) are the storage level limits in this section lower. Presumably because in this section more vessels arrive per year to lower the storage level again. This also explains the longer waiting times.

Chapter 9

Results of queueing models

To compare the results of the simulations with the results of the queueing models, the input of the case study first has to be adjusted to the needed parameters. Since all trucks bring the same amount of product, 30 ton, most sections use as unit of product a truckload instead of a ton. The throughput of the terminal is 5 000 000 ton per year, which are 166 667 trucks.

In all the models of the loading station, we assumed Poisson arrivals of the units of product into the storage area. The rate of these arrivals is λ per hour. In the case study this λ is equal to the 16 667 trucks divided by $365 \cdot 24$ hours a year, so $\lambda = \frac{5000000}{30 \cdot 365 \cdot 24} \approx 19.03$.

In the loading station there are two cranes which both can load 1 000 ton per hour into the vessels. By assumption these cranes have exponential service times per truckload. The rate of these cranes is $\mu_2 = \frac{1000}{30} \approx 33.33$ truckloads per hour. Most models have only one machine instead of two, therefore we adjust μ_2 to $2 \cdot \frac{1000}{30} \approx 66.67$.

All queueing models only work if the occupation rate of the system $\rho = \frac{\lambda}{s \cdot \mu}$ is at least less than one. In this study the terminal satisfies this condition since $\rho \approx \frac{19.03}{2 \cdot 33.33} \approx 0.285 < 1$.

The last important parameter is γ, the arrival rate of the vessels. In the case study 101 vessels arrive per year. This gives $\gamma = \frac{101}{365 \cdot 24} \approx 0.012$. Again we assume that those vessels arrive in a Poisson process.

In the mentioned models it is easier to assume that all vessels have the same load size distribution. Therefore we take as mean load size of the arriving vessels the total throughput divided by the number of arriving vessels. So $r = \frac{5000000}{101} \approx 49\,505$ ton $\approx 1\,650$ truckloads per vessel.

With all these parameters, the models given in chapters 5 and 6 about the probability distribution of the storage level can be evaluated. Because all the queueing models assume only one commodity, the results of this chapter will be compared with the simulations of one commodity. First all models will be explained with the input of the case study and some results with different values of the parameters are given. In the end of this chapter a summary of the results is given and compared with the results of the simulations.

9.1 $M|M|2$ queue

In the introduction of chapter 5 the standard $M|M|s_2$ queue, in this case an $M|M|2$ queue, is introduced. Equation 5.1 gives the steady state probabilities for the storage level as

$$\begin{aligned} p_0 &= \left(1 + 2\rho + \frac{2\rho^2}{1-\rho}\right)^{-1} \approx 0.56, \\ p_n &= 2\rho^n p_0, \qquad \text{for } n \geq 1. \end{aligned}$$

In this model 96 percent of the time the storage level will be below two truckloads and 98 percent of the time below three truckloads. This is far too little to really fulfill the needs of the terminal. The main reason for this is that this model assumes constant work for both the machines, while in the terminal the machines only work when there is a vessel at the quay.

One way so adjust this system to a more realistic case, is to lower μ_2 to the rate of the 5 000 000 ton per year. By the current speed of 1 000 ton per hour, only 5 000 machine hours are needed to load all the throughput into the vessels. With two machines, this is only 2 500 hours per machine, which is $\frac{2500}{365 \cdot 24} \approx 28.5\%$ of the time. When μ_2 is therefore lowered to $1\,000 \times 0.285 \approx 285$ ton or 9.5 truckloads per hour, the new ρ will be $\frac{19.03}{2 \times 9.5} = 1$, which gives an infeasible system. This is logical since we set the inflow of trucks equal to the handling time of the cranes.

When μ_2 is just adjusted to 10 trucks per hour, so a bit of overcapacity of the cranes, ρ will become $\frac{19.03}{2 \times 10} \approx 0.95$ and p_0 will be approximately 0.025. Now the storage level limits of 90, 95 and 98 percent are respectively 46, 60 and 78 truckloads. So 1 380, 1 800 and 2 340 ton, which is still far too small.

9.2 $M^G|M|1$ queue

The next model in chapter 5 is the $M^G|M|1$ queue (section 5.1). Here we will not assume truckloads as unit of product, but tons. Therefore μ_2 will now have value 2 000 ton per hour, since the two cranes will be seen as only one crane.

ρ is now $\frac{\lambda}{\mu_2} \approx 0.01$. The trucks have a constant load size of 30 ton, so in this model $p_0 = 1 - 30\rho \approx 0.71$.

Because of this constant load size of the trucks, the balance equations of this system can be simplified to

$$\begin{aligned} p_1 &= \rho p_0, \\ p_n &= (1+\rho)p_{n-1} = \rho(1+\rho)^{n-1}p_0, && \text{for } 1 < n \leq 30, \\ p_{n+1} &= (1+\rho)p_n - \rho p_{n-30}, && \text{for } n \geq 30. \end{aligned}$$

These can be solved in two ways; iteratively or by taking $p_n = x^n p_0$ and solving $0 = x^{31} - (1+\rho)x^{30} + \rho$, which has one real solution less than one.

With the ρ of the case study the 90, 95 and 98 percent limits of the storage level are at 25, 31 and 47 ton, which are in the same order as the $M|M|2$ queue. Again we have the assumption that the cranes always work instead of only when there are vessels at the quay. By again adjusting μ_2 to 20 truckloads or 600 ton per hour, ρ becomes approximately 0.03 and $p_0 \approx 0.05$. The storage level limits are now 710, 927 and 1 214 ton which are even lower than results of the $M|M|2$ queue model. The assumptions of these two models are to strict to determine a realistic storage level.

9.3 Virtual storage level

The next model (section 5.3) is a model that looks at the virtual storage level, so with instant loading into the vessels. In this model we assume there is a maximum number of vessels that can be at the terminal, V, and that the vessels will bring more workload than the trucks do. So $\lambda < \gamma \cdot r$. Therefore we will increase the value of γ by 10 percent and limit the number of vessels at the terminal to $V = 3$.

Solving equation 5.3 with $\lambda \approx 19.03$, $\gamma \approx 0.012 * 1.1 \approx 0.013$ and $r \approx 1\,650$, we find $\bar{x} \approx 0.99988$. Now with equation 5.4 the storage level limits are found at 15 530, 21 443 and 29 259 truckloads, so 465 900, 643 290 and 877 770 ton.

In table 9.1 results for other values of γ and V can be found. As you can see, the difference of the storage levels when V changes is exactly this change times $r \approx 49\,500$ ton, because the process is only shifted that amount to the left or right and the parameters of λ, γ and r, which determine $\bar{x}$ stay constant. When the number of vessels that arrive increases, the storage level decreases because the outflow will be larger.

Parameters		**Storage level** (×1000 ton)		
γ (%)	V	90 %	95 %	98 %
105	3	1 038	1 388	1 850
105	5	939	1 289	1 751
110	3	466	643	878
110	5	367	544	779
120	5	81	172	293

Table 9.1: Results for model of virtual storage level

9.4 Breakdowns

For the system with breakdowns (section 5.4), we need four parameters; λ, μ_2, γ and θ. The first three are already known; 19.03, 66.67 and 0.012. The last parameter, θ, is the rate in which the system goes from working to not-working. The system is only stable if $\frac{\lambda}{\mu_2} < \frac{\gamma}{\theta+\gamma}$, so only if $\theta < \frac{\gamma(\mu_2-\lambda)}{\lambda} \approx 0.029$. The mean time the system is continuously working has therefore to be larger than $\frac{1}{\theta} \approx 35$ hours.

The mean load size of a vessel is about 49 505 ton, the cranes will need approximately $\frac{r}{\mu_2} \approx 24.75$ hours per vessel. With equation 5.6 the value of θ becomes

$$\frac{1}{\theta} \approx \sum_{i=1}^{\infty} \frac{i^{i-1}}{i-1!} \cdot 24.75^i \cdot 0.012^{i-1} \cdot \exp(-i \cdot 24.75 \cdot 0.012) \approx 34.6.$$

This would give $\theta \approx 0.029$, which does not satisfy the stability condition since now $\frac{\lambda}{\mu_2} = \frac{\gamma}{\theta+\gamma}$. Therefore we will increase the mean loading time per vessel, T, with 10 percent to 27 hours, which gives θ approximately as 0.025.

With these parameters, the matrix R becomes

$$\begin{bmatrix} \frac{\lambda(\theta+\mu_2)}{\mu_2(\lambda+\gamma)} & \frac{\lambda}{\mu_2} \\ \frac{\lambda\theta}{\mu_2(\lambda+\gamma)} & \frac{\lambda}{\mu_2} \end{bmatrix} \approx \begin{bmatrix} 0.9998 & 0.2854 \\ 0.0004 & 0.2854 \end{bmatrix}.$$

With equation 5.5 the probability of a storage level of $n > 0$ truckloads is

$$\begin{aligned} p_n &= \frac{\lambda}{\mu_2(\lambda+\gamma)} \begin{bmatrix} \frac{(\theta+\mu_2)(\mu_2\gamma-\lambda\theta-\lambda\gamma)}{\mu_2(\lambda+\gamma)} & \frac{\mu_2\gamma-\lambda\theta-\lambda\gamma}{\mu_2} \end{bmatrix} \begin{bmatrix} \frac{\lambda(\theta+\mu_2)}{\mu_2(\lambda+\gamma)} & \frac{\lambda}{\mu_2} \\ \frac{\lambda\theta}{\mu_2(\lambda+\gamma)} & \frac{\lambda}{\mu_2} \end{bmatrix}^{n-1} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ &\approx \begin{bmatrix} 0.000055 & 0.000016 \end{bmatrix} \begin{bmatrix} 0.9998 & 0.2854 \\ 0.0004 & 0.2854 \end{bmatrix}^{n-1} \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \end{aligned}$$

Together with the probability of an empty storage $p_0 = 1 - \frac{\lambda(\gamma+\theta+\mu_2)}{\mu_2(\lambda+\gamma)} \approx 0.000055$, the storage level limits can be found at 896 730, 1 116 670 and 1 523 520 ton.

Parameters		Storage level (×1000 ton)			Working fraction
T	θ	90 %	95 %	98 %	$\frac{\gamma}{\gamma+\theta}$
110%	0.025	897	1 117	1 524	0.31
120%	0.022	489	636	831	0.34
130%	0.020	353	460	600	0.37

Table 9.2: Results for model with breakdowns

In table 9.2 the storage limits are shown for different values of T, which induces different values of θ. If θ becomes smaller, the fraction of time the cranes are working increases, so the needed storage capacity decreases. In the last column the fraction of time the machines are working is shown.

9.5 Finite queue of vessels

The last mentioned model in chapter 5 has a finite queue of vessels (section 5.5). It has a lot of similarities with the model with breakdowns, but now the matrices have larger dimensions.

The same values for the parameters are used again, so $\lambda \approx 19.03$, $\mu_2 \approx 66.67$ and $\gamma \approx 0.012$. But since the number of vessels at the terminal is limited, γ will be increased by 40 percent. The last parameter in this model is q, the probability that a vessel leaves after handling one unit of product. This probability is the inverse of the load size of a vessel, so $q = \frac{1}{r} \approx 0.0006$.

Lets start with a maximum number of vessels at the terminal of two, so the matrices of section 5.5 have dimension 3×3.

$$A_0 = \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix}, \qquad A_1 = \begin{bmatrix} -(\lambda+\gamma) & \gamma & 0 \\ 0 & -(\lambda+\gamma+\mu_2) & \gamma \\ 0 & 0 & -(\lambda+\mu_2) \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 0 & 0 & 0 \\ q\mu_2 & (1-q)\mu_2 & 0 \\ 0 & q\mu_2 & (1-q)\mu_2 \end{bmatrix}, \qquad B_0 = \begin{bmatrix} -(\lambda+\gamma) & \gamma & 0 \\ 0 & -(\lambda+\gamma) & \gamma \\ 0 & 0 & -\lambda \end{bmatrix}.$$

Now the matrix R can be found, by iteratively solving $R_{k+1} = -(A_0 + R_k^2 A_2)A_1^{-1}$ until R_k is equal to R_{k+1}, starting with $R_0 = 0$. With this R the null space of $B_0 + RA_2$ can be found to determine p_0 together with normalisation equation $1 = p_0(I-R)^{-1}e$.

With the mentioned values of the parameters the matrix R and vector p_0 become

$$R \approx \begin{bmatrix} 1.00 & 0.22 & 0.07 \\ 0.00 & 0.29 & 0.00 \\ 0.00 & 0.00 & 0.29 \end{bmatrix}, \qquad p_0 \approx \begin{bmatrix} 0.00 & 0.11 & 0.17 \end{bmatrix}.$$

Parameters			**Storage level** (×1000 ton)			**Number of vessels** (%)					
γ (%)	$\frac{1}{q}$	V	90 %	95 %	98 %	0	1	2	3	4	5
140	1 650	2	262	363	496	43	28	29			
140	1 650	5	90	176	290	14	10	11	15	21	29
120	1 650	5	362	535	762	30	15	12	12	14	17
120	2 000	5	72	164	285	11	9	11	15	22	32
100	2 000	5	401	598	858	28	14	12	13	15	18
100	2 000	2	851	1 135	1 511	57	26	18			

Table 9.3: Results for model with finite queue of vessels

With $p_i = p_0 R^i$ all probabilities are calculated and the storage level limits are determined. For this system these limits are around 262 000, 363 000 and 496 000 ton. With this input the system has 43 percent of the time no vessels at the terminal, 28 percent one vessel and 29 percent of the time two vessels.

In table 9.3 results can be found for other values of the parameters γ, q and V. When the maximum number of vessels decreases, more storage area is needed, since the probability of no vessels becomes larger. When the arriving rate of the vessels decreases, also more storage area is needed, because it takes a longer time before a new vessel arrives. When the vessels become larger, so q smaller, the outflow becomes bigger, so the storage level limits decrease.

9.6 Total system with geometric load sizes

In chapter 6 two models are explained for a combination of the unloading and loading station. In these models the queues of *vehicles* at the unloading and loading station are limited. A maximum of V_1 trucks and V_2 vessels can be simultaneously at the terminal. First we will look at the model with the geometric load size distributions (section 6.1).

At the unloading station the trucks arrive with $\lambda \approx 19.03$ with a load size of 30 ton, so $q_1 = \frac{1}{30}$. In this station three machines are handling 1 000 ton per hour, for the model we take one machine with $\mu_1 = 3\,000$. At the loading station vessels arrive with $\gamma \approx 0.012$ with mean load size of 49 505 ton, so $q_2 \approx \frac{1}{49505} \approx 2 \cdot 10^{-5}$. In this station two cranes are handling 1 000 ton per hour, so $\mu_2 = 2\,000$.

For the total system, again the matrix R has to be found. The order of this square matrix is $(V_1 + 1)(V_2 + 1)$, which increases fast when V_1 and V_2 become larger. The convergence of the iterative method to find R is slower than for the previous model of only the loading station.

Parameters		**Storage level** (×1000 ton)			**Number of vessels** (%)				
γ (%)	V_2	90 %	95 %	98 %	0	1	2	3	4
140	2	254	352	482	42	28	30		
140	4	128	213	325	19	14	16	21	30
120	4	395	564	787	36	18	14	15	17

Table 9.4: Results for model of total system with geometric load sizes

Again γ will be increased by some percentage, to make sure that enough vessels arrive to pick up the goods. Since the waiting space is limited, not all vessels will go into the terminal so the total workload of the accepted vessels will be equal to the inflow of the trucks.

See table 9.4 for some results of this model. V_1 is set to 3 and because the probability of three trucks at the unloading station is in all these experiments around 0.005, this value is not changed. Decreasing γ induces higher storage level limits. And less waiting space for the vessels also leads to increasing storage level limits.

9.7 Total system with constant load sizes

In the previous section the load sizes of the trucks and vessels were geometric distributed. In figure 8.8 can be seen that this distribution has a lot more variation than the triangular distribution used in the simulations. The other model in chapter 6 (section 6.2) uses constant load sizes for the trucks and vessels, which is closer to the triangular distribution.

In this model with constant load sizes a square matrix R of order $(r_1V_1 + 1)(r_2V_2 + 1)$ has to be found, where r_i are the load sizes and V_i the maximum number of *vehicles* at the unloading and loading station. In the case study the values of these parameters are $r_1 = 30$ and $r_2 = 49\,505$ and for V_1 and V_2 we take values 3 and 2 respectively. With these values the order of R is $9\,010\,001$, which gives numerical problems by solving the iterative determination of R. When the units of products are truck loads instead of tons, this order still is $(1 \cdot 3 + 1)(1650 \cdot 2 + 1) = 13\,204$.

To decrease this order significantly, we will now assume that not trucks, but trains will deliver product to the terminal. These trains have load sizes of 1 000 ton, which implies that vessels have load sizes of 50 train loads. Now the order of R is $(1 \cdot 3 + 1)(50 \cdot 2 + 1) = 404$, which is low enough to find R_k close enough to the real R.

The values of the other parameters change also with this adjustment. λ becomes $\frac{5000000}{1000 \cdot 365 \cdot 24} \approx 0.57$, $\gamma \approx 0.012$ will be the same as before but again this value will be increased by some percentage, because the waiting area of the vessels is

Parameters		**Storage level** (×1000 ton)			**Number of vessels** (%)					
γ (%)	V_2	90 %	95 %	98 %	0	1	2	3	4	5
140	2	107	159	228	29	42	29			
140	5	2	3	58	2	5	9	18	37	30
120	5	86	177	296	13	11	13	18	27	18

Table 9.5: Results for model of total system with constant load sizes

limited. r_1 is 1, r_2 is 50 and V_1 is set to 3. The results of this model with multiple values for γ and V_2 are shown in table 9.5.

The storage level limits are lower than for the model with the geometric load sizes, which can be explained by the decreased load size variation. In the geometric model there could arrive a couple of small vessels after each other, which will increase the storage level. In the constant model this is not possible. The probability of no vessel at the terminal is also lower in this model, which also induces lower storage levels. Decreasing the arriving rate of the vessel implies increment of the storage level limits, which is also easy to explain by the decreasing number of vessels that arrives at the terminal. When the waiting area for the vessels is larger, the storage level limits decrease, because then a larger amount of time vessels are at the terminal to be loaded.

9.8 Summary

Now all the queueing models can be compared with the simulations of the terminal. For all models we take the same values for the parameters, which are $\lambda \approx 19.03$, $\gamma \approx 0.012$, $\mu_1 = 100$, $\mu_2 \approx 66.67$, $r \approx 1\,650$, $q_1 = 1$ and $q_2 = \frac{1}{r}$. When the number of vessels at the terminal is limited to $V = 5$, the rate γ will be increased with 20 percent. For the system with breakdowns the working time per vessel will be increased with 30 percent, so $T = 1.3\frac{r}{\mu_2}$, which gives $\theta \approx 0.020$. For the models of the total system, V_1 is set to 3 and V_2 to 5. To reduce the order of matrix R in the model with constant load sizes, the same parameters as in section 9.7 are taken.

In table 9.6 the results of the different models are shown together with the results of the Python simulation for one commodity with an empty initial storage. The models are compared with these experiments, because we assumed only one commodity and because the queueing models spend most of their time in the state of an empty storage.

In the simulation the difference between the 90 and 98 percent limit is much smaller than in the mathematical models. The first four experiments in table 9.6b are the closest to a real terminal. The models with the virtual storage level and for the total system with constant load size distribution, seem to give the most similar limits to these four experiments. The difference between these two

Model	**Storage level** (×1000 ton)		
	90 %	95 %	98 %
Virtual storage level	81	172	293
Breakdowns	353	460	600
Finite queue of vessels	362	535	762
Total system with geometric load sizes	344	511	731
Total system with constant load sizes	86	177	296

(a) Queueing models

Experiment	**Storage level** (×1000 ton)		
	90 %	95 %	98 %
Base case	83 ± 5	97 ± 7	114 ± 9
Seasonality of trucks	69 ± 2	77 ± 2	85 ± 3
Other arrival scheme of vessels	77 ± 10	86 ± 11	94 ± 12
Larger arrival delays	88 ± 7	106 ± 9	127 ± 10
Exponential arrivals	229 ± 49	261 ± 51	291 ± 53
One type of vessel	288 ± 74	319 ± 78	342 ± 80
Geometric vessel loads	411 ± 84	453 ± 86	490 ± 86
Geometric vessel loads and $V_2 = 5$	197 ± 47	228 ± 50	259 ± 53

(b) Python, one commodity with empty initial storage

Table 9.6: Summary of results for one commodity

systems is only the speed of the machines in the loading station. In the virtual storage level model the machines are not taken into account and the vessels are loaded in no time. In the total model these loading times are set to the input of the case study, but compared to the arrival rates of the trucks and vessels, these machines are relatively fast. The machines only need to work less than 30 percent of the year.

The last experiment of table 9.6b satisfies the most assumptions of the three other queueing models (breakdowns, finite queue of vessels and total system with geometric load sizes). This experiment has exponential arrivals of one type of vessel with geometric distributed load sizes and its waiting area for the vessels is limited to only 5 vessels at a time. These queueing models, however, seem to give much higher storage level limits than this experiment does and the other two queueing models are closer to the results of this experiment.

9.9 Five commodities

For the terminal with five commodities, the convolution of five probability distributions can be taken to find the storage level limits for the total terminal. To determine the probability distribution of the storage level per commodity, the two best models for one commodity are taken; the model with the virtual storage level and the model of the total system with constant load sizes. But first the parameters per commodity have to be determined.

For the model with the virtual storage levels, the values of λ_c, γ_c and r need to be known. The commodities are respectively responsible for 30, 25, 20, 15 and 10 percent of the total throughput. The arrival rates of the trucks per commodity, λ_c, are therefore this percentage multiplied with the value of λ used for one commodity, so approximately 5.7, 4.8, 3.8, 2.9 and 1.9. The arrival rates of the vessels, γ_c, are also the percentages multiplied with the previously used γ, so approximately 0.0035, 0.0029, 0.0023, 0.0017 and 0.0012. The mean load size of the vessels is the same for all commodities; $r \approx 1650$ truckloads per vessel. Because the number of vessels is in this model limited to a maximum of V, γ_c is increased for all commodities. Let V be 5 again and increase γ with 20 percent.

Per commodity the value for $\bar{x}_{c,min}$ can be determined with equation 5.3, which then gives the probability distribution for the storage levels per commodity. By taking the convolution of these five distributions, the storage level limits of the total terminal are determined.

For the model of the total system with constant load sizes again trains of 1 000 ton deliver the products to the terminal and vessels with load sizes of 50 train loads will pick it up. The values for λ_c are the throughput per commodity divided by the train load and the number of hours in a year. In this model the waiting queues at the unloading and loading station are limited by a maximum of 3 trains and 5 vessels. The values for γ_c are equal to the model of the virtual storage level and again multiplied by 1.2 to increase the outflow of the terminal over the inflow.

The results of these two models are shown in table 9.7a. The results are of the same order. Compared to the storage level limits of the simulation with five commodities again the difference between the 90% and 98% limit is in the mathematical models larger than in the simulations. The 98% storage level limit of the queueing models is around twice this storage level limit of the case study simulations.

Because these queueing models assume exponential arrivals of only one type of vessel, the results could also be compared to the experiment ‘one type of vessel’. In this way the 98% limit seems to come close by, but the other two limits are much lower than the results of this experiment.

Model	**Storage level** (×1000 ton)		
	90 %	95 %	98 %
Virtual storage level	352	474	630
Total system with constant load sizes	363	487	647

(a) Queueing models

Experiment	**Storage level** (×1000 ton)		
	90 %	95 %	98 %
Base case	241 ± 10	259 ± 11	277 ± 12
Seasonality of trucks	285 ± 34	343 ± 46	399 ± 57
Other arrival scheme of vessels	256 ± 13	276 ± 14	298 ± 16
Larger arrival delays	246 ± 8	265 ± 8	286 ± 9
Exponential arrivals	321 ± 37	350 ± 40	377 ± 41
One type of vessel	676 ± 76	708 ± 77	731 ± 76
Geometric vessel loads	960 ± 88	1 006 ± 88	1 046 ± 87
Geometric vessel loads and $V_2 = 5$	941 ± 77	1 002 ± 81	1 049 ± 82

(b) Python, five commodities with empty initial storage

Table 9.7: Summary of results for five commodities

Chapter 10

Conclusion

Finding the storage requirement of a dry bulk terminal is not an easy task. The various stochastic distributions and lots of variables with large impact on the storage level, make it hard to predict the maximum required storage level. As mentioned in the chapter about literature, nowadays the rule of thumb for dry bulk terminals is to have 10 percent of the annual throughput or two to four times the largest shipload per commodity as storage capacity. For the case study this would be 500 000 ton or 180 000-360 000 ton per commodity. Comparing this to the results of the simulation, you could say these amounts are a bit high, but when there is more seasonality over the year this extra storage area could be needed.

When the terminal stores more than one commodity, it is even harder to determine a storage requirement, since it also depends on the kind of commodities and storage area. When the storage area is one open area where the different commodities can be placed in the same area after each other, the results of the base case simulations with five commodities can be used. But when the storage area consists of silos and the terminal stores different commodities that need to be strictly separated, the total capacity of the terminal has to be larger. When in this case one commodity has a lower storage level than usual and another one has a peak storage, it is not possible to store the second commodity in the same silo as the first one. So for all commodities you need to have spare space to handle its peak volumes.

The main problem of this research was the balancing of the in- and outflow of the terminal. A real terminal will have an inflow of products completely equal to the outflow, since all products will be picked up sometime. In the case study 101 vessels arrived in the year of the simulation. Their arrivals were scheduled, so exactly 101 vessels arrived in the simulation with a total workload around 5 million ton. When the arrivals of the vessels were set to a Poisson process, to satisfy the assumptions of the mathematical models, the number of vessels that arrived per year could decrease to 90, but also increase to 120. When 10 vessels less arrive in a year, around 500 000 ton of product will not be picked up, so the storage levels will in this way be much higher than it normally would. The trucks also arrive in a Poisson process, but since they only deliver 30 ton and 19 trucks arrive per hour, at the end of the year the total delivered amount is not that far from 5 million ton. In all simulations without seasonality of truck arrivals, the trucks delivered between 20 000 ton above or below the 5 million throughput, which is not even 1 percent difference.

When in the mathematical models the in- and outflow were set equal to each other, a symmetric random walk arises. Because in that way the probability of an increasing storage level is equal to the probability of decreasing. But symmetric random walks have steady state probabilities that are equal for all states, so the upper limits for the storage level will grow to infinity.

The Poisson arrivals of the vessels is one of the reasons that the mathematical models of this research do not give the desired results to apply them on a real terminal. It could be that models with deterministic arrivals of vessels would have given better results, where only the storage levels at the points in time that vessels arrive were determined, instead of the total time horizon. Other reasons are that some models are too much simplified to compare with a terminal and others give numerical problems by solving it with real input.

Another problem in the mathematical models were the steady state probabilities. The simulations are all run for only one year, again because a real terminal can adjust the longer term scheduling of arrivals to control the storage level. But since the simulation experiments are very dependent on the initial stock, it can be concluded that in one year the steady state probabilities are never reached. Transient probabilities are more comparable to the results of the simulations than the longer term steady state probabilities. It is also possible to look at the probability that some storage level is reached after starting with an empty storage, because only the maximum storage level probabilities have to be determined.

Besides the problems of the mathematical models, the simulation also has its difficulties. It still stays a simplification of the reality and not all aspects of a terminal can be taken into account. For example the order of the vessels into the terminal has a lot of impact on the storage level when there are multiple commodities handled. But the initial stock and arrival scheme of vessels have impact on the storage level as well, so it is hard to conclude to a best way of determining the required capacity.

In the results of the simulations can be seen that a terminal with more commodities needs more storage capacity. For the case study the storage level limits for five commodities were almost twice the limits of one commodity. The seasonality of the arrivals of vessels had a lot of impact on these limits as well. When the trucks got the same seasonality pattern or when the vessel arrivals were better spread over the year, the storage level limits decreased a lot. When the delay times of the vessels became larger, the storage level limits increased as well as the waiting times of the vessels. So the punctuality of the vessels also has its impact on the required storage area.

A clear result of this research is the dependency of the waiting time of vessels and the storage levels. A smaller storage area induces the increment of the waiting times of the vessels. When there is not enough storage area to store the load of multiple vessels, multiple vessels that arrive in a relatively short time will always have long waiting times until new units of product arrive again to load into the vessels. A terminal should make a good consideration about the impact of the size of their storage area on the waiting times of the vessels.

Other subjects that can be investigated in this research area are import terminals and the impact of downtimes. The results in this research are all about an export terminal, which has other characteristics than import terminals. Simulating import terminals give other problems and will have other storage requirements. Downtimes can also affect the requirements, since maintenance and broken machines will slow down the unloading and loading processes. Dry bulk terminals also have a lot of downtimes because of the weather. In rain or strong wind the processes often be paused, which can lead to other results for the storage requirement.

Further research can also be done for this case study by using different mathematical methods. This research focuses on queueing theory and steady state probabilities, but, as already mentioned, transient probabilities could give better results and also deterministic arrivals of vessels are interesting to investigate. Or models used for insurances can be adjusted such that they are usable for this research area. Instead of the probability of going bankrupt, we are interested in the probability that the storage level becomes above some threshold value. Because the storage level is partly controllable by calling more or less vessels and trucks, decision theory models could also help finding a storage requirement.

Bibliography

H. Agerschou. *Planning and design of ports and marine terminals*. Thomas Telford, 2004.

A.S. Alfa. Time-inhomogenous bulk server queue in discrete time: a transportation type problem. *Operations Research*, 30(4):650–658, 1982.

M. Binkowski and B.J. McCarragher. A queueing model for the design and analysis of a mining stockyard. *Discrete Event Dynamic Systems*, 9(1):75–98, 1999.

U. Bugaric, D. Petrovic, Z. Petrovic, M. Pajcin, and G. Markovic-Petrovic. Determining the capacity of unloading bulk cargo terminal using queuing theory. *Strojniški vestnik-Journal of Mechanical Engineering*, 57(5):405–416, 2011.

M.L. Chaudhry and J.G.C. Templeton. *A first course in bulk queues*. Wiley, 1983.

D.J. Daley. Queueing output processes. *Advances in Applied Probability*, 8(2): 395–415, 1976.

M.E. El-Naggar. Application of queuing theory to the container terminal at Alexandria seaport. *Journal of Soil Science and Environmental Management*, 1(4):77–85, 2010.

U.C. Gupta and K. Sikdar. The finite-buffer M/G/1 queue with general bulk-service rule and single vacation. *Performance Evaluation*, 57(2):199–219, 2004.

W. Henderson and P.G. Taylor. Product form in networks of queues with batch arrivals and batch services. *Queueing systems*, 6(1):71–87, 1990.

J-Q. Hu. The departure process of the GI/G/1 queue and its MacLaurin series. *Operations Research*, 44(5):810–815, 1996.

D. Jagerman and T. Altiok. Vessel arrival process and queueing in marine ports handling bulk materials. *Queueing systems*, 45(3):223–243, 2003.

I.W. Kabak. Blocking and delays in M(x)/M/c bulk arrival queueing systems. *Management Science*, 17(1):112–115, 1970.

W.M. Kempa. The departure process for queueing systems with batch arrival of customers. *Stochastic Models*, 24(2):246–263, 2008.

W.K. Kim, K.P. Yoon, G. Mendoza, and M. Sedaghat. Simulation model for extended double-ended queueing. *Computers & Industrial Engineering*, 59 (2):209–219, 2010.

A. Krishnamoorthy and P.V. Ushakumari. A queueing system with single arrival bulk service and single departure. *Mathematical and computer modelling*, 31 (2-3):99–108, 2000.

G. Lodewijks, D.L. Schott, and J.A. Ottjes. Modern dry bulk terminal design. *Bulk Solids Handling*, 27(6):364, 2007.

F.A. Maraghi, K.C. Madan, and K. Darby-Dowman. Batch arrival queueing system with random breakdowns and Bernoulli schedule server vacations having general vacation time distribution. *International Journal of Information and Management Sciences*, 20(1):55–70, 2009.

J. Medhi. Waiting time distribution in a Poisson queue with a general bulk service rule. *Management Science*, 21(7):777–782, 1975.

A.M. Mulinas. *Seaport capacity manual: application to container terminals.* Technical Collection of Fundación VALENCIAPORT. Series Port Planning and Port Management. Fundación Valenciaport, 2012. ISBN 9788494035111.

T. Robenek, N. Umang, M. Bierlaire, and S. Ropke. A branch-and-price algorithm to solve the integrated berth allocation and yard assignment problem in bulk ports. *European Journal of Operational Research*, 235(2):399–411, 2014.

S.M. Ross. *Introduction to Probability Models, ISE.* Academic press, 2006.

D.L. Schott and G. Lodewijks. Analysis of dry bulk terminals: chances for exploration. *Particle & Particle Systems Characterization*, 24(4-5):375–380, 2007.

J.A. Schwarz and M. Epp. Performance evaluation of a transportation-type bulk queue with generally distributed inter-arrival times. *International Journal of Production Research*, 54(20):6251–6264, 2016.

D. Stanford and W. Fischer. The interdeparture-time distribution for each class in the $\Sigma_i M_i/G_i/1$ queue. *Queueing Systems*, 4(3):179–191, 1989.

A.V.S. Suhasini, K.S. Rao, and P.R.S. Reddy. On parallel and series non homogeneous bulk arrival queueing model. *Opsearch*, 50(4):521–547, 2013.

United Nations Conference on Trade & Development UNCTAD. *Port development : a handbook for planners in developing countries*. New York : United Nations, 1978.

T. van Vianen, J. Ottjes, and G. Lodewijks. Dry bulk terminal characteristics. *Bulk Solids India*, pages 1–10, 2011.

T. van Vianen, J. Ottjes, and G. Lodewijks. Simulation-based determination of the required stockyard size for dry bulk terminals. *Simulation Modelling Practice and Theory*, 42:119–128, 2014.

W. Whitt. Approximations for departure processes and queues in series. *Naval Research Logistics Quarterly*, 31(4):499–521, 1984.

Appendix A

Arrival scheme of vessels

In this appendix the arrival scheme of the vessels is shown, which is used in the Trafalquar and Python simulations. It shows the day a vessel arrives, the corresponding vessel type and which commodity it will pick up.

Arrival day	**Vessel type**	**Commodity**
1	Panamax	4
6	Handysize	2
8	Handymax	3
12	Capesize	3
20	Panamax	2
26	Handysize	1
27	Handymax	2
31	Handymax	1
34	Panamax	1
38	Handysize	1
40	Capesize	2
46	Handymax	2
49	Panamax	5
54	Handysize	3
55	Capesize	4
61	Panamax	1
66	Handymax	3
68	Capesize	1
74	Panamax	2
78	Handymax	1
81	Handysize	1
82	Handysize	2

Arrival day	**Vessel type**	**Commodity**
84	Panamax	3
88	Handymax	1
90	Handysize	4
92	Handymax	4
94	Capesize	1
99	Handymax	2
101	Handysize	1
102	Handysize	3
103	Capesize	5
108	Handymax	1
110	Panamax	2
114	Panamax	2
118	Handysize	2
119	Panamax	3
122	Panamax	4
126	Capesize	2
131	Panamax	2
135	Handymax	3
138	Panamax	1
141	Handysize	2
143	Capesize	4
148	Handymax	1
150	Handysize	1
151	Handysize	5
153	Handysize	3
154	Handymax	2
157	Panamax	3
161	Handymax	4
164	Handysize	1
165	Panamax	1
170	Panamax	5
174	Handymax	1
177	Capesize	3
183	Handymax	1
186	Handysize	2
188	Handymax	2
191	Handysize	1
192	Handymax	3
195	Panamax	3

Arrival day	**Vessel type**	**Commodity**
200	Handysize	4
202	Panamax	2
207	Capesize	1
213	Handymax	1
217	Handymax	4
220	Capesize	5
228	Panamax	1
233	Handymax	2
237	Panamax	1
242	Handysize	1
244	Capesize	2
251	Handymax	1
254	Panamax	3
259	Panamax	4
265	Handysize	1
266	Handysize	2
268	Handymax	5
271	Handymax	3
274	Handymax	1
277	Capesize	3
285	Handymax	2
288	Handysize	4
289	Capesize	4
296	Handysize	3
298	Handysize	1
300	Panamax	5
305	Capesize	2
312	Handysize	1
314	Handysize	2
316	Capesize	1
323	Panamax	4
328	Handymax	3
332	Handymax	2
335	Handymax	2
338	Handysize	3
340	Handymax	1
344	Panamax	5
349	Capesize	1
357	Panamax	3
362	Handymax	4

Appendix B

Used distributions

In this appendix the differences of the distributions used in Trafalquar and Python are shown. The first two figures show the unloading time of a truck of 30 ton and loading time of a vessel of 50 000 ton, both served by only one machine. The third figure shows the distribution of the inter-arrival times of the trucks and the last figure shows the delay time distribution of the vessels relative to their scheduled arrival time.

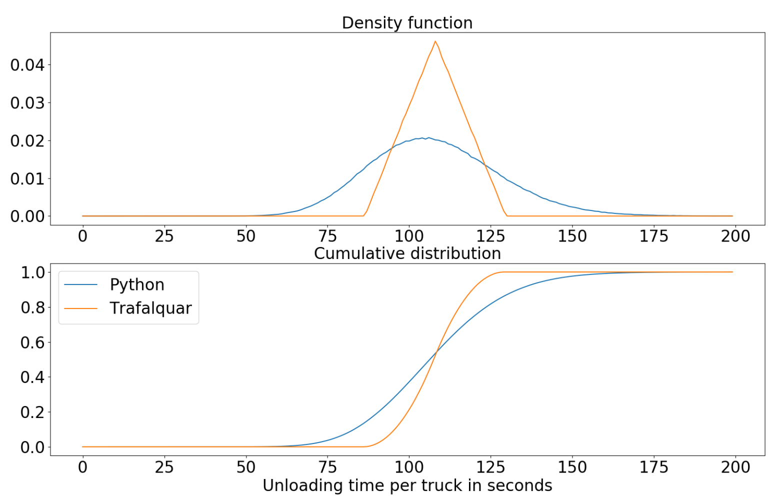

Figure B.1: Unloading time distribution of truck of 30 ton

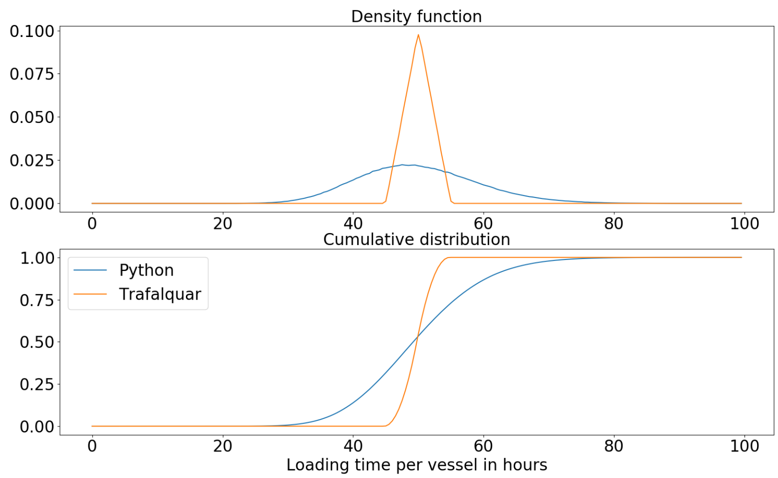

Figure B.2: Loading time distribution of vessel of 50 000 ton

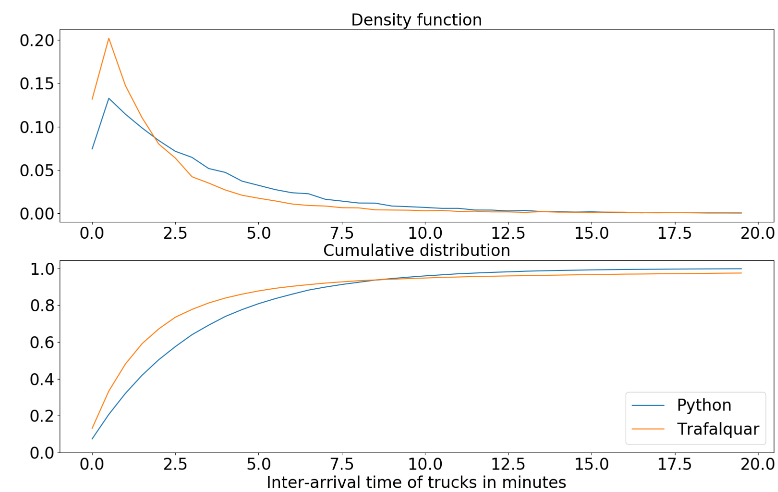

Figure B.3: Inter-arrival time distribution of trucks

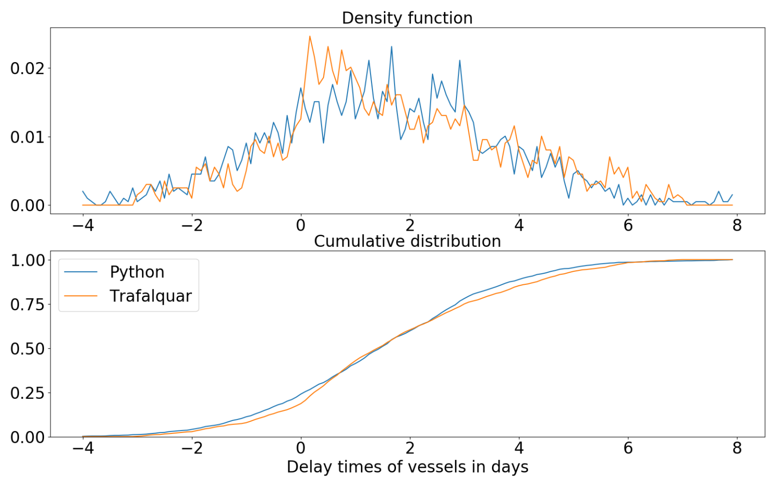

Figure B.4: Delay time distribution of vessels