Project Assignment Part II: Simulation of a base scenario

Jef Jacobs
Toon Eeraerts
Wout Deleu

Semester 2

Contents

1 Introduction

For this assignment, a basic scenario of the 'Yard storage assignment problem' was implemented and simulated. During this report, the basics design choices will be discussed, as well as the results of the simulation.

The basic technology used is Python. The code is written as dynamic as possible, using global variables and booleans to variate parameters and the overall flow of the simulation. It is based on a discrete event simulation, with the possibility of online simulation.

To refresh the environment in brief, the goal is to simulate a yard, in which containergroups[1] come and go. They can arrive from vessels or from trucks and trains. Each container needs to be stored on the yard for a specific duration. The goal is to simulate the storage situations in the yard, as a result of the in and out flow.

2 Simulation parameters

The first and maybe most primary parameter to know is the amount of simulations that needs to be run to get significant results. This can be calculated by the following formula:

$$S = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \overline{X})^2}{n-1}}$$

$$\frac{S}{\sqrt{k}} < d$$

The resulting value of the simulation we wanted to take in account, is the travel distance. So travel distance and average travel distance will be the factor which we will evaluate the variance on.

S the sample variance, the variance of our results. The number of simulations, so the value we want to calculate is k. The accepted standard deviation is d. This is a threshold we want to achieve, and can be chosen in function of the given results. X represents a system result of the simulation. In our case is this

[1]Containers arrive only in group. Containers are (for now) not looked at individually. This means they can't be split up, they are stored in the same place, they enter and leave the yard at the same time.

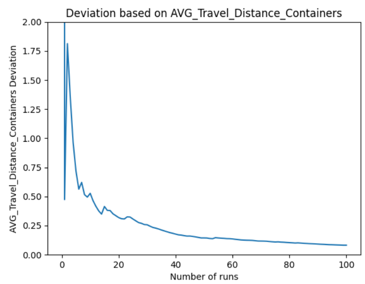

Figure 1: Deviation average travel distance of the containers

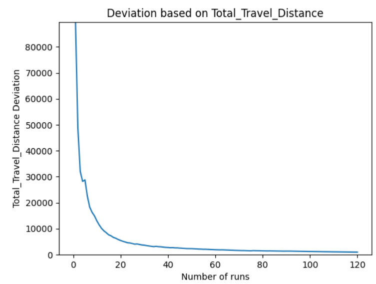

Figure 2: Deviation total travel distance of the containers

the average or total travel distance gotten from running the simulation. When the sample variance based on the k values is within an acceptable range, the calculation is stopped and the value of k is chosen for the amount of simulations that needs to be run.

The acceptable deviation differs for the average and total travel distance. For the average travel distance the deviation chosen is 0.1. This is a low value, but given the speed of execution and the and magnitude of this value, it was a achievable. For the total distance the accepted deviation is 1000. If we look at the magnitude of the total travel distance, we see it surpasses a million. A deviation of a thousands seems in that case reasonable. The total distance travelled is a much larger value than the average distance, because of this the accepted deviation is larger.

The deviation of the average distance travelled is 0.082 after 100 runs. The deviation of the total distance travelled is 997.08 after 120 runs. The evolution of the deviation for both features is shown in figure **??**. In the graph of the average distance, the beginning has more fluctuations than the graph of the total travel distance. This is because the deviation is far smaller than the total distance deviation. From this information we can conclude that at least 120 simulations must be done to get consistent results.

2.1 Generation of parameters

In order to emulate the basic behavior of the yard, there needs to be a stream of containergroups, with their own properties. To perform an online simulation, this stream needs to be generated on the fly. In the prior assignment, a study was held to figure out distributions, which will be used to generate the necessary

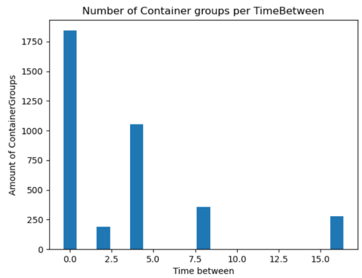

Figure 3: Inter arrival time in prior analysis

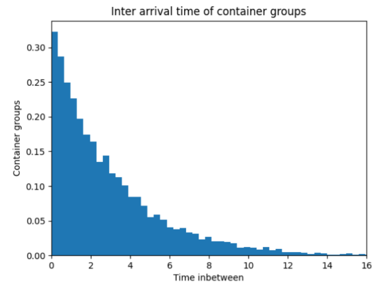

Figure 4: Inter arrival time sample distribution used in the simulation

information regarding the containergroups. This is necessary to generate the input to feed the simulation.

The generation of these groups happens at random times, which are calculated using the arrival time of the previous group, and a random inter arrival time. This interval time is being generated using an exponential function based upon the input data analysis of Nick De Bruyckere, Enrique Miron and Dries Van de Velde (Figure **??**).

When the arrival time of a batch of containers is determined, some properties of that container group needs to be generated. The properties are:

- The type of containers
- The number of containers in the group
- The service time needed
- The arrival position of the group
- The departure position of the group
- The flow type (which can be import or export)

The type of containers is also based upon the input analysis of Nick De Bruyckere, Enrique Miron and Dries Van de Velde represented in Figure **??**. Normal containers and reefer container occur respectively 69% and 31% of the times.

Each group has a certain amount of containers in them. This is chosen based upon our own analysis of the input data. The distribution is represented by a steep exponential distribution which is never lower than 1. (Figure: **??**)

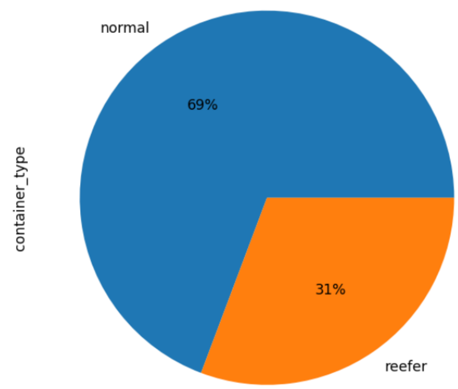

Figure 5: Container type analysis

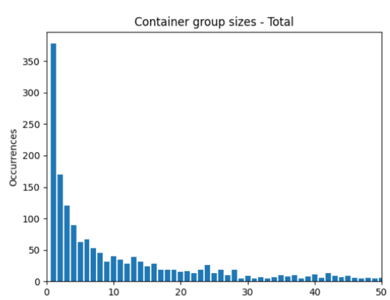

Figure 6: container group size analysis

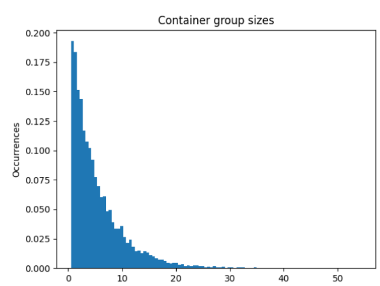

Figure 7: Container group size sample distribution

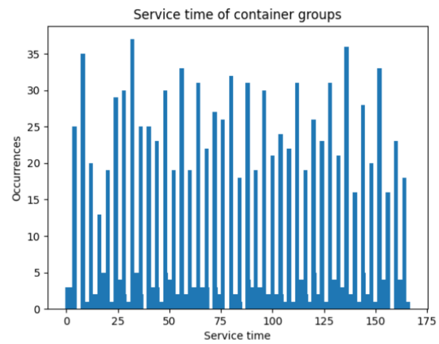

Figure 8: Service time analysis

Each instance has a service time, which is the time containers need to stay in the yard before further actions are taken. This factor is directly dependent on the flow type. If it is import or export, the service time is by default 48 hours. But if it is stated as a transhipment (which is technically a subtype of export), it can vary between 0 and 166 hours. This is based upon our own analysis of the input data which shows a somewhat uniform distribution between 0 and 166 hours (Figure 8). For this reason, the service time sample always takes a random number between these two values.

The arrival and departure position of vessels and trucks is randomly selected out of the available locations.

2.2 Resulting values

Each simulation run will store different characteristics which will be used to evaluate the performance of the simulation. The most important characteristics are:

- Amount of rejected containers and container groups (total and for each type individually)
- The total and average travel distance
- Per YardBlock:
  - The maximal occupancy at any given time
  - The maximal occupancy per day (= **average occupancy per day**)

- Average daily occupancy over all containers: The average occupancy per day over all the YardBlocks

$$\forall i \in days : \frac{\sum_{(x \in YardBlocks)} \overline{Occupancy_{i,x}}}{\#YardBlocks}$$

After seeing the first results, it was clear that with 159 yardblocks in total, it is not the most useful thing to list the individual occupancy for every block, and include this in this rather short report. But to give an idea of this data, some basic statistics are given over the data:

- The amount of YardBlocks which are close to being full (90% average occupancy[2])
- The amount of YardBlocks which are never used (5% average occupancy[2])

3 Results

3.1 Basic scenario

The basic scenario that is discussed in this report describes a decision rule which stores every containergroup that arrives, if there is space in the yard. This will be referred to as *FIFO (First In First Out)*. This apply's to arrival of containers, and wether or not they are stored in the yard. FIFO says that the first container that arrives, has a priority over the ones which arrive after the first one. If there is no space for an arriving containergroup, it will be rejected. While the next containergroup arrives, the check for space will happen again, and so on.

Two different approaches to block assignments are implemented. The block assignment rule has affect on which block is chosen to store a containergroup. The two different situations studied here are *arrival based* and *departure based*. Arrival based and departure based are both based on the minimal distance between 2 points. The yard block chosen is the closest block to the arrival point, in case of arrival based, or closest to the departure point, in case of departure based.

The 2 different approaches give somewhat similar results. The main difference is in the travel distance. This is significantly higher with departure based ap-

[2] Because the simulation is run many times, averages of 100% for example are extremely hard to reach, thats why a small margin of 10% has been taken into account.

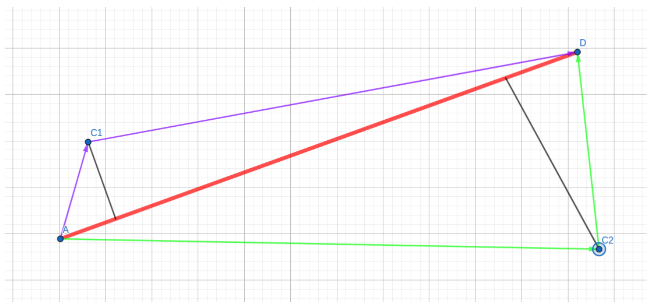

Figure 9: Closest block to arrival and departure point

proach. This could be explained by the fact that groups leave the yard on more similar positions, than they arrive, which could lead to yardblocks getting full, and so containers being stored somewhat further from their optimal point. This is a rather unlikely scenario, because individual blocks are (almost) never even close to full. It is consequently unlikely that containergroups would be stored somewhere else than it's optimal position. Another more likely explanation is that the temporary storage point found for containers on the yard in the departure based implementation is further from the optimal route than when searching for the closest block to the arrival point. This can maybe a vague description, but Figure 9 could clarify this. Point **A** could be an arrival point, point **B** a departure point. **C1** is the yardblock closest to the arrival point, and **C2** to the departure point. The shortest path from A to B is the straight line. We can see that C2 is further from the optimal path than C1. In other words, there are more blocks available close to arrivals (and closer to the arrival points) than to departure points. Trucks have in the current implementation only four possible location. This is approximately ten times less than the amount of berthlocations.

The fact that there are no containers rejected, and the fact that the occupancy is so low, is noteworthy. This implies that the yard is too big, and the stream of containers is to little to stress the simulation even in the slightest way. This could imply that the stream of containers is to light for this yard.

3.2 Stressing the system

The results show that the majority of the yard is not occupied with the normal distribution values. In order to get more relevant results, the decision was made to tune parameters, regardless of the samples. The parameters were tuned in order to get more containers into the yard, to see how it would react to a more intense input.

Containers Rejected	0.0
CG Rejected	0.0
Normal Rejected	0.0
Reefer Rejected	0.0
Total Travel Distance	6051683.63
AVG Travel Distance Containers	380.31
AVG daily total Occupancy	0.0366
Portion of YB never used	0.7799
Portion of YB close to full (at some point)	0
Portion of YB close to full (average)	0

Table 1: FIFO arrival based statistics

Containers Rejected	0.0
CG Rejected	0.0
Normal Rejected	0.0
Reefer Rejected	0.0
Total Travel Distance	7468862.53
AVG Travel Distance Containers	468.10
AVG daily total Occupancy	0.05272
Portion of YB never used	0.7799
Portion of YB close to full (at some point)	0
Portion of YB close to full (average)	0

Table 2: FIFO departure based statistics

A first option could be to decrease the inter-arrivaltimes, so that containergroups arrive faster. To increase the yard occupancy we can also increase the size of the containergroups and increase the service time. Increasing the size of the containergroups has limitations, this is due to the requirement that containergroups need to be stored in a single yardblock. Beyond a threshold, increasing the size of the containergroups will result into a lot of rejected containers without increasing the occupancy. Removing the constraint that containers from a container group can only be stored in a single yardblock would make larger containergroups possible.

Table 3 shows the impact of changing parameters on the rejection of the containers. Table 4 shows the occupation changes when parameters are changed. When the containtergroups arrive up to four times faster than usual, no containers get rejected. Changing the inter-arrivaltime has some impact on the yard occupancy, however the yard is still quite empty. Decreasing this parameter further is not efficient due to the impact it has on the simulation time.

To increase the yard occupation the sizes of the containergroups can be upped.

When the containergroup size is increased to five times the normal size, reefer containers start to get rejected. This is because the largest yardblock for reefer containers is 171, which is much smaller than the largest normal yard block which has the size of 836. This is not in proportion to the 31/69 distribution when generating containers.

The occupancy level of the yard does increase when more containers arrive, but the yard is still quite empty at five times the containergroup size, with a max of 8% occupation.

The service time can be increased to 50 times without having to reject containergroups. The yard is still relatively empty when increasing the service time 50 times with a max occupation of 6%.

In an effort to really stress the yard the three parameters are changed in a single simulation. The size of the containergroups are increased with two, to limit the rejections due to the restriction on storing the containergroup into a single yardblock. When increasing the service time with a factor of 50, doubling the containers in a group and receiving containers at four times the usual speed the yard comes closer to getting full. With a max of 48% and an average of 15%. Still 37% of the yardblocks are never used. Increasing these parameters has mostly impact on the reefer containers. These containers reach the limit much faster than the normal containers with 1109 reefer containergroups rejected and 0 normal containergroups rejected. From this we can conclude that the reefers are more susceptible to parameter changes than the normal containers.

Changed parameters	CG Rejected	Normal Rejected	Reefer Rejected
0.5 * interarrival time	0	0	0
0.25 * interarrival time	0	0	0
2 * containers	0	0	0
5 * containers	1.03	0	202.79
2 * service time	0	0	0
5 * service time	0	0	0
10 * service time	0	0	0
20 * service time	0	0	0
50 * service time	0	0	0
50 * service time 2 * containers 0.25 * interarrival time	1109.8	0	21148.5

Table 3: Container rejection analysis

Changed parameters	AVG Occupancy	YB never used	YB close to full AVG	YB close full Max
0.5 * interarrival time	0.015	0.735	0.006	0.0125
0.25 * interarrival time	0.0243	0.729	0.006	0.018
2 * containers	0.016	0.729	0.006	0.0125
5 * containers	0.03	0.622	0.006	0.0817
2 * service time	0.0133	0.748	0.006	0.006
5 * service time	0.0139	0.735	0.006	0.0125
10 * service time	0.02974	0.723	0.006	0.0125
20 * service time	0.0481	0.71	0.006	0.0314
50 * service time	0.0856	0.6729	0.0125	0.061
50 * service time 2 * containers 0.25 * interarrival time	0.409	0.37	0.1572	0.48

Table 4: yard occupation analysis

4 Conclusion

We can conclude that using sampling based on the original is not sufficient to really stress the yard. The yard is to big, or the flow is to light. A second point is the fact that there could be a real impact of the fact that there are only four arrivalpoints for trucks, which could really impact the travel time. The arrival based implementation is the better choice.

While stressing the system, it can be said that the generation progress of reefers relative to the normal containers is not in proportion to the storage capacity of them. The bottleneck are the reefer containers. A last conclusion is the same as the first one, the yard is to big or the flow is to light. And while tweaking parameters, it is clear it is far to light. We need to push the system in rather extreme ways to get the yard to fill up.