Project Assignment Part II: Simulation of a base scenario

Jef Jacobs
Toon Eeraerts
Wout Deleu

Semester 2

Contents

1 Introduction

During this assignment, a base was implemented to build a simulation for the 'Yard storage assignment problem'. The basic structure was designed and there is a basic scenario implemented. 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. The simulation 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 gow. They can arrive from vessels or from trucks and trains. 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:

$$JEFHEEFTEENKLEINEPENIS$$

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

The generation of the groups happens at random times, which are calculated

[1]Containers arrive only in group. Containers individually 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.

using the arrival time of the previous group, and a random interval time. This interval time is being generated using a .... **TODO:: UITLEGGEN - distributie**

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 in this case randomly selected with a weight of 69% in favor of the *normal* containers. The other 31% is designated to the *reefer* containers. Each group has a certain amount of containers in them. This is chosen based on .... **TODO:: UITLEGGEN - distributie**. Each instance has a service time, which is the time containers needs 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), than it can vary between .... **TODO:: UITLEGGEN distributie**.

The arrival position or berthing location (**TODO:: Klopt dit??**)of a vessel is randomly selected .... **TODO:: UITLEGGEN distributie**. The same principle holds for the departure position.

2.2 Evaluation of results TODO::: EEN ANDERE TITEL

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 tim
  - The maximal occupancy per day (**= average occupancy per day**)
- Average daily occupancy over all containers: The average occupancy per day over all the YardBlocks

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

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. If there is no space, the container is rejected. This will be referred to as *FIFO (First In First Out)*. This apply's to arrival of containers, and wether or not they are kept 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 assignment 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 both 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.

3.2 Departure based

3.2.1 Arrival based

3.3 Stressing the system

Containers Rejected	0.0
CG Rejected	0.0
Normal Rejected	0.0
Reefer Rejected	0.0
Total Travel Distance	7471576.566740074
AVG Travel Distance Containers	467.63038278646087
AVG daily total Occupancy	0.012189307994940618

Table 1: FIFO departure based statistics

Containers Rejected	0.0
CG Rejected	0.0
Normal Rejected	0.0
Reefer Rejected	0.0
Total Travel Distance	6069403.996520092
AVG Travel Distance Containers	380.0611989034411
AVG daily total Occupancy	0.011198756687437594

Table 2: ...