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

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 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:

[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.

- 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

3.2 Departure based

YB1	YB2	YB3	YB4	YB5	YB6	YB7	YB8	YB9
0.32	0.02	0.0	0.04	0.08	0.16	0.18	0.0	0.0

Table 1: ...

YB10	YB11	YB12	YB13	YB14	YB15	YB16	YB17	YB18
0.0	0.0	0.0	0.0	0.27	0.65	0.05	0.13	0.1

Table 2: ...

YB19	YB20	YB21	YB22	YB23	YB24	YB25	YB26	YB27
0.11	0.12	0.16	0.12	0.22	0.12	0.17	0.11	0.24

Table 3: ...

YB28	YB29	YB30	YB31	YB32	YB33	YB34	YB35	YB36
0.55	0.37	0.07	0.04	0.0	0.04	0.0	0.09	0.49

Table 4: ...

YB37	YB38	YB39	YB40	YB41	YB42	YB43	YB44	YB45
0.0	0.01	0.04	0.02	0.02	0.02	0.02	0.0	0.02

Table 5: ...

3.2.1 Arrival based

3.3 Stressing the system

YB46	YB47	YB48	YB49	YB50	YB51	YB52	YB53	YB54
0.01	0.02	0.01	0.02	0.02	0.01	0.01	0.02	0.01

Table 6: ...

YB55	YB56	YB57	YB58	YB59	YB60	YB61	YB62	YB63
0.0	0.03	0.01	0.02	0.02	0.02	0.03	0.03	0.05

Table 7: ...

YB64	YB65	YB66	YB67	YB68	YB69	YB70	YB71	YB72
0.05	0.02	0.04	0.02	0.02	0.03	0.01	0.01	0.03

Table 8: ...

YB73	YB74	YB75	YB76	YB77	YB78	YB79	YB80	YB81
0.02	0.04	0.0	0.02	0.02	0.01	0.0	0.0	0.0

Table 9: ...

YB82	YB83	YB84	YB85	YB86	YB87	YB88	YB89	YB90
0.0	0.0	0.0	0.0	0.02	0.05	0.02	0.02	0.02

Table 10: ...

YB91	YB92	YB93	YB94	YB95	YB96	YB97	YB98	YB99
0.02	0.0	0.0	0.02	0.02	0.01	0.0	0.0	0.0

Table 11: ...

YB100	YB101	YB102	YB103	YB104	YB105	YB106	YB107	YB108
0.0	0.0	0.0	0.01	0.01	0.02	0.02	0.03	0.02

Table 12: ...

YB109	YB110	YB111	YB112	YB113	YB114	YB115	YB116	YB117
0.01	0.01	0.0	0.0	0.0	0.01	0.01	0.01	0.0

Table 13: ...

YB118	YB119	YB120	YB121	YB122	YB123	YB124	YB125	YB126
0.0	0.01	0.0	0.07	0.0	0.02	0.01	0.06	0.04

Table 14: ...

YB127	YB128	YB129	YB130	YB131	YB132	YB133	YB134	YB135
0.08	0.04	0.0	0.01	0.03	0.02	0.01	0.02	0.01

Table 15: ...

YB136	YB137	YB138	YB139	YB140	YB141	YB142	YB143	YB144
0.02	0.04	0.02	0.03	0.05	0.04	0.03	0.03	0.03

Table 16: ...

YB145	YB146	YB147	YB148	YB149	YB150	YB151	YB152	YB153
0.03	0.0	0.0	1.0	0.21	0.11	0.05	0.09	0.13

Table 17: ...

YB154	YB155	YB156	YB157	YB158	YB159
0.06	0.22	0.03	0.0	0.0	0.58

Table 18: ...

YB1	YB2	YB3	YB4	YB5	YB6	YB7	YB8	YB9
0.05	0.0	0.0	0.0	0.0	0.01	0.02	0.0	0.0

Table 19: ...

YB10	YB11	YB12	YB13	YB14	YB15	YB16	YB17	YB18
0.0	0.0	0.0	0.0	0.02	0.15	0.0	0.0	0.0

Table 20: ...

YB19	YB20	YB21	YB22	YB23	YB24	YB25	YB26	YB27
0.0	0.0	0.0	0.0	0.01	0.0	0.01	0.0	0.01

Table 21: ...

YB28	YB29	YB30	YB31	YB32	YB33	YB34	YB35	YB36
0.11	0.04	0.0	0.0	0.0	0.0	0.0	0.0	0.08

Table 22: ...

YB37	YB38	YB39	YB40	YB41	YB42	YB43	YB44	YB45
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 23: ...

YB46	YB47	YB48	YB49	YB50	YB51	YB52	YB53	YB54
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 24: ...

YB55	YB56	YB57	YB58	YB59	YB60	YB61	YB62	YB63
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 25: ...

YB64	YB65	YB66	YB67	YB68	YB69	YB70	YB71	YB72
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 26: ...

YB73	YB74	YB75	YB76	YB77	YB78	YB79	YB80	YB81
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 27: ...

YB82	YB83	YB84	YB85	YB86	YB87	YB88	YB89	YB90
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 28: ...

YB91	YB92	YB93	YB94	YB95	YB96	YB97	YB98	YB99
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 29: ...

YB100	YB101	YB102	YB103	YB104	YB105	YB106	YB107	YB108
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 30: ...

YB109	YB110	YB111	YB112	YB113	YB114	YB115	YB116	YB117
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 31: ...

YB118	YB119	YB120	YB121	YB122	YB123	YB124	YB125	YB126
0.0	0.0	0.0	0.01	0.0	0.0	0.0	0.0	0.0

Table 32: ...

YB127	YB128	YB129	YB130	YB131	YB132	YB133	YB134	YB135
0.02	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 33: ...

YB136	YB137	YB138	YB139	YB140	YB141	YB142	YB143	YB144
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 34: ...

YB145	YB146	YB147	YB148	YB149	YB150	YB151	YB152	YB153
0.0	0.0	0.0	1.0	0.03	0.01	0.0	0.01	0.02

Table 35: ...

YB154	YB155	YB156	YB157	YB158	YB159
0.0	0.06	0.0	0.0	0.0	0.17

Table 36: ...

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 37: ...

YB1	YB2	YB3	YB4	YB5	YB6	YB7	YB8	YB9
0.34	0.0	0.0	0.0	0.0	0.23	0.26	0.0	0.0

Table 38: ...

YB10	YB11	YB12	YB13	YB14	YB15	YB16	YB17	YB18
0.0	0.0	0.0	0.0	0.33	0.38	0.21	0.0	0.0

Table 39: ...

YB19	YB20	YB21	YB22	YB23	YB24	YB25	YB26	YB27
0.0	0.21	0.0	0.0	0.28	0.0	0.0	0.0	0.21

Table 40: ...

YB28	YB29	YB30	YB31	YB32	YB33	YB34	YB35	YB36
0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5

Table 41: ...

YB37	YB38	YB39	YB40	YB41	YB42	YB43	YB44	YB45
0.0	0.06	0.11	0.05	0.0	0.0	0.07	0.0	0.0

Table 42: ...

YB46	YB47	YB48	YB49	YB50	YB51	YB52	YB53	YB54
0.06	0.04	0.04	0.05	0.0	0.0	0.06	0.04	0.0

Table 43: ...

YB55	YB56	YB57	YB58	YB59	YB60	YB61	YB62	YB63
0.0	0.13	0.0	0.06	0.05	0.05	0.08	0.05	0.12

Table 44: ...

YB64	YB65	YB66	YB67	YB68	YB69	YB70	YB71	YB72
0.1	0.05	0.09	0.07	0.0	0.11	0.0	0.0	0.1

Table 45: ...

YB73	YB74	YB75	YB76	YB77	YB78	YB79	YB80	YB81
0.0	0.12	0.0	0.0	0.08	0.08	0.0	0.0	0.0

Table 46: ...

YB82	YB83	YB84	YB85	YB86	YB87	YB88	YB89	YB90
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 47: ...

YB91	YB92	YB93	YB94	YB95	YB96	YB97	YB98	YB99
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 48: ...

YB100	YB101	YB102	YB103	YB104	YB105	YB106	YB107	YB108
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 49: ...