Technical University of Denmark DTU

TECHNICAL UNIVERSITY OF DENMARK – MANAGEMENT ENGINEERING

MASTER THESIS

Simulation study of different yard and crane side operations at a container terminal

---

Xavier Estadella Valls

Student number: 176397

Supervised by Peter Jacobsen and Dario Pacino

22nd June 2018

Abstract

With the expected growing in maritime container terminal and the exponential growth in vessels size, container terminals operations are facing higher demands. Being able to simulate the operations that occur inside container terminals can lead to obtain valuable insights, help to identify possible problems and solve them, and also some decision-making tools can be created.

This thesis, then, uses simulation to model the operations of a container terminal in Denmark. The movements of containers from the storage area to the quay crane for loading and from the quay crane to the storage area for unloading are modelled. The simulation software used is AnyLogic. Once the model is created, an explanation of it and its limitations are presented. The databases used to run the simulation are explained and a validation of the model is performed with real data. After that, a probabilistic function to generate new data is found in order to run simulations of future vessels. Four different vessels are analysed with real data and generated data. A comparison of the results is made and some conclusions are extracted. To end, some recommendations for future work are presented.

Acknowledgements

I would like to thank my two supervisors, Dario Pacino and Peter Jacobsen, for their help, support and encouragement during the entire project. I would like to thank them, as well, for not only accepting to supervise me, but encouraging me and helping me define which way to go.

To my parents and my brother, an immensely grateful thank you: your support, love and advise have helped me get here. To my fellow exchange friends, thank you for your invaluable comments and criticism, and all the experiences we shared throughout this semester. In general, I want to thank all those friends who (directly or indirectly, knowing it or not) have helped me.

Contents

Abstract ***1***

Acknowledgements ***2***

List of Figures ***5***

List of Tables ***7***

List of Abbreviations ***8***

1. Introduction and problem statement ***9***

1.1 Introduction **9**

1.2 Problem statement **10**

2. APM Terminals ***11***

3. Literature review ***14***

4. Analysis ***17***

4.1 Model **17**

4.1.1 Layout 17
4.1.2 Agents 20
4.1.2.1 Container40 20
4.1.2.2 Container20 21
4.1.2.3 StraddleCarrier 21
4.1.3 Processes flows 22
4.1.3.1 Unloading Process 24
4.1.3.2 Loading Process 26
4.1.4 Model Limitations 28

4.2 Databases **29**

4.2.1 DIS_40feet and DIS_20feet 30
4.2.2 LOAD_40feet and LOAD_20feet 31

4.3 Model validation **32**

4.4 Data fitting **34**

4.4.1 Data validation 34
4.4.2 Basic dataset information: Graphical summary 37
4.4.3 Selection of candidate distributions and Estimation of model parameters (Maximum Likelihood Estimation) 38
4.4.4 Validation of the candidates 39
4.4.4.1 Visual Validation 39
4.4.4.2 QQ Plots 41
4.4.4.3 Goodness of fit: Kolmogorov-Smirnov Test (K-S) and Pearson Test (Chi-squared) 45
4.4.5 Resume for all Datasets 48
4.4.6 Generating new data 50
4.4.6.1 Generation 50
4.4.6.2 Merge 51

4.5 Case analysis **52**
4.5.1 Vessel 09710AAT 52
4.5.1.1 Vessel Information 52
4.5.1.2 Real Data analysis 53
4.5.1.3 Generated Data analysis (Gamma) 54
4.5.1.4 Generated Data analysis (Weibull) 56
4.5.2 Vessel 09097AAT 58
4.5.2.1 Vessel Information 58
4.5.2.2 Real Data analysis 59
4.5.2.3 Generated Data analysis (Gamma) 60
4.5.2.4 Generated Data analysis (Weibull) 62
4.5.3 Vessel 09506AAT 64
4.5.3.1 Vessel Information 64
4.5.3.2 Real Data analysis 65
4.5.3.3 Generated Data analysis (Gamma) 66
4.5.3.4 Generated Data analysis (Weibull) 68
4.5.4 Vessel 09505AAT 70
4.5.4.1 Vessel Information 70
4.5.4.2 Real Data analysis 71
4.5.4.3 Generated Data analysis (Gamma) 72
4.5.4.4 Generated Data analysis (Weibull) 74
4.6 Conclusions of the analysis **76**
5. Recommendations and perspective ***78***
6. Bibliography ***79***

List of Figures

Figure 1. Yard layout of the container terminal 12
Figure 2. Container positions inside a block. 13
Figure 3. Network, SC Zone, Vessel Zone and QC points 18
Figure 4. Storage Zone 18
Figure 5. Block Simplification 19
Figure 6. Model layout 19
Figure 7. Unloading 40feet flow process 24
Figure 8. Unloading 20feet flow process 25
Figure 9. Loading 40feet flow process, introducing containers 26
Figure 10. Loading 40feet flow process, picking up containers 27
Figure 11. Aarhus port terminal yard layout 28
Figure 12. Acf of the dataset 35
Figure 13. Boxplot of the dataset 35
Figure 14. Histogram of the data 37
Figure 15. Visual comparison: Dataset and Gamma distribution candidate 39
Figure 16. Visual comparison: Dataset and Weibull distribution candidate 40
Figure 17. Visual Comparison: Dataset and Exponential distribution candidate 40
Figure 18. QQ-Plot of Gamma distribution candidate 42
Figure 19. QQ-Plot of Weibull distribution candidate 43
Figure 20. QQ-Plot of Exponential distribution candidate 44
Figure 21. How a position is defined inside a vessel 51
Figure 22. Real Data analysis for 09710AAT vessel 53
Figure 23. Generated Data analysis (Gamma) for 09710AAT vessel 54
Figure 24. Real and Generated Data (Gamma) analyses for 09710AAT vessel 55
Figure 25. Generated Data analysis (Weibull) for 09710AAT vessel 56
Figure 26. Real and Generated Data (Weibull) analyses for 09710AAT vessel 57
Figure 27. Real Data analysis for 09097AAT vessel 59
Figure 28. Real and Generated Data (Gamma) analyses for 09097AAT vessel 60
Figure 29. Real and Generated Data (Gamma) analyses for 09097AAT vessel 61
Figure 30. Generated Data (Weibull) analyses for 09097AAT vessel 62
Figure 31. Real and Generated Data (Weibull) analyses for 09097AAT vessel 63
Figure 32. Real Data analysis for 09506AAT vessel 65
Figure 33. Generated Data (Gamma) analysis for 09506AAT vessel 66
Figure 34. Real and Generated Data (Gamma) analyses for 09506AAT vessel 67
Figure 35. Generated Data (Weibull) analysis for 09506AAT vessel 68
Figure 36. Real and Generated Data (Weibull) analyses for 09506AAT vessel 69

Figure 37. Real Data analysis for 09505AAT vessel 71
Figure 38. Generated Data (Gamma) analysis for 09505AAT Vessel 72
Figure 39. Real and Generated Data (Gamma) analyses for 09505AAT Vessel 73
Figure 40. Generated Data (Weibull) analysis for 09505AAT Vessel 74
Figure 41. Real and Generated Data (Weibull) analyses for 09505AAT Vessel 75

List of Tables

Table 1. Equity weighted throughout 11
Table 2. Explanation of AnyLogic blocks used in the model 23
Table 3. Comparison of the real time and simulation time needed for vessel 32
Table 4. Estimated parameters of the three probabilistic candidates 48
Table 5. Validation tests results for Gamma and Weibull distributions 48
Table 6. Validation tests results for Exponential distribution 49
Table 7. Parameters of the probabilistic distributions chosen 49
Table 8. 09710AAT Vessel information 52
Table 9. 09097AAT Vessel information 58
Table 10. 09506AAT Vessel information 64
Table 11. 09505AAT Vessel information 70
Table 12. Average difference in maximum resource utilization 76

List of Abbreviations

CT – Container Terminal

AGV – Automated Guided Vehicle

ALV – Automated Lifting Vehicle

RS - Reach Stacker

YT - Yard Truck

SC – Straddle Carrier

1. Introduction and problem statement

1.1 Introduction

Containers appeared in the market for international conveyance of sea freight almost five decades ago. In terms of value, global maritime container trade is believed to account for approximately 60 percent of all world maritime trade, which was valued at around 12 trillion U.S dollars in 2017. The ones in charge of managing this big market are the container terminals (CT). Due to an expecting growth in the global market demand of 4,7% between 2016 – 2019 period and the exponential growth in vessels size; container terminals, container logistics and management are facing higher demands. They need to adapt in order to remain productive and attractive to shipping lines in a highly competitive environment. To accomplish this, container terminals must be able to handle an increased amount of container traffic, expansion projects or operations research are the two answers.

The majority of all container ports have the same work flow. When a vessel berths at least one quay crane (QC) is assigned to it, but also more QCs can be assigned. Right after a vessel has berthed, all the QCs assigned start unloading and loading containers. The containers are stored for further transportation by truck, train or vessel. The storage area also temporarily holds containers that will be loaded onto the vessels.

Usually, container terminals are characterized by means of their specific equipment and stacking facilities. Vehicles are used to transport containers from the storage area to the QCs and vice versa. There are many options to carry out all these movements: automated guided vehicles (AGV), automated lifting vehicles (ALV), reach stackers (RS), yard trucks (YT), or straddle carriers (SC).

AGVs are only able to move containers horizontally. Yard cranes or gantry cranes are needed at the quay and in the yard to load and unload containers. The only difference with AVLs is that these last ones can lift and drop containers on their own so no yard cranes or gantry cranes are needed. YTs are basically a manually driven AGV and SCs are the most flexible solution since they are able to lift, drop and stack containers by themselves. If SCs are used, yard cranes are not needed in the storage area while with all the other vehicles they are needed. Finally, RSs are also able to lift, drop and stack by themselves but they are normally used in small ports. The container port terminal studied in this project uses Straddle Carriers to transport containers from the storage area to the QCs and vice versa.

The focus of this project will be the container movements between QCs and the storage area.This project is based on data of the container port terminal of Aarhus (Denmark). APM Terminals, the company that operates the container terminal, has provided the data. At this port, SCs are used to transport containers between the QCs and the stacking area. A model that reproduces the movements that SCs have to do each time a vessel berth will be created. And thanks to this, simulations for future vessels can be done and the decision of how many SCs should be used can be taken.

The remainder of this project is structured as follows. A brief explanation of the company APM Terminals will be presented. Then some literature review will be performed looking for papers about simulation in container terminals, operational research, straddle carrier ports optimization, etc. After that, the analysis will be performed. First of all, an explanation of how the model has been build will be presented, followed by another explanation about the databases that are needed to run the simulations. A model validation will also be presented to check that the model reproduces the real world. Next, a fitting process to find a probabilistic distribution capable of generating new times for run future simulations will be presented. And to end, four case analyses will be studied with real times and generated times to compare the results and some conclusions will be drawn.

1.2 Problem statement

Being able to determine how many SCs are needed to perform all the movements in a vessel's berth is crucial. A good optimization of the resources always helps to minimize costs and achieve a reasonable performance. Having a decision-making tool available that helps determining the number of SCs needed is a good solution. Here is where simulation is used. It allows to reproduce of all the movements of vessels that will berth in the CT. Allowing to recollect value data, more specifically the SC utilization and movements done, plot a graph and use it to determine the amount of SC needed.

2. APM Terminals

APM Terminals is an international container terminal company that operates a global terminal network of 20.000 professionals with 76 operating port and terminal facilities in 41 countries, as well as 117 Inland Services in 37 countries representing a global presence in 59 countries on five continents. Five new facilities are in development, 4 of them will open in 2019 and the last one in 2020 with a total investment of USD $3,7 billion. Apart from that, 10 terminals are under upgrade or expansion.

Based in The Hague (The Netherlands) and founded in 2001, the company works with 60 shipping lines, importers and exporters, governments, business leaders and the entire global supply chain to provide solutions that help nations achieve their ambitions and business reach their performance goals. APM Terminals accomplishes it through high productivity operations and port capacity in economically, environmentally and socially responsible ways.

Looking at the owned terminals, 24 are in Europe, Russia and the Baltics, 20 in Asia, 16 in Africa and the Middle East and 15 in The Americas. About the five new facilities, three will be opened in Africa (Morocco, Ghana and Ivory Coast) one in Europe (Italy) and the last one in The Americas (Costa Rica). As expected, those regions with more terminals have higher equity weighted throughout (see in *Table 1*).

(Million TEU)	2016	2015
The Americas	6,4	6,7
Europe, Russia and the Baltics	11,8	10,6
Asia	12,5	12,1
Africa and the Middle East	6,6	6,6
TOTAL	37,3	36

Table 1. Equity weighted throughout

APM Terminals generated USD $4,17 billion in revenue in 2016, of which USD $436 million are profit making it a 5,7% of Return on Investment (ROI). During 2016 APM Terminals acquired eight out of eleven terminals from the Spanish Grup Marítim TCB's port and rail interests. The acquisition added a combined 2 million in TEU equity-weighted volume to APM Terminals.

APM Terminals – Aarhus A/S will be the container terminal under study in this master's thesis, strategically located on the Baltic Sea in Denmark's largest port, and second largest city, close to the principle manufacturing and industrial centers of Western Denmark. Offering 15 meters of deepwater berth is one of the most productive container facilities in Europe. MSC, Maerks Line, Eimskip, K-Line, Containerships, Unifeeder, Sea Connect and Teamlines are the shipping lines calling APM Terminals – Aarhus A/S nowadays.

The Container Terminal is equipped with some of the largest container cranes on the market, which have a high level of efficiency. The cranes are able to move up to 35 containers on average per hour per crane. The Terminal also has railway tracks all the way to the quays and good facilities for cooling and refrigerating containers. The Port of Aarhus is an important hub for ports in Northern, Eastern and Southern Europe as well as for eastern Mediterranean and the Far East, to which it has regular routes. The rest of the world is reached through weekly feeder connections with continental ports.

See *Figure 1* below the yard layout of the Container Terminal:

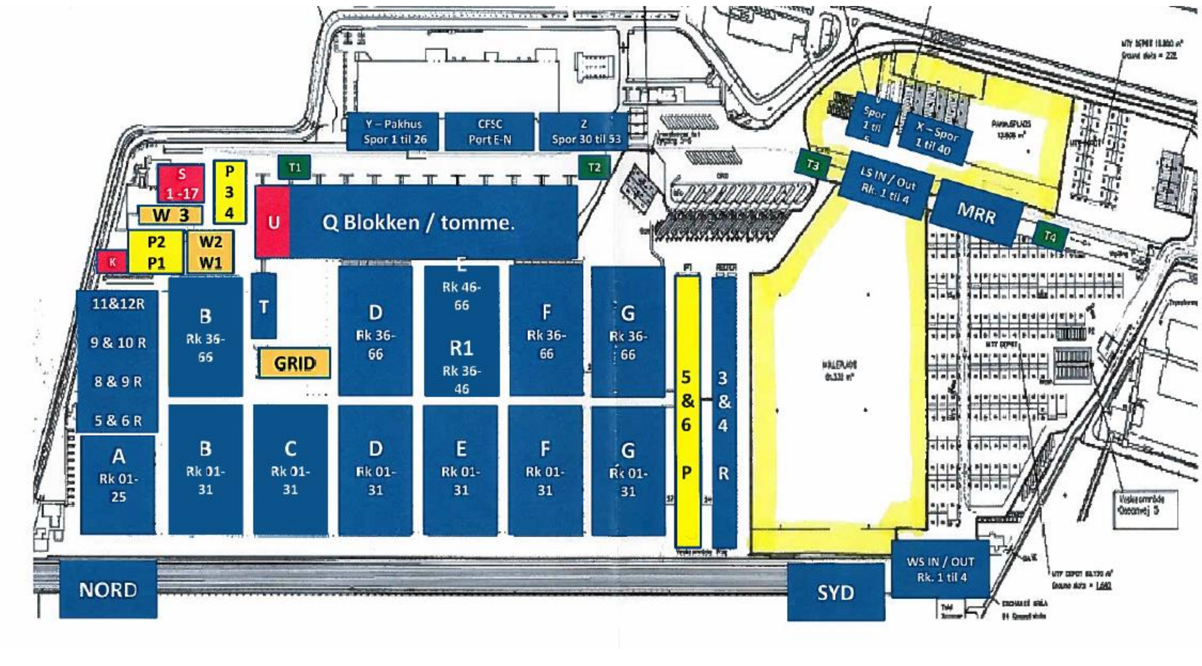


Figure 1. Yard layout of the container terminal

Each of the rectangles, named with letters, is called a block and is where the containers are stored. All the positions of all the blocks have been provided and have the following look:

$$014C02.B$$

The first three numbers indicate the row, lowest numbers towards the quayside. Then the letter is the block. The next two numbers indicate the bay, the position inside the row, lowest numbers towards the north part of the quayside. Finally, the letter indicates the tier, being A the one on the bottom and C the one on the top. To better understand the positions inside a block, see *Figure 2* below:

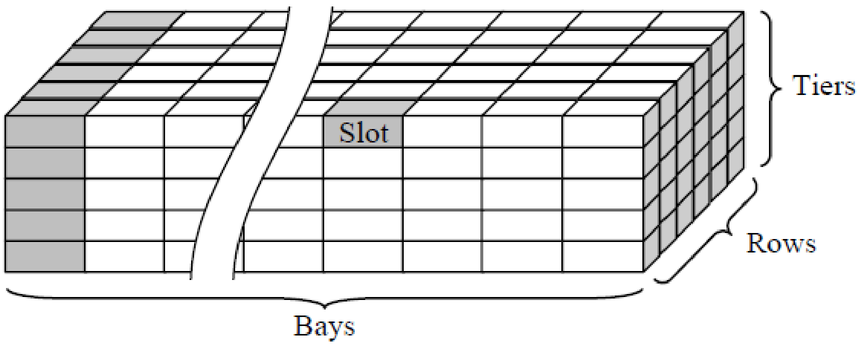


Figure 2. Container positions inside a block.

3. Literature review

Logistic operations in container terminals are becoming more and more important. Therefore, an ever-increasing number of publications on container terminals have appeared in the literature. First of all, some literature review about general container terminal systems will be presented, followed by a more specific operational problems literature review and will end with a literature review about simulation in container terminals, since, simulation will be the basis of this project.

Meersmans and Dekker [7] talk about the use of operation research models and methods in the design and operation of container terminals. Activities that take place at a container terminal are described and an overview of the relevant decision problems is given. For each of these problems the appropriate operations research contributions are discussed. Murty et al. [18] describe a variety of inter-related decisions made during daily operations at a container terminal. They work to develop decision support tools and discuss the mathematical models and algorithms used in their design, the reasons for using these approaches, and some experimental results.

Focusing more on literature about specific operational problems, berth planning is the first one someone may think about. These problems can be formulated as different combinatorial optimization problems depending on the specific objectives and restrictions that have to be observed. Legato and Mazza [16] present a queuing network model of the logistic activities related to the arrival, berthing, and departure processes of vessels at a container terminal. Imai et al. [13] modify the existing formulation of the berth allocation problem in order to treat calling vessels at various service priorities by developing a genetic algorithm-based heuristic for the resulting non-linear problem.

Stowage planning, the act of allocating space to containers on board of a container ship in the order of the discharge ports, is the core of ship planning. The shipping line's stowage plan has to be designed for all ports of a vessel's rotation. The objective is to minimize the number of shifts during port operation and to maximize the vessel's utilization. Avriel et al. [9] [10] deal with stowage plan for containers in a container ship. The paper aims to find a stowage plan that minimizes the shifting cost, showing that the shift problem is NP-complete. Shifting is defined as the temporary removal from and placement back of containers onto a stack of containers. Dubrosky et al. [11] develop an efficient heuristic for solving the stowage problem. A genetic algorithm technique is used for solving the problem. A compact and efficient encoding of solutions is developed and the efficiency of it is demonstrated through an extensive set of simulations runs.

The third step of ship planning is the allocation of QCs to vessels and the vessels' sections – called the crane split scheduling. Depending on the vessel's size more QCs are used or not and not all QCs can operate a berth because of possible technical difficulties. That occurs because terminals are historically grown, meaning that different type of QCs exist at the terminals and maybe some old ones cannot operate with the newest vessels because of technical differences. Daganzo [8] examines crane scheduling for ports. The paper presents exact and approximate solution methods for crane scheduling. The approximation methods are based on optimally principles and are easy to implement. The exact methods can only be used for a few ships.

Storage and stacking logistics has become more and more important because of the increasing of container as a method of goods transportation therefore more containers have to be sorted in container terminals as traffic grows continuously and space is becoming a scarce resource. Kim [14] propose a methodology to estimate the expected number of rehandles to pick up an arbitrary container and the total number of rehandles to pick up all the containers in a bay for a given stacking configuration.

A literature review regarding quayside transport is distinguished mainly based on the means of transport used. As mentioned in the introduction, AGVs, AVLs, RSs, YTs and SCs. The number of references of AGVs is enormous, Ever and Koppers [12] proposes a new modelling technique which has been used to successfully model the relevant aspects of traffic control. The control can be imposed by using a hierarchical system of so called semaphores, thus it is possible to follow a structural approach in the design of a traffic control configuration. Kim and Kim [15] discusses how to route straddle carriers during the loading operation of export containers in port terminals. The objective of the routing is to minimize the total travel distance of straddle carriers in the yard.

Finally, a literature overview of simulation modelling in ports and container terminals is presented below. Is based in a review paper of this specific field, 219 papers have been analysed in the paper. Among these, 209 present a simulation of a port o container terminal operations and 10 are review papers. What is significant is that 32 papers are from 1961 to 1999, which are substantially lower than the 187 published papers in the period of 2000-2015. That indicates that simulation in container terminals is becoming more and more useful for research purposes, decision support tool and design of new ports.

Most of the papers, exactly 75,8% of them, are focused in container terminal operations and the remaining 24,8% are focused in port operations (bulk operations, ports in general and port traffic). AnyLogic will be the simulation software used to model the yard movements in the container port terminal of Aaruhs. Two papers that work with this software are presented below. Kondratyev [22] discusses a technique for modelling cargo port activity, it consists of an object-oriented approach of the port activity. A port modelling framework is implemented using the proposed technique and AnyLogic simulation software. Longo et al. [23] develop a simulation model to recreate the complexity of a medium-sized Mediterranean seaport and analyse the performance evolution of such system with particular reference to the ship turnaround time.

4. Analysis

This chapter consists of an explanation of the model build with a view of the databases needed to run the simulation. A validation of the model created, a fitting process to obtain a probabilistic function capable of generating new data and four case analysis and the conclusions obtained from them.

4.1 Model

An explanation of how the model is built and how it works is going to be presented in this section. The aim of that is making sure that anyone can understand the model, so improvements or changes can be made, if needed, to be able to perform future analysis on the container port terminal.

The simulation software AnyLogic, based in Java, with a free student licence has been used. Basically, the model reproduces the loading and unloading of containers from vessels in the container port terminal of Aarhus. In order to accomplish that, a discrete event model has been built using the Process Modeling Library from AnyLogic. This library contains all the tools to reproduce the container terminal layout, create the agents needed to run the model and build the processes flows.

4.1.1 Layout

Five space markups elements from the Process Modeling Library have been used to create the layout of the container port terminal:

- *Paths* to create the lines that Straddle Carriers will use to move through the container terminal.
- *Point Nodes* to create the points where the Quay Cranes will load and unload the containers.
- *Rectangular Node* to create the zone that define where the vessel is or where the Straddle Carriers park.
- *Attractors* to define the positions of the Straddle Carriers inside a *Rectangular Node.*
- *Pallet Rack* to create the storage zones for the containers.

To better understand all the elements of the layout, two figures are shown below:

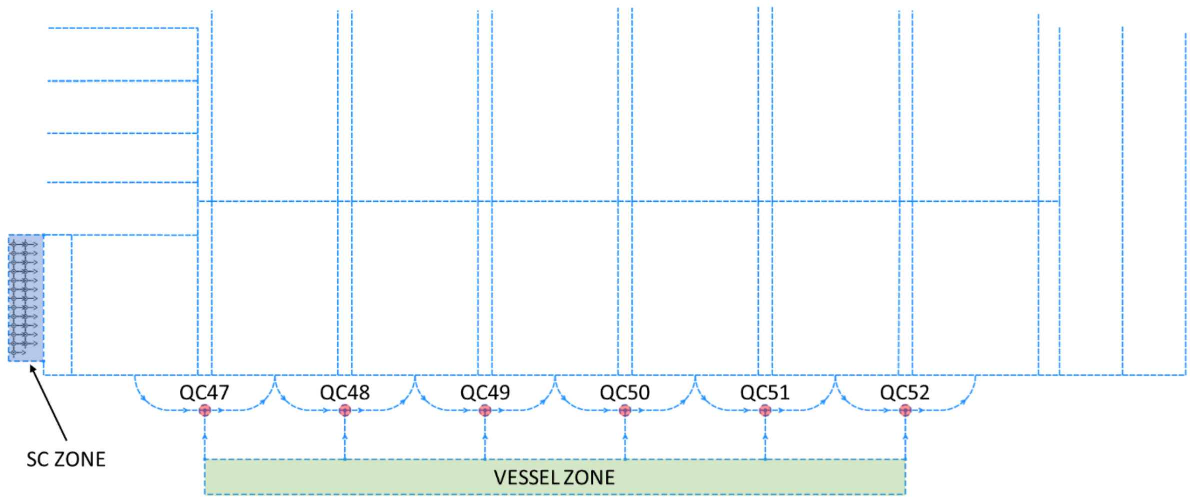


Figure 3. Network, SC Zone, Vessel Zone and QC points

In *Figure 3*, blue dashed lines define the paths that Straddle Carriers will use to move through the terminal. The Vessel Zone describe the region where vessels berth inside the terminal. The Straddle Carrier Zone (SC Zone) defines the region where all Straddle Carriers park, inside SC Zone some attractor points have been created to make sure that Straddle Carriers park with a certain order. Finally, Quay Cranes are represented as points where containers will be brought by Straddle Carriers during the loading process or they will appear there during the unloading process. To simplify the model, QC positions will remain fixed and those have been spread equidistantly in front of the main storage blocks (A – G).

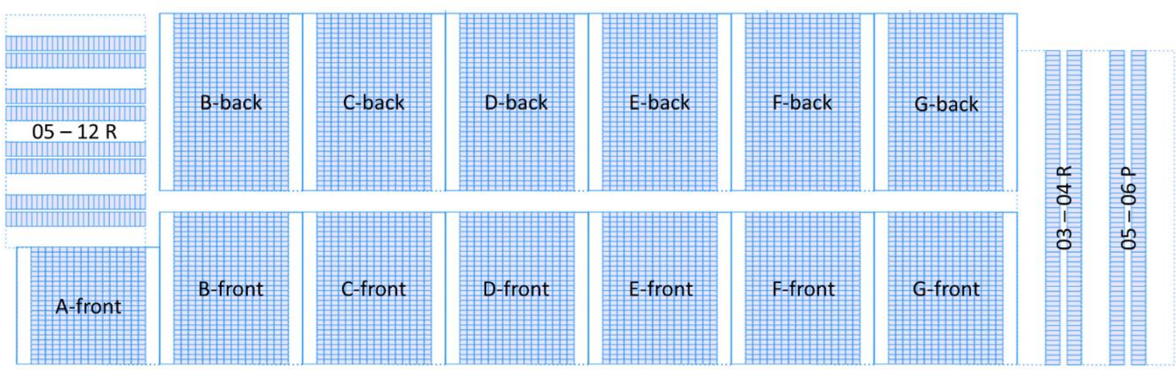


Figure 4. Storage Zone

In *Figure* 4, all the storage zone is represented. The space markup element Pallet Rack has been used in order to define it. This element creates simple storage zones and also allows to create complex storage zones, such as a block, by combining them. That is how the main blocks that are modelled have been created in AnyLogic. To simplify the model and the processes flows all the front blocks have been considered as a big one called $front$ as well as the back blocks called $back$, 05 – 12 R are also considered as one block called $northR$ while 03 – 04 R are considered as $southR$ block. To conclude, 05 – 06 P are considered as $blockP$. See *Figure 5* below for a better understanding of this simplification.

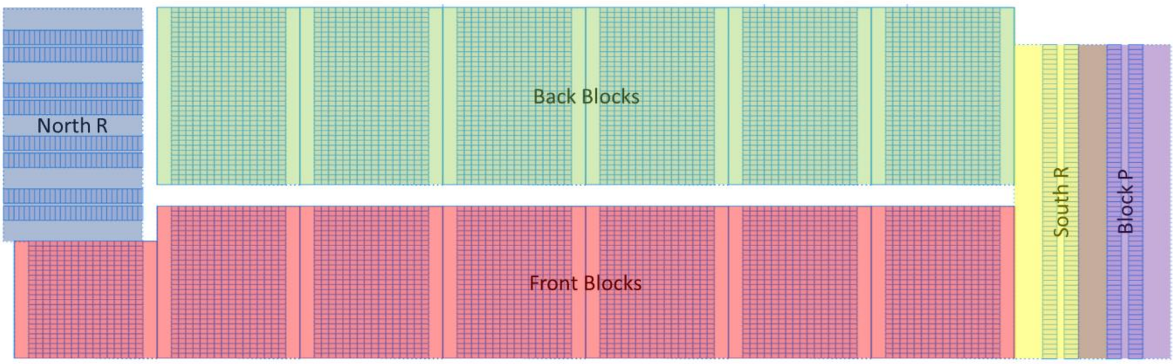


Figure 5. Block Simplification

To sum up, in *Figure 6* below all the layout is represented.

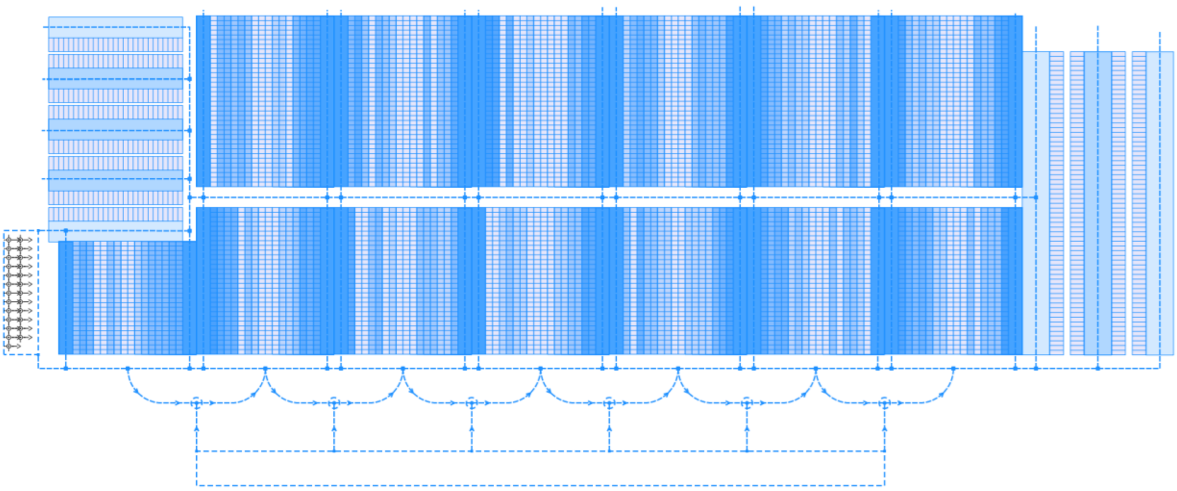


Figure 6. Model layout

4.1.2 Agents

Agents are main building blocks of AnyLogic model. Agent is a unit of model design that can have behaviour, memory (history), timing, contacts, etc. Within an agent you can define variables, events, statecharts, System Dynamics stock and flow diagrams, you can also embed other agents, add process flowcharts. You can define as many agent types in your model as there are different type of agents.

To perform the simulation three agents have been created. One for each type of container, 40 and 20 feet, and one for the Straddle Carriers.

4.1.2.1 Container40

The agent *Container40*, represents all the 40 feet containers that appear during the simulation. All 40 feet containers have the following parameters:

- Quaycrane – This parameter allows the model to know in which QC the container is unloaded or in which QC the straddle carriers have to bring the container if has to be loaded.
- Bay – This parameter allows the model to know in which bay of the block has the position this container.
- Row – This parameter allows the model to know in which row of the block has the position this container.
- Tier – This parameter allows the model to know in which tier of the block has the position this container.
- Pos – This parameter allows the model to know how to position the container in order to not invade other positions while 2D and 3D animation is running.
- Block – This parameter allows the model to know in which block the container has its position.
- Delay – This parameter allows the model to know the delay needed for the loading process.

4.1.2.2 Container20

The agent *Container20,* represents all the 20 feet container that appear during the simulation. All 20 feet containers have the following parameters:

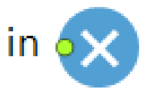
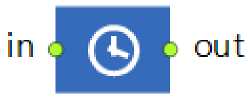
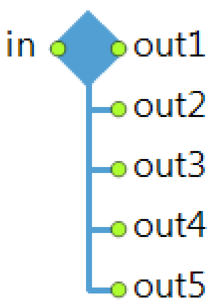
- Quaycrane – This parameter allows the model to know in which QC the container is unloaded or in which QC the straddle carriers have to bring the container if has to be loaded.
- Bay – This parameter allows the model to know in which bay of the block has the position this container.
- Row – This parameter allows the model to know in which row of the block has the position this container.
- Tier – This parameter allows the model to know in which tier of the block has the position this container.
- Block – This parameter allows the model to know in which block the container has its position.
- Delay – This parameter allows the model to know the delay needed for the loading process.

4.1.2.3 StraddleCarrier

The agent *StraddleCarrier,* represents all the Straddle Carriers used during the simulation. This agent has no parameters, it is just created to be able to call them as a resource pool and perform the movements of the containers.

4.1.3 Processes flows

For a better understanding of the processes flows, a brief explanation of all the blocks used for their design is presented below.

Block	Icon	Explanation
Source		It basically generates agents. It is usually a starting point of a process model. Arrivals can be defined in different ways, such as a rate, interarrival rate, arrival time in a database... The location of arrival is also defined here and can be any point of the network.
Sink		It disposes agents. It is usually an end point of a process model. Unless you use it, the agents would not be removed from the model and disposed. It is a must to finish a process, you cannot leave an unconnected port at the end of a process.
Delay		Delays agents for a given amount of time. The delay time is evaluated dynamically, may be stochastic and may depend on the agent as well as on any other conditions.
SelectOutput		Routes the incoming agents to one of the two output ports depending on (probabilistic or deterministic) conditions. The condition may depend on the agent as well as on any external factors. The agents spend zero time in SelectOutput.
SelectOutput5		This object routes the incoming agents to one of the five output ports depending on (probabilistic or deterministic) conditions. This block is used to sort agents according to certain criteria, to randomly split the agent flow... The agents spend zero time in SelectOutput5.

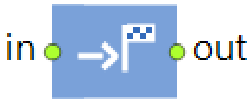
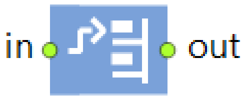
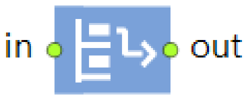
MoveTo		Moves an agent to a new location. If any resources are attached to the agent, they will move with it. The speed will be the agent speed regardless of the attached resources speed. The time spent by the agent in this object equals the length of the shortest route from the agent current location to the destination divided by the agent speed. The agent is animated moving along the route
RackStore		RackStore puts an agent into a cell of a given Pallet Rack or RackSystem. The agent is moved from its current location in the network to the cell location, optionally with the help of moving resources. A delay may be associated with putting an agent into a higher level. The cell may be specified explicitly as (row, position, level) or chosen automatically. If resources are used to move the agent, RackStore seizes them, brings to the agent location, attaches to the agent, moves the agent to the cell, executes (an optional) delay, and then releases the resources.
RackPick		RackPick removes an agent from a cell in the specified Pallet Rack or RackSystem and moves it to the specified destination location. This is optionally done with the help of moving resources, and, also optionally, a delay may be associated with picking the agent. The delay may depend on the level of the agent. If resources are used to move the agent, RackPick seizes them, brings to the agent cell location, executes (an optional) delay, attaches resources to the agent, moves the agent to the destination, and then releases the resources.

Table 2. Explanation of AnyLogic blocks used in the model

4.1.3.1 Unloading Process

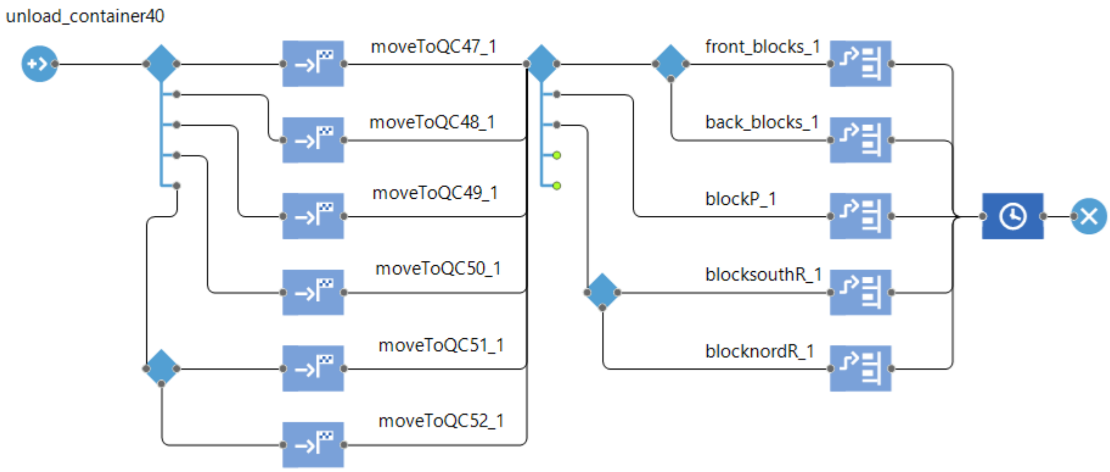


Figure 7. Unloading 40feet flow process

A *Source* block is needed to generate containers, they are generated following the arrival times defined in a Database. Real data from unloading and loading processes has been used during all the design and validation of the model, that means that these times are real times and correspond to the times when a Straddle Carrier driver accepts the task of moving the container. Moreover, a probabilistic distribution has been found (see *section 4.4*) fitting these times to be able to run future simulations were times are not available. Furthermore, thanks to this block, the container also gets all the values for the different parameters that it has from the Database used to read the arrival times.

Once a container is generated, it passes through a *SelectOutput* block that redirects the container to the *MoveTo* block that will move the container to the assigned QC. This block basically reads the *Quaycrane* parameter from the container agent and send it to the convenient block. After that, the container passes through another *SelectOutput* block that sends it to the *RackStore* block in charge of moving the container in the assigned storage block.

In the same way as the previous *SelectOutput* block, this one reads the *Block* parameter from the container agent to redirect it to the convenient block. As seen in *Figure 7* there are two *SelectOutput* blocks more after the one with five ports. The one on the top is needed to differentiate between front and back blocks (A – G Blocks), it reads the *Row* parameter to decide. On the other hand, the one on the bottom is needed to differentiate *northR* from *southR*, it also reads the *Row* parameter to decide.

About the *RackStore* block, this one reads the parameters *Bay*, *Row* and *Tier* from the container agent to exactly know where it is going inside the storage yard. It also seizes one of the SC from the resource pool in order to perform the task of moving the container from the QC to the assigned position inside the storage yard.

Finally, a *Delay* block is added to make sure that the containers unloaded remain in the model to basically see in which positions the containers have been stored. A *Sink* block is always needed to end the process as mentioned in section 5.1.3. See below the *Figure 8* that corresponds to the unloading process but for 20 feet containers, it is completely the same as the 40 feet one but using a different Database that contains the 20 feet containers to unload.

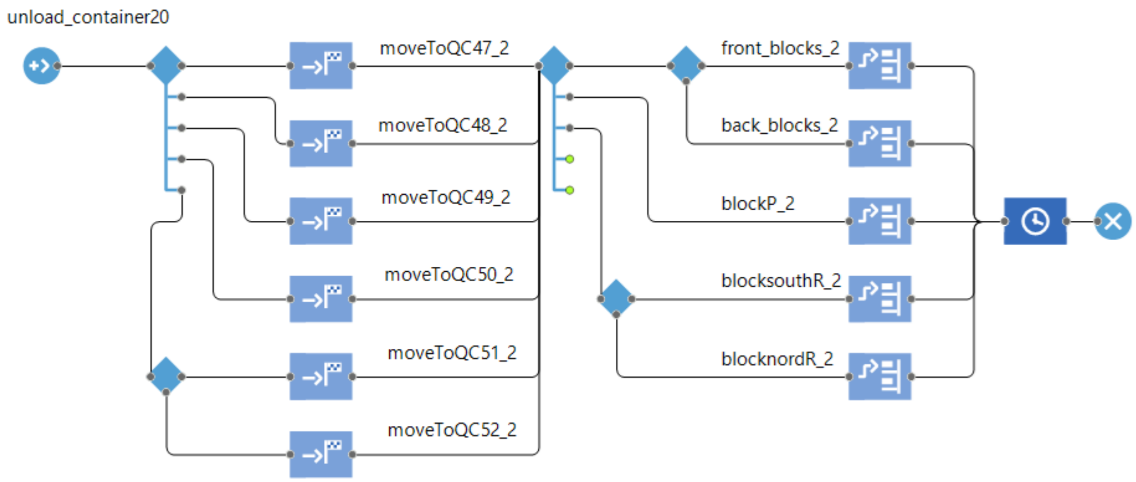


Figure 8. Unloading 20feet flow process

4.1.3.2 Loading Process

First of all, in order to perform the loading process, containers must be already placed in their respective positions inside storage yard so *RackPick* block can call them without crashing the simulation, otherwise the model would try to call an empty position and an error message would appear. To accomplish this the first part of the process consists of introducing all containers in their positions. See *Figure 9* below:

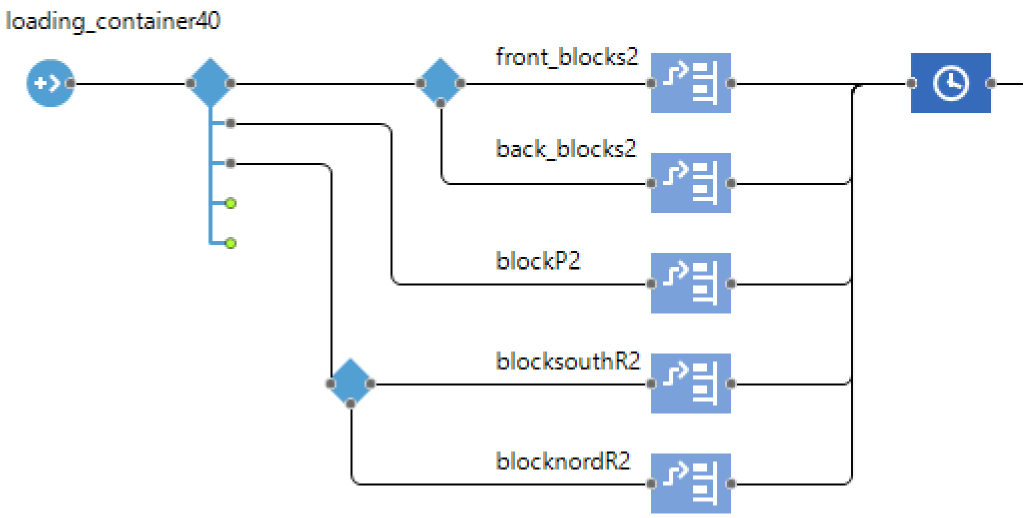


Figure 9. Loading 40feet flow process, introducing containers

In the same way as the unloading process, loading process starts with a *Source* block in order to generate all container agents. However, in this process the arrivals are not defined by arrival times in a Database because, as explained in the previous paragraph, we are placing all containers in their positions to be able to call them in a near future to load them. It is like setting the initial conditions of our simulation to be able to run it. It is set to generate one container each one millisecond getting all the parameters needed from the database.

After that, a *SelectOutput* system is added to redirect the container to the assigned block, this system is exactly the same as in the unloading process. Then, the *RackStore* block is in charge of sending each container to its place inside the storage yard, to perform that this block reads the parameters *Bay, Row* and *Tier* from the container agent. To clarify, this part of the process does not seize any Straddle Carrier and the speed used to perform it is high enough to not affect the overall simulation.

Once all the containers have been placed in their respective locations, a *Delay* block is added to wait the exact amount of time that the container waits in the yard storage until it can be loaded. A subtraction of the dispatch time with the berth start time has been computed to obtain the delay needed. This delay is inside the Database and is read and stored in the *Delay* parameter when the container is generated in the *Source* block.

When the delay finishes the container passes through two *SelectOutput* systems, the first one is in charge of redirect the container to the QC assigned for loading and the second one, same as the one in the introducing process (*Figure 10)*, is in charge of redirect the container to the *RackPick* block that corresponds its position. See in *Figure 10* below the flow process, note that there are four more of the second *SelectOutput* systems in the middle gap.

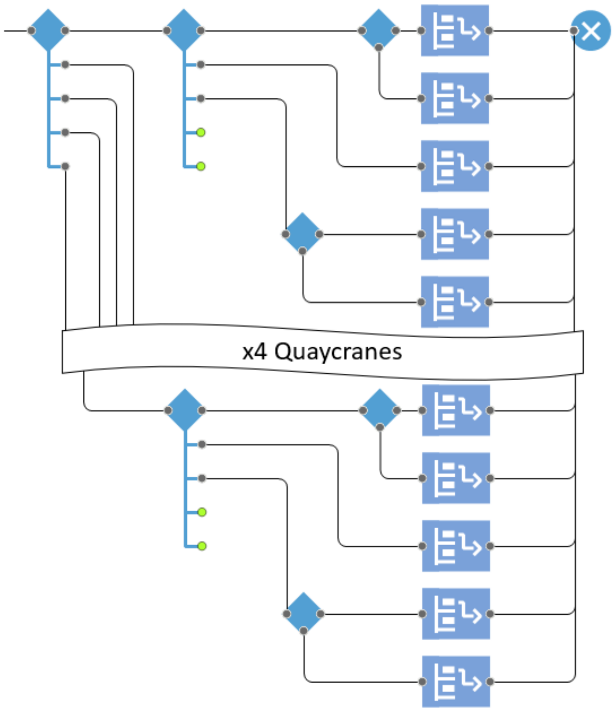


Figure 10. Loading 40feet flow process, picking up containers

This flow process could be simplified if Anylogic could allow to select the QC depending on one parameter in the *RackPick* block. But since is not possible in this actual version more blocks are needed to accomplish it.

To conclude, as seen in the unloading process, a *Sink* block is always needed to end the process. There is a little variation in the process of loading 20 feet containers, a five seconds *Delay* block is added after the *Source* block to avoid creating 40 feet containers and 20 feet containers at the same time because it crashes the simulation.

4.1.4 Model Limitations

In this section the limitations of the model created will be presented, just to clarify what is modelled and what is not. See in *Figure 11* below the yard layout of the container port terminal that has been modelled.

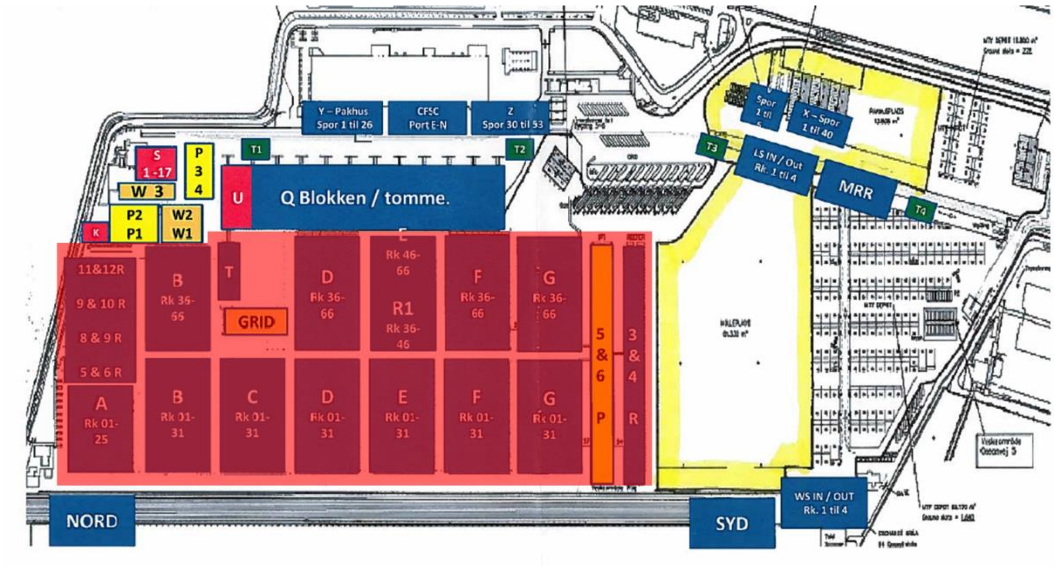


Figure 11. Aarhus port terminal yard layout

Just the red part is actually modelled, the other parts are not included, that means that all the movements that go or come from that other zones are deleted from the databases before running the simulations. The QCs positions have been fixed and spared through the quayside, normally QCs assigned to a berth move to the zone that the vessel is going to be.

The process of reshuffling inside yard blocks is also not modelled, that really affects in the unloading processes for big vessels. Reshuffling happens when one movement is done, normally in a position near to the QC and then some time later it is moved to the final position. That allows a faster unload of the vessel so berth turnaround time is minimum. Another thing that has not been modelled are the possible breaks during the berth or the possible incidents that may occur.

The last thing is about *dispatch* times in the databases. Two equal *dispatch* times cannot be in the same database because AnyLogic will understand that these two container agents that is generating are the same but they do not. To solve this, if there is the existence of two equal times a second is added in one of the two *dispatch* times.

4.2 Databases

In this section the Databases needed to run the simulation will be explained. There are four Databases in total:

- DIS_40feet
- DIS_20feet
- LOAD_40feet
- LOAD_20feet

The two first ones are used for the unloading process, and as it can be seen one is for the 40 feet containers and the other one for the 20 feet containers. The same happens with the last two, they correspond to the loading process and one is for the 40 feet containers and the other one for the 20 feet containers. These Databases can come basically from two sources.

The first one is called Sparcs and it is a Database from APM Terminals that contains all the movements performed in all vessels. This Database contain a lot of information for each movement, for example if it is a load or an unload movement, dispatch time and complete time, the SC used, from where it comes and where it goes... Having this big Database has been useful for validating the model and during the design of it. But this data is generated while the operations are being done so for future analysis all this information will be missing, therefore is not the source that will be used for future analysis.

The second source are called the Benchmarks files, there is one file for each vessel and these files are generated before the vessel berths in the container port terminal. Each of these files starts with the following information:

- ShipID – The call Index for this specific berthing.
- BerthingDateTime – Date of the berthing and the minute of the day at which the operations start.
- TotalToLoad – Total number of containers to load in the vessel.
- TotalToUnload – Total number of containers to unload from the vessel.

Afterwards each container movement has a line for itself, first all the containers to load and after that all the containers to unload. Each line contains the following information separated by a single space:

- Container ID – The identification number of the container
- Container Class – Categorisation of the container type
- Container Weight – The weight of the container
- Origin – Position where the container is moved from (this position may be either on the yard or on the vessel)
- Destination – Position where the container is moved to (this position may be either on the yard or on the vessel)
- Quay crane – QC used to perform the load or unload of the container

4.2.1 DIS_40feet and DIS_20feet

Since both databases have practically the same info they will be presented together. The only difference is that the one for 20 feet containers does not contain the information *pos*. See below the information they contain:

- Id – Index
- Container number – The identification number of the container
- Dispatch – Corresponds to the time when a SC driver accept the task of moving the container from the QC assigned to the destination position
- Quay crane – QC used to perform the unload of the container
- To – Position of the yard where the container is moved to
- Row – Row of the block in which the container has its destination
- Bay – Bay of the block in which the container has its destination
- Block – Block in which the container has its destination
- Tier – Tier of the block in which the container has its destination
- Pos – Position that the container has to adopt to fit in its destination. This information is just contained in the 40 feet database

4.2.2 LOAD_40feet and LOAD_20feet

Like the unloading databases, both databases have practically the same info so they will be presented together. The only difference is that the one for 20 feet containers does not contain the information *pos*. See below the information they contain:

- Id – Index
- Container number – The identification number of the container
- Dispatch – Corresponds to the time when a SC driver accept the task of moving the container from the QC assigned to the destination position
- Delay – Delay time (in seconds) that the container remains in its position before moving it to the QC assigned. It is calculated by subtracting the *dispatch* time from the berthing start time and converting it to seconds
- Quay crane – QC used to perform the unload of the container.From – Position of the yard where the container is moved from.
- Row – Row of the block in which the container has its origin
- Bay – Bay of the block in which the container has its origin
- Block – Block in which the container has its origin
- Tier – Tier of the block in which the container has its origin
- Pos – Position that the container has to adopt to fit in its origin. This information is just contained in the 40 feet database

4.3 Model validation

In this section the model will be validated with real data. That means that similar results are expected from the simulation as it happens in the reality. Basically, a comparison on the time needed for all the movements in real life and the one needed in the simulation will be made.

Data of four vessels have been cleaned to run the model and be able to perform the validation. Each vessel data contains a column named *complete* with the times when a SC driver finish a task (task means a loading or unloading movement). A calculation of the total time needed in real to perform all the movements can be performed subtracting the latest *complete* time from the operations starting time. In the simulation, the finish time from the last movement is also saved to compute the total time needed in the simulation to finish all the movements. See below in *Table 3* the results:

Vessel	Starting time	Finish Time		Total Time [sec]		Absolute Error [sec]	Relative Error
		Real	Simulation	Real	Simulation		
09097AAT	07:00:00	12:21:05	12:20:03	19265	19203	-62	-0,32%
90505AAT	13:00:00	21:26:31	21:24:32	30391	30272	-119	-0,39%
09506AAT	13:20:00	20:46:45	20:43:37	26805	26617	-188	-0,70%
09710AAT	07:00:00	12:27:50	12:25:43	19670	19543	-127	-0,65%

Table 3. Comparison of the real time and simulation time needed for vessel

The absolute error, known as the difference between the measured or inferred value of a quantity $x_0$and its actual value x ($\Delta x \equiv x_0 - x$), and the relative error, known as the ratio of the absolute error of a measurement to the measurement being taken ($\delta x = \Delta x / x$), have been also included.

It can be seen in *Table 3* that for four different vessels the time needed in the simulation is always less than the needed in real life. But, at least for these four vessels, the difference is always less than 1%. This difference is mainly due to two reasons; the first one is that in each vessel some movements have been removed because they were going or coming from positions in the yard that are not included in the model, and the second one is because the model does not take into account possible breaks that SC drivers may take. Although these reasons, the model behaves as expected and simulates the real world with less than 1% of relative error.

Since the four vessels are simulated with less than 1% of relative error looking at the total time needed to perform all the movements, it can be said that the model reproduces the reality and therefore it is validated and suitable to use it because will give similar results to reality.

4.4 Data fitting

In order to provide a good tool to decide the amount of SCs needed for future vessels, dispatch times must be generated to run future simulations. Hence, a probabilistic distribution to generate these times must be found. In this chapter, the process followed to find this probability distribution will be explained and the results will be presented.

To be more precise, all the loads and unloads from a QC have been sorted by the time the task is started (dispatch time) and times between tasks have been computed. The probabilistic distribution must fit these times to be able to generate new ones randomly. Data from five different vessels has been used to perform the fitting, each vessel worked with two different QCs so in total ten different QCs.

A full fitting process will be presented below using data from one QC, R Software has been used to perform this task. The data used contains 248 observations. Furthermore, results from the other nine QCs will be also shown and the selected probabilistic distribution will be chosen.

4.4.1 Data validation

A dataset is valid when it fulfils two requirements, on one hand all its observations or registrations must be independent one from each other. To prove it, the R-command 'acf' or autocorrelation function shows the correlation between values of the process at different times, as a function of the two times or of the time lag. Thus, autocorrelation plot is a useful tool for checking randomness in a data set. If it is random, such autocorrelations should be around zero for any time-lag separations.

```
> ## Validation of the dataset
> ## Correlation of the data values
> ac <- acf(dades$Times, main="Autocorrelation function");ac

Autocorrelations of series 'dades$Times', by lag

     0      1      2      3      4      5      6      7      8      9     10     11     12 
 1.000 -0.135  0.237 -0.006  0.239  0.031  0.112  0.051  0.192  0.025  0.204  0.011  0.140 
    13     14     15     16     17     18     19     20     21     22     23 
 0.047  0.053  0.077 -0.030  0.162 -0.070  0.067 -0.013  0.090  0.000  0.021 
```

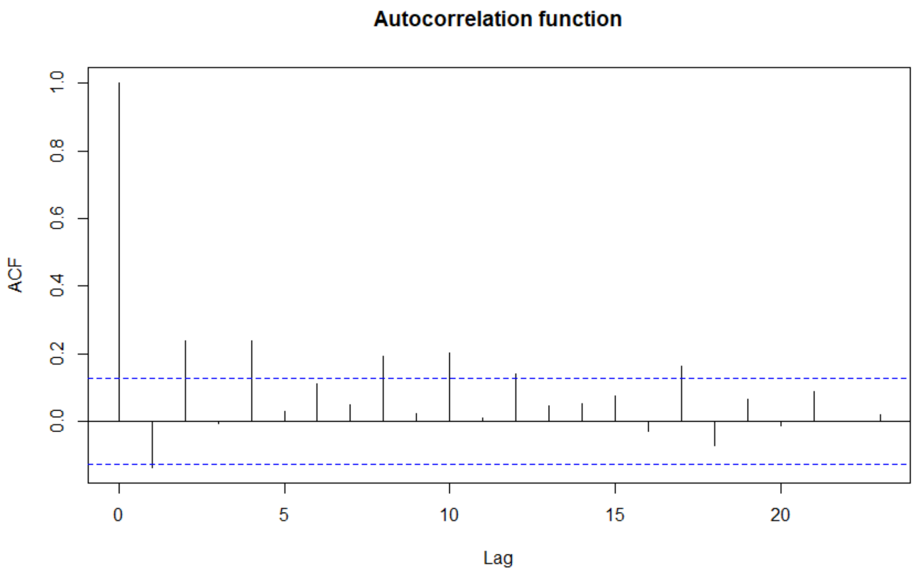


Figure 12. Acf of the dataset

As it can be seen, apart from the first 0-lag correlation that is always one, all other correlations are approximately zero showing the randomness behaviour of the dataset.

On the other hand, the second requirement is that the dataset must not have more than certain number of outliers; i.e. more than 1% of extreme outliers and 5% of mild outliers. The reason is that outliers take extreme values to the inner variability of the dataset or due to a measurement error. Taking into account these extreme values, the whole study would be distorted and consequently the result and conclusions obtained would be wrong. For that reason, is a matter of great importance to check the presence of outliers.

A boxplot of the dataset can show the outliers, see in *Figure 13* below:

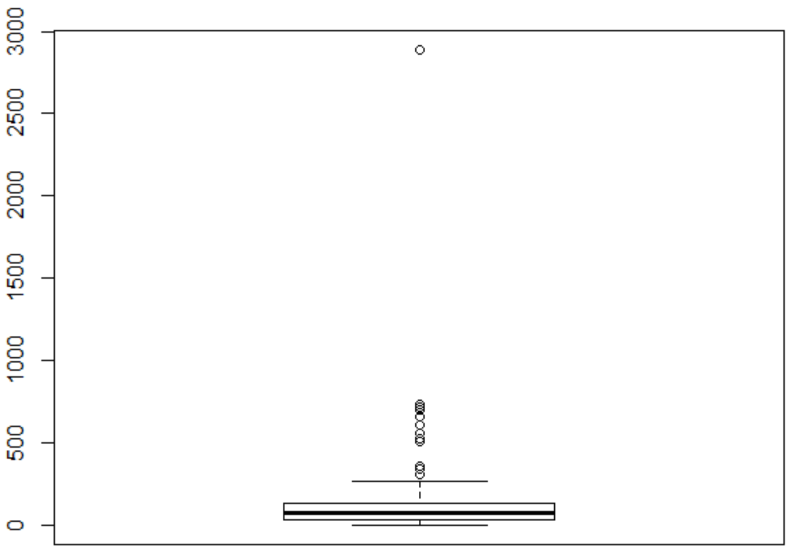


Figure 13. Boxplot of the dataset

In the concerned case, the only possible outliers are found towards positive x which make sense because negative times are not possible. For that reason just the upper zone is checked:

$$mild\ outliers = Q3 + 1{,}5IQR$$

$$extreme\ outliers = Q3 + 3IQR$$

Being Q3 the third quartile and IQR the interquartile range. Now the number of mild and extreme outliers will be computed to check that are below 5% and 1% respectively.

```
> ## Identifiction of outliers
> #Mild outliers
> mild <- quantile(dades$Times,0.75)+1.5*IQR(dades$Times);mild
    75%
278.125
> mo <- which(dades$Times > mild);mo
 [1]  50 152 162 181 237 242 243 244 245 246 247 248
> lenmo <- length(mo);lenmo
[1] 12
> #Extreme outliers
> extreme <- quantile(dades$Times,0.75)+3*IQR(dades$Times);extreme
   75%
424.75
> eo <- which(dades$Times > extreme);eo
[1] 152 237 242 243 244 245 246 247 248
> leneo <- length(eo);leneo
[1] 9
```

The dataset contains 248 observations meaning in one hand that there are 12 out of 248 mild outliers and consequently less than the 5%. But on the other hand, there are 9 extreme outliers out of 248 that correspond more than the 1%. That means extreme observations have to be removed to achieve these thresholds and continue with the fitting. Only one extreme outlier has to be removed every time and check again everything because the thresholds change every time one observation is removed. After this iterative process, 241 observations remain and there are less than 5% of mild outliers and 1% of extreme outliers.

4.4.2 Basic dataset information: Graphical summary

After validating the dataset, the first step is to get its basic information: its shape and range of values. Depending on it, a first discretization will be done as each probabilistic distribution is characterized differently; i.e. is not the same negative than positive values, integer or float values, as well as how frequent the data is recorded. Therefore, a histogram of the assigned data is depicted as it is shown in *Figure 14*.

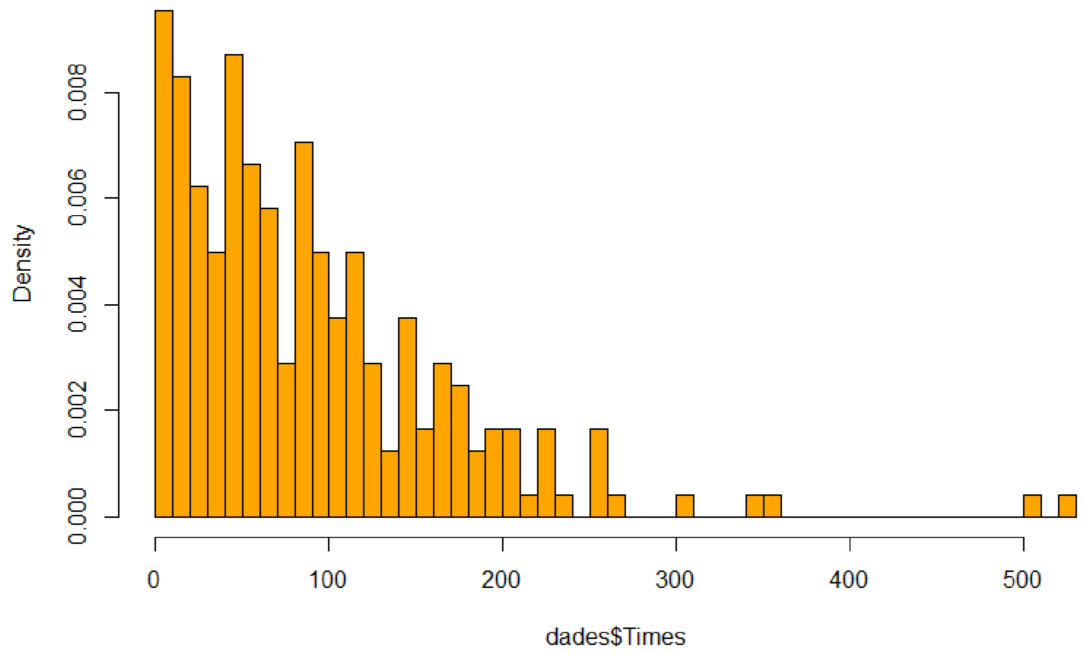


Figure 14. Histogram of the data

All values are positive, integer and gathered in the left-hand side. There is not any visual shift to take into account. To support the observations made at a glance, some numeric values are computed through the R-command 'summary'.

```
> summary(dades$Times)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
    1.0    32.0    70.0    90.3   124.0   527.0 
```

The R-command 'summary' also provides information about first, second and third quantile as well as the data mean and the maximum value of the dataset. Nothing special can be said from these values.

4.4.3 Selection of candidate distributions and Estimation of model parameters (Maximum Likelihood Estimation)

The candidates that are going to be used with all the QC datasets are based on the previous observations and the computation of the coefficients of variation of each dataset. As in all datasets the coefficient of variation has been less than one or approximately one, the candidates that have been selected for the study are the following ones:

- Gamma
- Weibull
- Exponential

Specifically, for this dataset the coefficient of variation is the following one:

```
> CV <- mysd/mymean; CV
[1] 0.8805189
```

Once the candidates that can fit have been decided, the R-command 'fitdistr' is used to estimate the parameters of the model (Maximum Likelihood Estimation).

- For Gamma distributions, shape and rate are estimated.
- For Weibull distributions, shape and scale are estimated.
- For Exponential distributions, rate is estimated.

```
> gamres<-fitdistr(dades$Times,"gamma")
> gamres$estimate # MLikelihood estimate
    shape      rate
1.2659795 0.0140199

> webres<-fitdistr(dades$Times,"weibull",lower=0.0001)
> webres$estimate # MLikelihood estimate
    shape     scale
 1.156329 95.072028

> expres<-fitdistr(dades$Times,"exponential")
> expres$estimate # MLikelihood estimate
      rate
0.01107435
```

4.4.4 Validation of the candidates

There are three ways of validating if the fitted distributions really reproduce our dataset: visual validation, QQ Plots and Goodness of Fit. Visual validation is a good choice to start because one can discard some candidates that obviously do not follow the dataset.

Secondly, QQ Plots are graphs that represent a comparison between the quantiles of the sample with the quantiles of the candidate probabilistic function. Thus, a perfect match would be depicted by a perfect straight line or regression line.

Finally, two Goodness of Fit tests can be performed, the Kolmogorov-Smirnov Test and the Pearson Test. These two last ones have a greater importance in the validation since they are methods of acceptance or rejection of a null hypothesis, using the P-value approach.

4.4.4.1 Visual Validation

A visual validation is the perfect first choice of validation because it allows one to discard obvious candidates that do not fit the dataset. It is a good method for obvious candidates that not follow the dataset. But on the other hand, for final decisions it falls on one's opinion when selecting one or another candidate, so in the end other methods are needed to select the chosen candidate.

```
> # Gamma distribution
> hist(dades$Times, freq = F, breaks = 50, col = "orange", main = "My data and
Gamma distribution candidate curve")
> curve(dgamma(x,shape=gamres$estimate[1],rate=gamres$estimate[2]),col="black",
lwd=3,add=T)
```

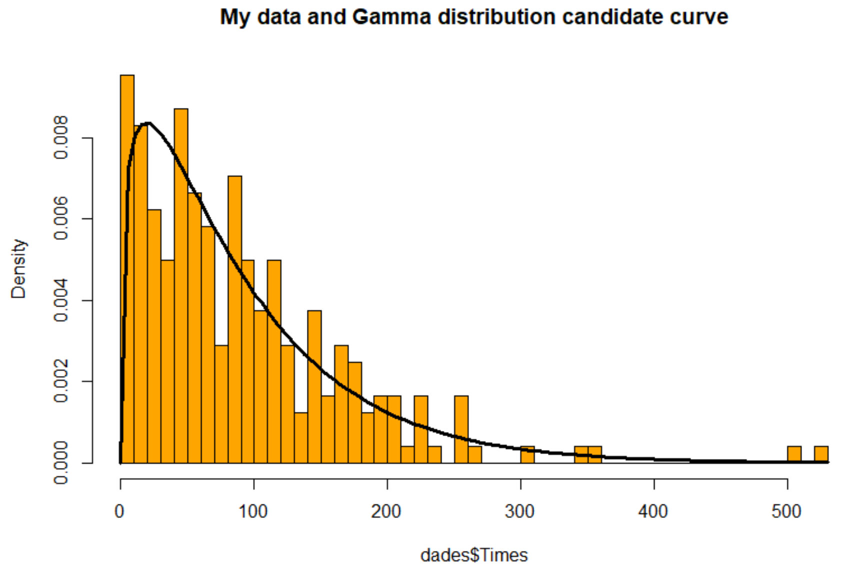


Figure 15. Visual comparison: Dataset and Gamma distribution candidate

```
> # Weibull distribution
> hist(dades$Times, freq = F, breaks = 50, col = "orange", main = "My data and
Weibull distribution candidate curve")
> curve(dweibull(x,shape=webres$estimate[1],scale=webres$estimate[2]),col="blac
k",lwd=3,add=T)
```

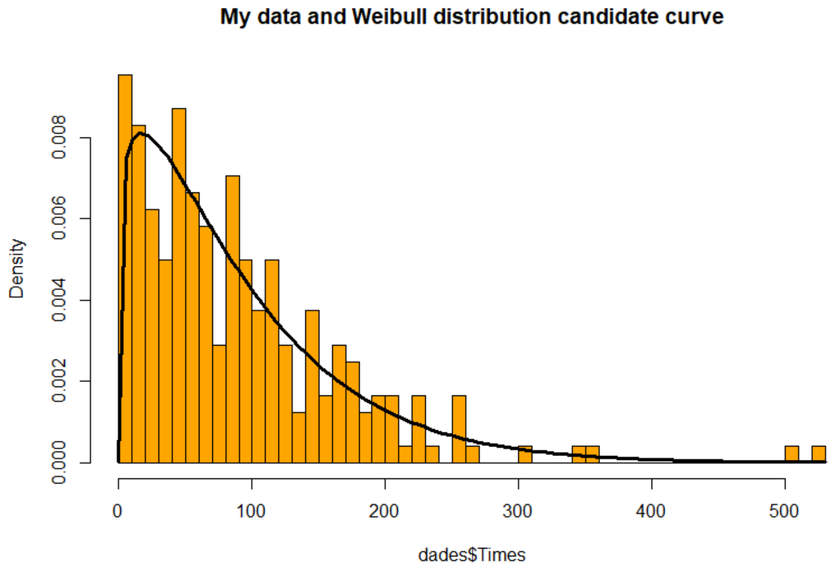


Figure 16. Visual comparison: Dataset and Weibull distribution candidate

```
> # Exponential distribution
> hist(dades$Times, freq = F, breaks = 50, col = "orange", main = "My data and
Exponential distribution candidate curve")
> curve(dexp(x,rate=expres$estimate),col="black",lwd=3,add=T)
```

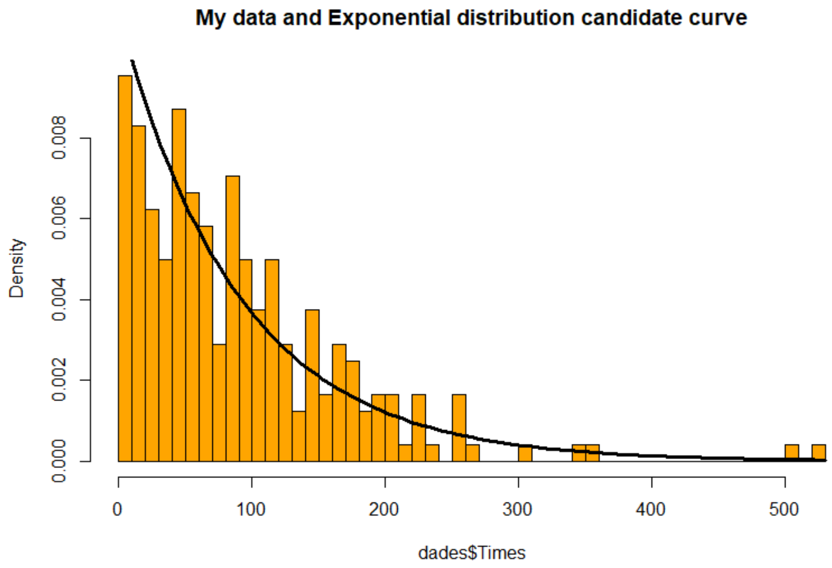


Figure 17. Visual Comparison: Dataset and Exponential distribution candidate

By just looking at the graphs, no conclusion can be drawn because there is no obvious candidate that does not fit the dataset and could be rejected and moreover all three candidates could fit the dataset.

4.4.4.2 QQ Plots

QQ Plots graph compare the quantiles of the dataset with the ones of the candidate probabilistic distribution. Obtaining a good regression line would mean that the candidate fits the dataset.

```
> # Gamma: First estimation of model parameters based on QQplot
> xx<-qgamma(ppoints(dades$Times),gamres$estimate[1])
> yy<-sort(dades$Times)
> mm<-lm(yy~xx)
> plot(qgamma(ppoints(dades$Times),gamres$estimate[1]),sort(dades$Times),col="b
lack", main="QQ-Plot Gamma")
> abline(mm,col="red",lwd=2,lty=2)
> summary(mm)

Call:
lm(formula = yy ~ xx)

Residuals:
    Min      1Q  Median      3Q     Max 
-37.259  -2.944   1.112   2.728 102.669 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   1.0298     0.9324   1.104     0.27    
xx           70.5962     0.5529 127.691   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.576 on 239 degrees of freedom
Multiple R-squared:  0.9856,	Adjusted R-squared:  0.9855 
F-statistic: 1.631e+04 on 1 and 239 DF,  p-value: < 2.2e-16
```

The Multiple R-squared value, also known as the coefficient of determination, is a measure of how close the data is to the fitted regression line, in this case, the Gamma probabilistic distribution. The value can range from 0 to 1 depending whether the model explains none or all the variability of the response data around its mean, respectively. Consequently, in general terms, the higher the Multiple R-squared, the better the model fits the dataset.

The Multiple R-squared value obtained from QQ-plot is a fairly high value. Consequently, it cannot be stated that the dataset does not follow a Gamma distribution. See the graph in *Figure 18 below, it* represents the QQ Plot graph for the Gamma distribution candidate, most of the data follows the red line as expected with 0.9856 Multiple R-squared value.

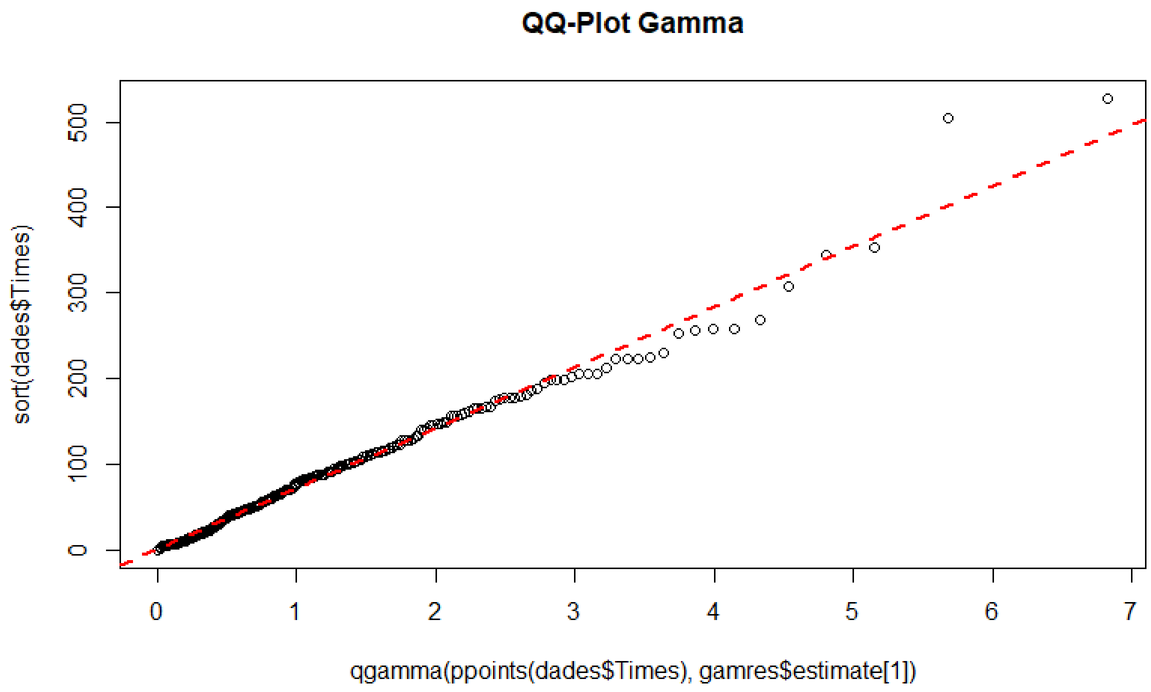


Figure 18. QQ-Plot of Gamma distribution candidate

Now, the Weibull and Exponential QQ Plots will be presented followed by a final statement on what it is concluded from them.

```
> # Weibull: First estimation of model parameters based on QQplot
> xx<-qweibull(ppoints(dades$Times),webres$estimate[1])
> yy<-sort(dades$Times)
> mm<-lm(yy~xx)
> plot(qweibull(ppoints(dades$Times),webres$estimate[1]),sort(dades$Times),col=
"black", main="QQ-Plot Weibull")
> abline(mm,col="red",lwd=2,lty=2)
> summary(mm)

Call:
lm(formula = yy ~ xx)

Residuals:
    Min      1Q  Median      3Q     Max
-33.667  -1.862   0.775   2.211 113.942

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.9671     1.0096  -0.958    0.339
xx           96.1453     0.8055 119.354   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10.23 on 239 degrees of freedom
Multiple R-squared:  0.9835,    Adjusted R-squared:  0.9834
F-statistic: 1.425e+04 on 1 and 239 DF,  p-value: < 2.2e-16
```

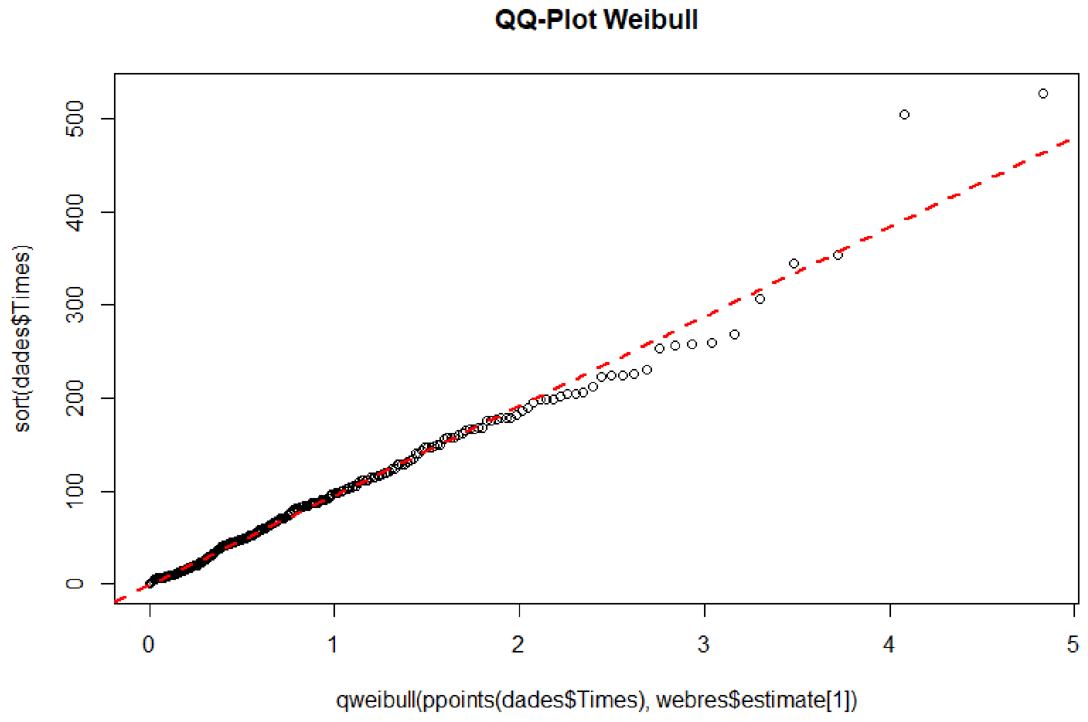


Figure 19. QQ-Plot of Weibull distribution candidate

```
> # Exponential: First estimation of model parameters based on QQplot
> xx<-qexp(ppoints(dades$Times),expres$estimate)
> yy<-sort(dades$Times)
> mm<-lm(yy~xx)
> plot(qexp(ppoints(dades$Times),expres$estimate),sort(dades$Times),col="black"
, main="QQ-Plot Exponential")
> abline(mm,col="red",lwd=2,lty=2)
> summary(mm)

Call:
lm(formula = yy ~ xx)

Residuals:
    Min      1Q  Median      3Q     Max 
-42.300  -6.181   2.244   5.377  90.466 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 10.958436   0.935257   11.72   <2e-16 ***
xx           0.879907   0.007364  119.48   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10.22 on 239 degrees of freedom
Multiple R-squared:  0.9835,	Adjusted R-squared:  0.9835 
F-statistic: 1.428e+04 on 1 and 239 DF,  p-value: < 2.2e-16
```

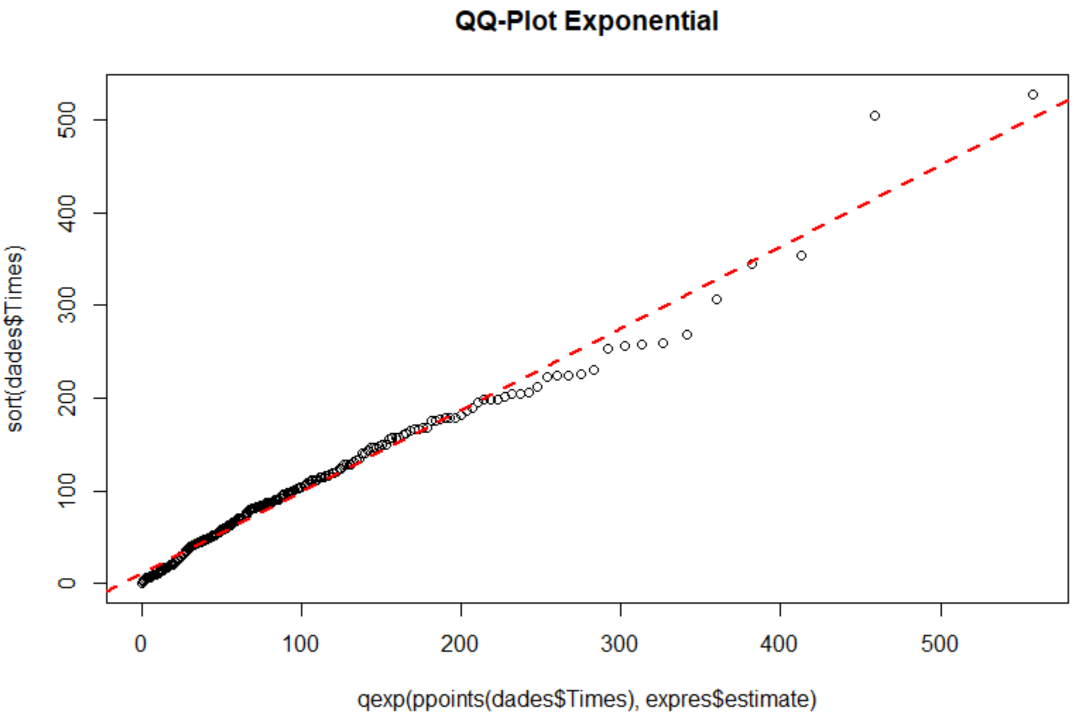


Figure 20. QQ-Plot of Exponential distribution candidate

In both cases, for the Weibull and Exponential candidates, the Multiple R-squared is 0,9835. A high value that means that the candidates may fit the dataset, also the QQ Plot graphs depict the same. To conclude, all three probabilistic distributions are still potential candidates, the final decision will be taken with Goodness of Fit.

4.4.4.3 Goodness of fit: Kolmogorov-Smirnov Test (K-S) and Pearson Test (Chi-squared)

Both tests are based in acceptance or rejection of null hypothesis, the null hypothesis assumed by R when implementing these tests is that the dataset follows the candidate distribution. Consequently, if a P-value greater than 0,05 is obtained, the null hypothesis should not be rejected. On the other hand, if the P-value is much lower than 0,05, the null hypothesis can be rejected stating that the sample considered does not follow the candidate distribution.

Kolmogorov-Smirnov Test is another way to verify whether two samples are drawn from identical distributions.

K-S Test for the Gamma distribution

```
> # Gamma distribution
> ks.test(dades$Times,"pgamma",shape=gamres$estimate[1],rate=gamres$estimate[2])

        One-sample Kolmogorov-Smirnov test

data:  dades$Times
D = 0.041057, p-value = 0.8113
alternative hypothesis: two-sided
```

K-S Test for the Weibull distribution

```
> # Weibull distribution
> ks.test(dades$Times,"pweibull",shape=webres$estimate[1],scale=webres$estimate[2])

        One-sample Kolmogorov-Smirnov test

data:  dades$Times
D = 0.033229, p-value = 0.9529
alternative hypothesis: two-sided
```

K-S Test for the Exponential distribution

```
> # Exponential distribution
> ks.test(dades$Times,"pexp",rate=expres$estimate)

        One-sample Kolmogorov-Smirnov test

data:  dades$Times
D = 0.075718, p-value = 0.1261
alternative hypothesis: two-sided
```

All three tests give a P-value greater than 0,05. Therefore, the null hypothesis cannot be rejected and consequently, by means of Kolmogorov-Smirnov Test, the dataset could follow a Gamma, Weibull or Exponential probabilistic distributions.

Person Test, most commonly known as Chi-squared Test, also verifies if the dataset follows a concerned probabilistic distribution. It measures the difference between the expected and the real number of observations per interval considering the candidate distribution. Therefore, the null hypothesis of the test considers that the sample follows the candidate distribution. Consequently, as in the Kolmogorov-Smirnov Test, if a P-value greater than 0,05 is obtained, the null hypothesis should not be rejected and if it is lower than 0,05, there are evidences to reject the null hypothesis.

The number of breaks used for this test must be determined before performing it. This number must be less than the square root of our amount of values in our dataset. In this case:

$$Breaks < \sqrt{Amount\ of\ values\ in\ dataset}$$

$$Breaks < \sqrt{241}$$

$$Breaks < 15{,}52$$

Fifteen breaks are going to be used to perform the Chi-squared Test for this dataset.

```
> obs <- 241
> nu <- 15
```

Chi-squared test for the Gamma distribution candidate

```
> # Gamma distribution
> sequence<-seq(0,1,by=1/nu)
> percgamma<-qgamma(sequence,shape=gamres$estimate[1],rate=gamres$estimate[2])
> percgamma[nu+1]<-max(dades$Times)
> xx<-cut(dades$Times,percgamma)
> ss<-as.vector(table(cut(dades$Times,percgamma))) # count number of observations i
n sample per interval
> a <- obs/nu
> s <- sum(((ss-a)^2)/a)
> 1-pchisq(s,nu-1)
[1] 0.3293361
```

Chi-squared test for the Weibull distribution candidate

```
> # Weibull distribution
> sequence<-seq(0,1,by=1/nu)
> percweb<-qweibull(sequence,shape=webres$estimate[1],scale=webres$estimate[2])
> percweb[nu+1]<-max(dades$Times)
> xx<-cut(dades$Times,percweb)
> ss<-as.vector(table(cut(dades$Times,percweb))) # count number of observations in
sample per interval
> a <- obs/nu
> s <- sum(((ss-a)^2)/a)
> 1-pchisq(s,nu-1)
[1] 0.7425217
```

Chi-squared test for the Exponential distribution candidate

```
> # Exponential distribution
> sequence<-seq(0,1,by=1/nu)
> percexp<-qexp(sequence,rate=expres$estimate)
> percexp[nu+1]<-max(dades$Times)
> ss<-as.vector(table(cut(dades$Times,percexp))) # count number of observations in
sample per interval
> xx<-cut(dades$Times,percexp)
> a <- obs/nu
> s <- sum(((ss-a)^2)/a)
> 1-pchisq(s,nu-1)
[1] 0.105632
```

Again, in all three cases the P-value is greater than 0,05 meaning that the three candidates fit the dataset. Nevertheless, the Weibull probabilistic distribution is the one with the highest P-values making it the more suitable for fitting the dataset. If it was the only QC studied to obtain a probabilistic distribution function to generate new data, the Weibull distribution with shape 1,1563 and scale 95,0720 would have been the one chosen. But since a probabilistic distribution function is wanted to generate new data for any QC, the same study has been performed with nine more datasets from other QC and Vessels to try to obtain one that can be used for all.

4.4.5 Resume for all Datasets

Just to clarify, the same procedure as described previously has been used with all the other nine datasets coming from different QC and Vessels. See in *Table 4* below a resume of all the estimated parameters of the three probabilistic candidates:

Vessel	Quaycrane	Gamma		Weibull		Exponential
		Shape	Rate	Shape	Scale	Rate
09506AAT	QC47	1,2660	0,0140	1,1563	95,0720	0,0111
	QC48	1,3924	0,0152	1,2359	98,1781	0,0109
09097AAT	QC50	1,1876	0,0118	1,0977	104,4289	0,0099
	QC51	1,2365	0,0148	1,1248	87,2602	0,0120
09710AAT	QC49	1,6432	0,0179	1,2918	99,9330	0,0109
	QC50	1,2178	0,0123	1,1914	104,3789	0,0101
09010AAT	QC47	2,2791	0,0249	1,6462	102,2326	0,0109
	QC49	1,7651	0,0174	1,3480	111,2128	0,0098
09505AAT	QC48	1,5686	0,0178	1,3561	96,1396	0,0113
	QC49	1,2182	0,0136	1,1981	94,2813	0,0112

Table 4. Estimated parameters of the three probabilistic candidates

In order to complement this, in *Table 5* and *Table 6* below all the results of the validation tests for each candidate are shown. Note that the QQ Plot value is the Multiple R-Squared and for the K-S and the Chi-Squared Test the value corresponds to a P-value.

Vessel	Quaycrane	Gamma			Weibull		
		QQ	K-S	Chi-squared	QQ	K-S	Chi-Squared
09506AAT	QC47	0,9856	0,8113	0,3293	0,9835	0,9529	0,7425
	QC48	0,9899	0,4564	0,3864	0,9891	0,7426	0,4478
09097AAT	QC50	0,9079	0,7078	0,1933	0,9029	0,7890	0,3993
	QC51	0,9848	0,9467	0,8623	0,9815	0,8664	0,5013
09710AAT	QC49	0,8979	0,9568	0,7317	0,8853	0,8762	0,7438
	QC50	0,9772	0,0049*	0,0000*	0,9760	0,0199*	0,0001*
09010AAT	QC47	0,9853	0,0631	0,0032*	0,9779	0,2962	0,0023*
	QC49	0,9461	0,7951	0,2271	0,9313	0,3932	0,1215
09505AAT	QC48	0,9899	0,1178	0,0427*	0,9933	0,3997	0,0670
	QC49	0,9861	0,0039*	0,0001*	0,9897	0,0261*	0,0006*

Table 5. Validation tests results for Gamma and Weibull distributions

Vessel	Quaycrane	Exponential		
		QQ	K-S	Chi-Squared
09506AAT	QC47	0,9835	0,1261	0,1056
	QC48	0,9826	0,0440*	0,0301*
09097AAT	QC50	0,9123	0,5169	0,2085
	QC51	0,9869	0,2584	0,4381
09710AAT	QC49	0,9073	0,0063*	0,0025*
	QC50	0,9731	0,0003*	0,0000*
09010AAT	QC47	0,9616	0,0000*	0,0000*
	QC49	0,9572	0,0000*	0,0000*
09505AAT	QC48	0,9724	0,0000*	0,0000*
	QC49	0,9794	0,0000*	0,0001*

Table 6. Validation tests results for Exponential distribution

Note that all the values with an asterisk in the end mean that are below the **0,05** threshold to accept the null hypothesis. Consequently, the null hypothesis is rejected and the probabilistic distribution candidate does not fit the dataset.

As can be seen from the results, all three candidates have a Multiple R-squared near to one, so no conclusion can be taken from the QQ Plot validation Test. On the other hand, looking at the Goodness of fit Tests, the Exponential distribution candidate can be discarded since just three of the ten datasets passes the K-S and Chi-Squared tests.

Both, Gamma and Weibull probabilistic distributions can be used to generate new data for run future simulations since both of them pass the majority of the K-S and Chi-Squared test for all ten QC (6 out of 10). For this reason, the case studies that are going to be presented in the next chapter will be performed with data generated from both probabilistic distributions, Gamma and Weibull.

To have a common probabilistic distribution for each one, it has been decided to compute the mean of each parameter with the QC datasets that passes all the test, just to make sure that the parameters used really represents the real data. See below in *Table 7* the final probabilistic distributions that will be used to generate new data:

Gamma		Weibull	
Shape	Rate	Shape	Scale
1,4151	0,0152	1,2091	99,3475

Table 7. Parameters of the probabilistic distributions chosen

4.4.6 Generating new data

In this section, how new data is generated and how is merged with current data to perform new analysis will be explained.

4.4.6.1 *Generation*

R has been used to generate new data, see below the coded used:

```
#Gamma distribution parameters
gshape <- 1.4151
grate <- 0.0152

#Weibull distribution parameters
wshape <- 1.2091
wscale <- 99.3475

#Number of values that want to be generated
nvalues <- 146

gamma <- rgamma(nvalues, gshape, grate)
write.csv(gamma, file = "Gamma_09710AAT_QC50.csv")
weibull <- rweibull(nvalues, wshape, wscale)
write.csv(weibull, file = "Weibull_09710AAT_QC50.csv")
```

First of all, the parameters for the Gamma and Weibull probabilistic distribution are set. After that, one must indicate the number of values that want to be generated and then run the R-commands *rgamma* and *rweibull* with the distribution parameters set and the number of values that want to be generated. Once done that, random values with these probabilistic distributions will be generated. To have them available for merging, these values are saved in a csv file so that they can be imported into an excel file and merge them with the current data.

4.4.6.2 Merge

Once the new data is generated, the new *dispatch times* must be merged with Benchmark file data. This process is not straightforward, some steps have to be done to merge them correctly. Since the new times generated correspond to just one QC, some rules must be followed to be consistent with real processes. That means that QCs normally work in a way that our data must follow.

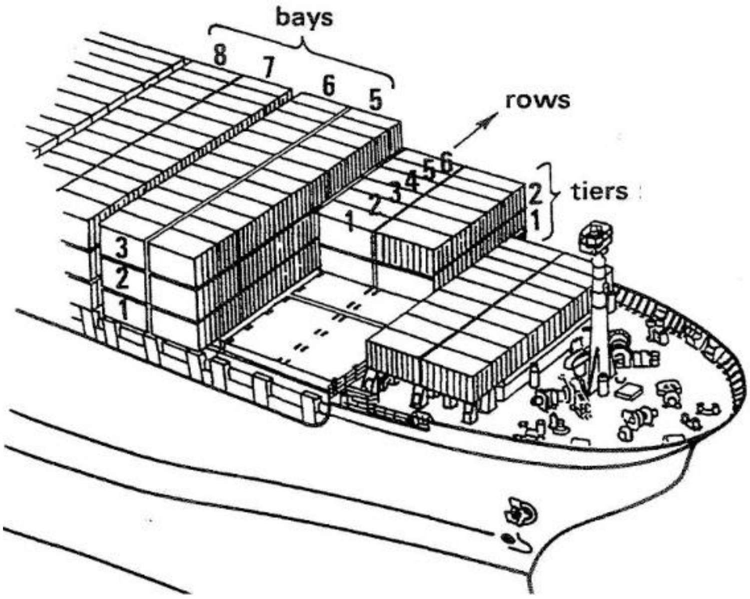


Figure 21. How a position is defined inside a vessel

Container vessels have their own way to position the containers inside of it, *Figure 21* shows how a container position is defined, they work in a similar way to yard blocks. Each position has three parameters: bay, row and tier and with these three parameters all the positions in a vessel are described.

One QC normally takes a certain number of bays and realize all the operations needed for them. For example, let's say that one QC works from bay 1 to bay 14, normally the QC will start performing all the movements from one extreme, then move the next one and so on finishing in the other extreme.

To follow this process, it has been decided that each QC will perform all its movements from the bay with largest number and finish with the least number; i.e. for the previous case from bay 14 to bay 1. To accomplish that, for each QC data, movements will be sorted by the largest bay to the least one and new times will be merged following that order.

4.5 Case analysis

In this chapter, four vessels will be studied and a sensitive analysis about the number of SC used for the unloading and loading operations will be performed. Three analyses will be performed for each vessel:

- One with the real data
- One with the data generated with the Gamma probabilistic distribution
- One with the data generated with the Weibull probabilistic distribution

Since the container terminal possess a resource pool of twenty-five SCs, exactly twenty-five simulations will be made for each analysis. Using from one up to twenty-five SCs. Basically, two things have been computed during the simulation:

- The resource utilization in each moment
- The total number of movements performed when the simulation has finished

Finding the maximum resource utilization from all the simulations and with the total number of movements performed a graph can be plotted and some conclusions can be reached.

4.5.1 Vessel 09710AAT

4.5.1.1 Vessel Information

This vessel berth took place on Tuesday 20th of August in 2013. The operations started at 07:00 in the morning and finished at 12:30 in the afternoon. During five hours and thirty minutes seven SCs have been working to realize 337 container movements.

Berth Date		20/08/2013
Number of SC used		7
Operations	Start Time	07:00:00
	Finish Time	12:30:00
	Hours	5,5
	Minutes	330
	Seconds	19800
Moves	Total	337
	After clean	316

Table 8. 09710AAT Vessel information

Note that after cleaning the data there is still a 93,8% of the movements which means that almost the totality of them will be simulated. Cleaning the data consists of eliminating container movements that their positions within the yard are not modelled and prepare it to run the simulation.

4.5.1.2 Real Data analysis

The following graph *(Figure 22)* is plotted after running all the twenty-five simulations with real data:

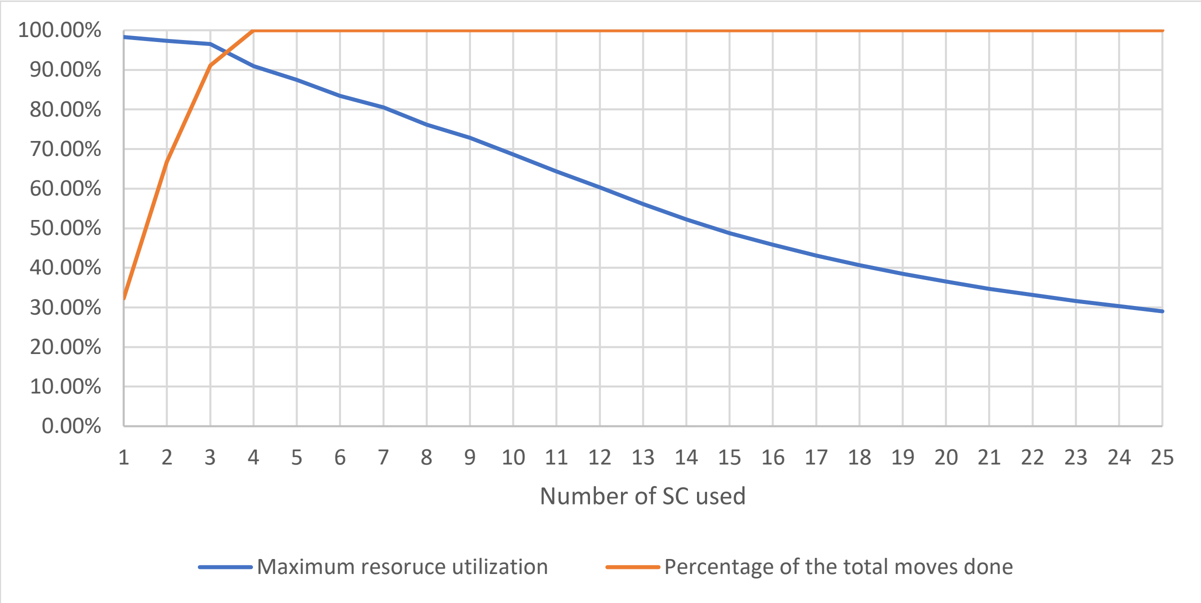


Figure 22. Real Data analysis for 09710AAT vessel

Basically, this graph gives the information of the maximum resource utilization and the percentage of the total movements done during the whole simulation. At first sight, the left part of the graph until four SCs used is not interesting because not all the moves needed for the vessel have been performed. Then, the interesting part starts from four SC.

In this particular vessel, seven SCs were used and reading the graph means that an 80,50% of the time were in use. And as could be expected, the use of more SCs decreases the maximum resource utilization. That happens because for the same amount of movements more SCs are used, therefore, SCs will be idle more time.

One more thing that can be observed is that the decrease slope for maximum resource utilization is higher when the number of SCs used is low and it tends to decrease and converge with higher values of SCs used. In this particular case, at low numbers of SCs used the decreasing is around 3-4% and for high numbers of SCs tends to decrease until 1,2%.

4.5.1.3 Generated Data analysis (Gamma)

The following graph *(Figure 23)* is plotted after generating the data, merging it with the Benchmark file and running all the twenty-five simulations:

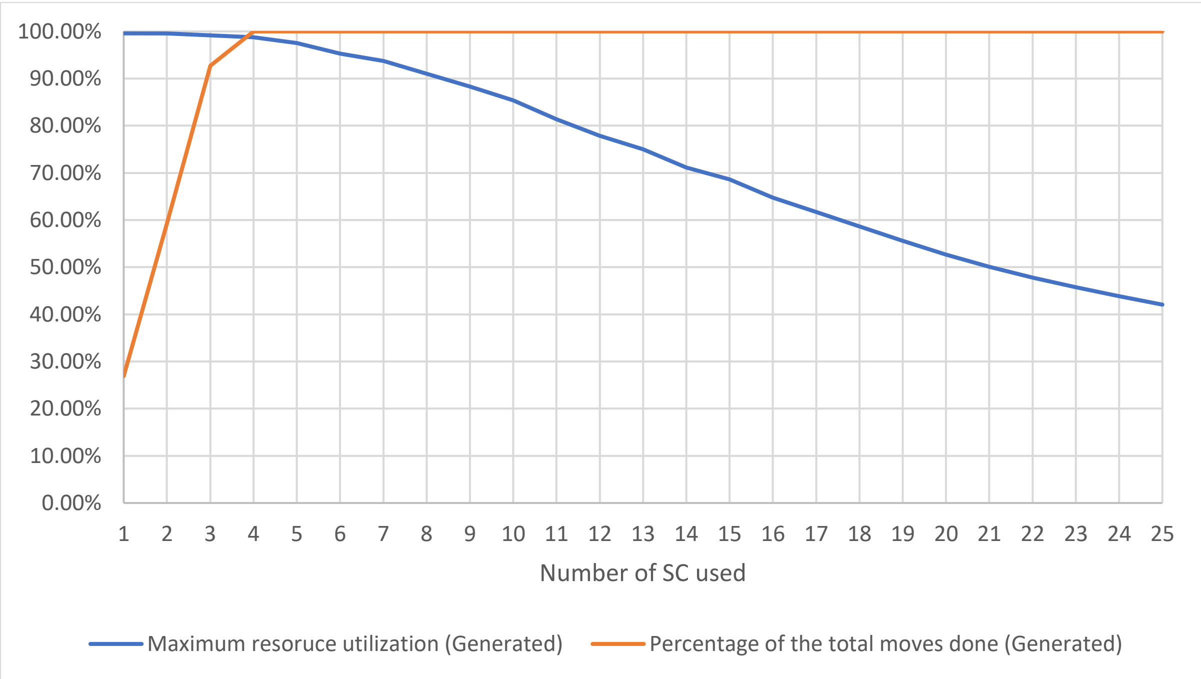


Figure 23. Generated Data analysis (Gamma) for 09710AAT vessel

At first sight looking at *Figure 23,* one can think that the graph is the same as the one with real data. The *percentage of the total moves done* line increases and reaches 100% using four SCs and the *maximum resource utilization* starts from almost 100% and decreases when more SCs are used.

Nevertheless, if one look directly at the numbers a slight difference is appreciated, the *maximum resource utilization* values are higher. To see better this difference, the graph with real data and the one with generated data have been merged into one so a comparison can be made and some conclusions can be reached.

Below in *Figure 24*, both previous graphs are merged into one:

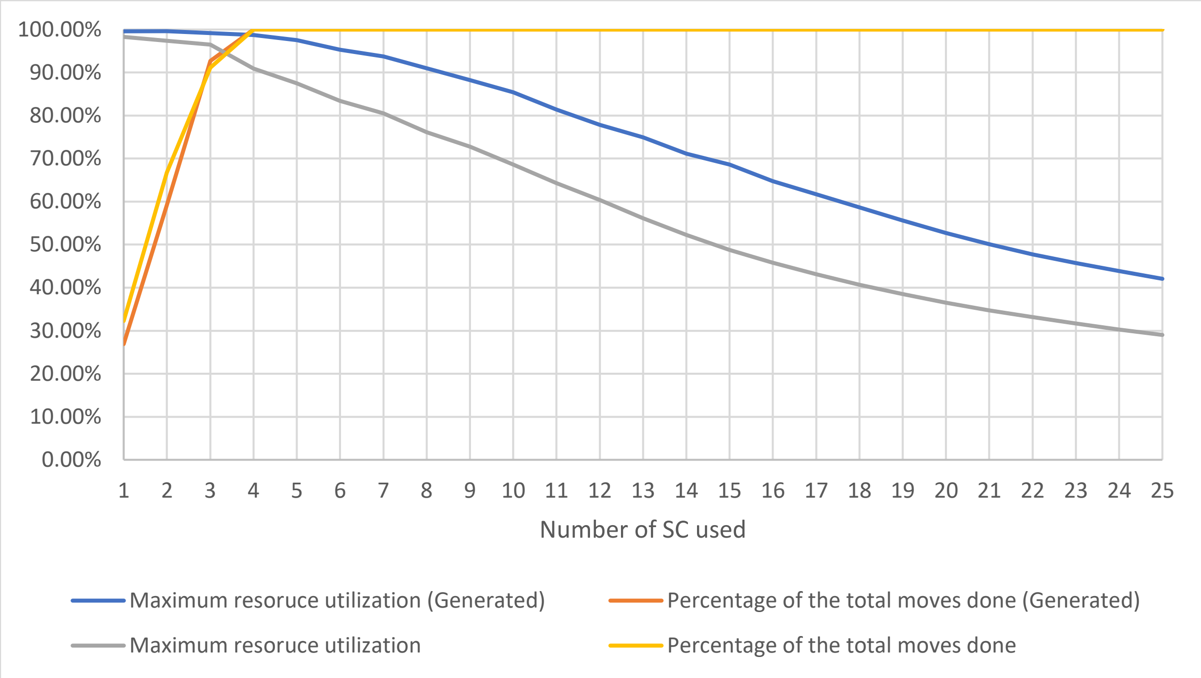


Figure 24. Real and Generated Data (Gamma) analyses for 09710AAT vessel

In *Figure 24,* the difference between both analyses can be seen. First of all, looking at the *percentage of the total moves done* it can be stated that both are practically the same but there are some minor differences. This difference can be explained because the order of the movements may be different between real data and the one with times generated. That happens because, as explained in *section 4.4.6.2,* a QC normally perform all the operations from one bay before moving to the one next to it, but it can be seen in the real data that this is not always happening. Because of this, when new times are generated and merged with the Benchmark file, the order of the movements may differ and consequently some movements can take more time to perform or less time because of its container position within the yard.

Secondly, looking at the *maximum resource utilization* it can be stated that values in the generated one are higher than the real ones with an average of 15,53% of difference (without counting less than four SC used because all the movements are not completed). This difference can be understood because of the following reason: possible breaks that SC and QC drivers may do or stops because some problems happen are not contemplated while fitting the data to generate the new ones. That is because those times are normally large values and while trying to validate the dataset they are considered as outliers and many of them have been removed from the dataset before starting all the fitting process. Consequently, no breaks or stops are considered and all the movements are done without any stop saturating the SC resource pool more than with the real data.

4.5.1.4 Generated Data analysis (Weibull)

The following graph *(Figure 25)* is plotted after generating the data, merging it with the Benchmark file and running all the twenty-five simulations:

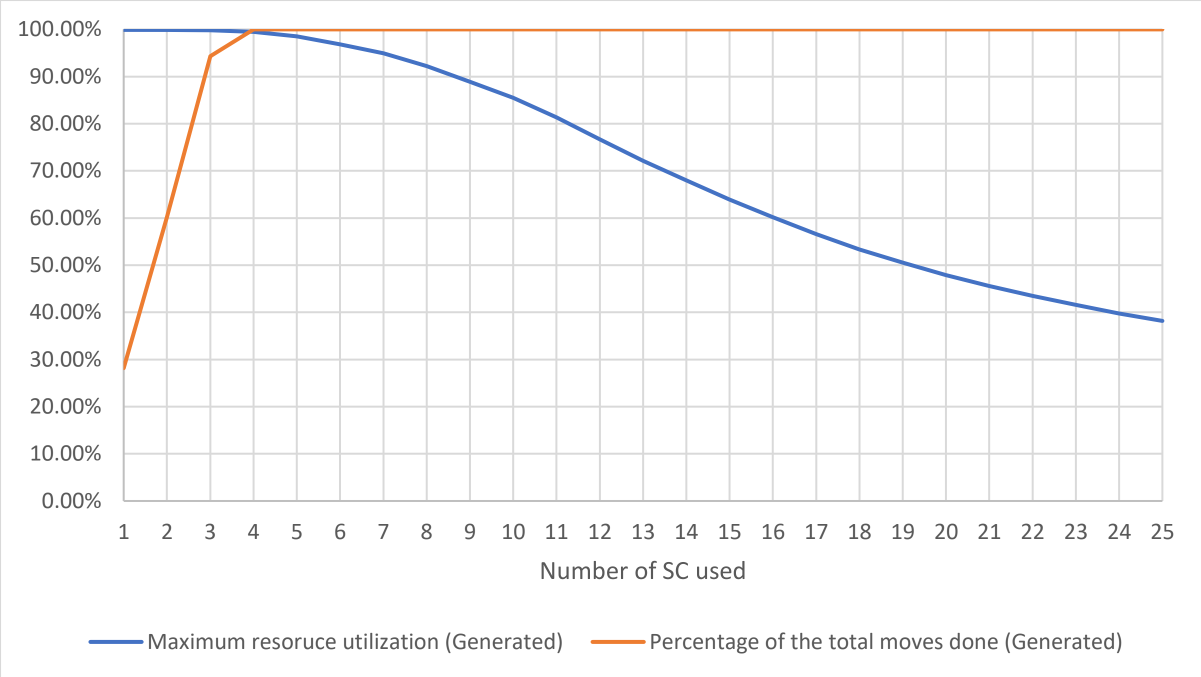


Figure 25. Generated Data analysis (Weibull) for 09710AAT vessel

At first sight looking at *Figure 25*, it happens the same as with the Gamma one, one can think that the graph is equal to the one with real data. The *percentage of the total moves done* line increases and reaches 100% using four SC and the *maximum resource utilization* starts from almost 100% and decreases when more SC are used.

Nevertheless, if one look directly at the numbers a slight difference is as well appreciated, the *maximum resource utilization values* are higher. To better see this difference, the graph with real data and the one with generated data have been merged into one so a comparison can be taken and some conclusions can be determined.

Below in *Figure 26*, previous graph and the real data one are merged into one:

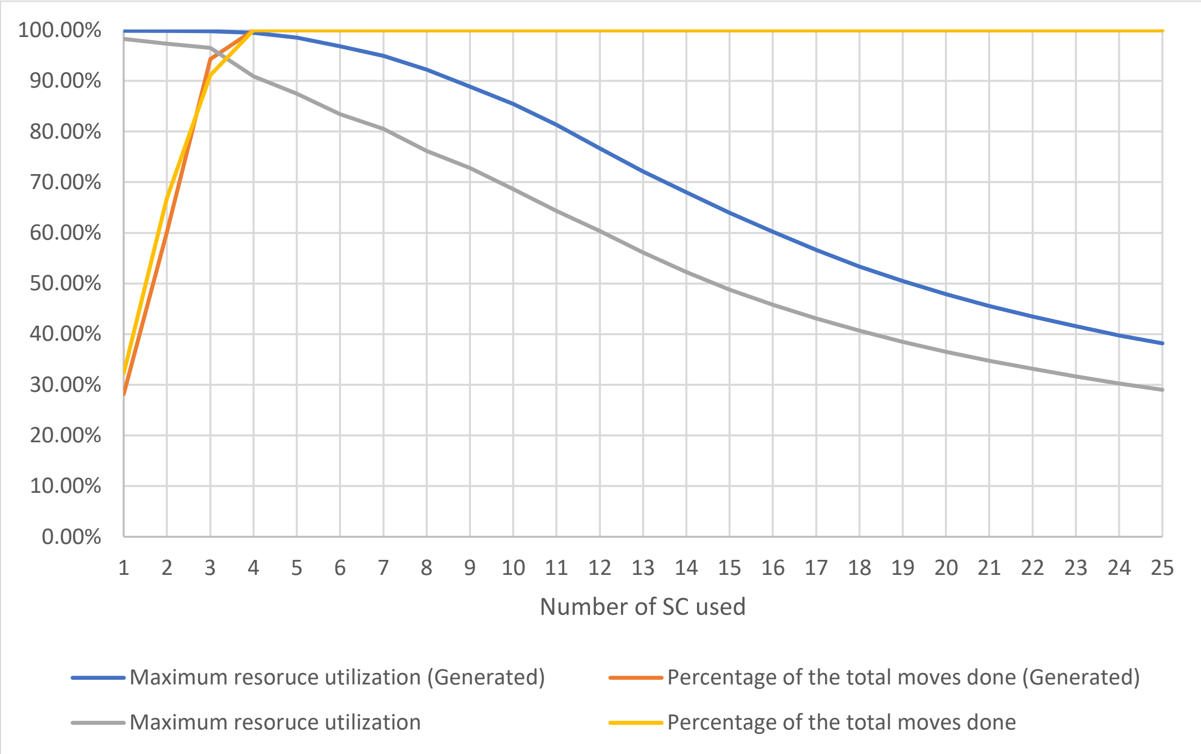


Figure 26. Real and Generated Data (Weibull) analyses for 09710AAT vessel

In *Figure 26* it can be seen the difference between both analyses. First of all, looking at the *percentage of the total moves done* it can be stated that both are practically the same but there are some minor differences. The reason why this is happening is explained in the previous *section 4.5.1.3.*

Secondly, looking at the *Maximum resource utilization* it can be stated that values in the generated one are higher than the real ones with an average of 13,19% of difference (without counting less than four SC used because all the movements are not completed). The reason of this difference is also explained in the previous *section 4.5.1.3.*

Finally, having seen both analyses with generated data and their comparison with the one with real data, it can be stated that at least for this vessel the Weibull distribution reproduces the reality better than the Gamma distribution but not completely.

4.5.2 Vessel 09097AAT

4.5.2.1 Vessel Information

This vessel berth took place on Tuesday 23th of April in 2013. The operations started at 07:00 in the morning and finished at 12:30 in the afternoon. During five hours and thirty minutes seven SC have been working to realize 355 container movements.

Berth Date		23/04/2013
Number SC used		7
Operations	Start Time	07:00:00
	Finish Time	12:30:00
	Hours	5,5
	Minutes	330
	Seconds	19800
Moves	Total	355
	After clean	351

Table 9. 09097AAT Vessel information

Note that after cleaning the data there is still a 98,9% of the movements which means that almost the totality of them will be simulated. Cleaning the data consists of eliminating container movements that their positions within the yard are not modelled and prepare it to run the simulation.

4.5.2.2 Real Data analysis

The following graph *(Figure 27)* is plotted with the maximum resource utilization and the percentage of the total moves done from each twenty-five simulations using real data.

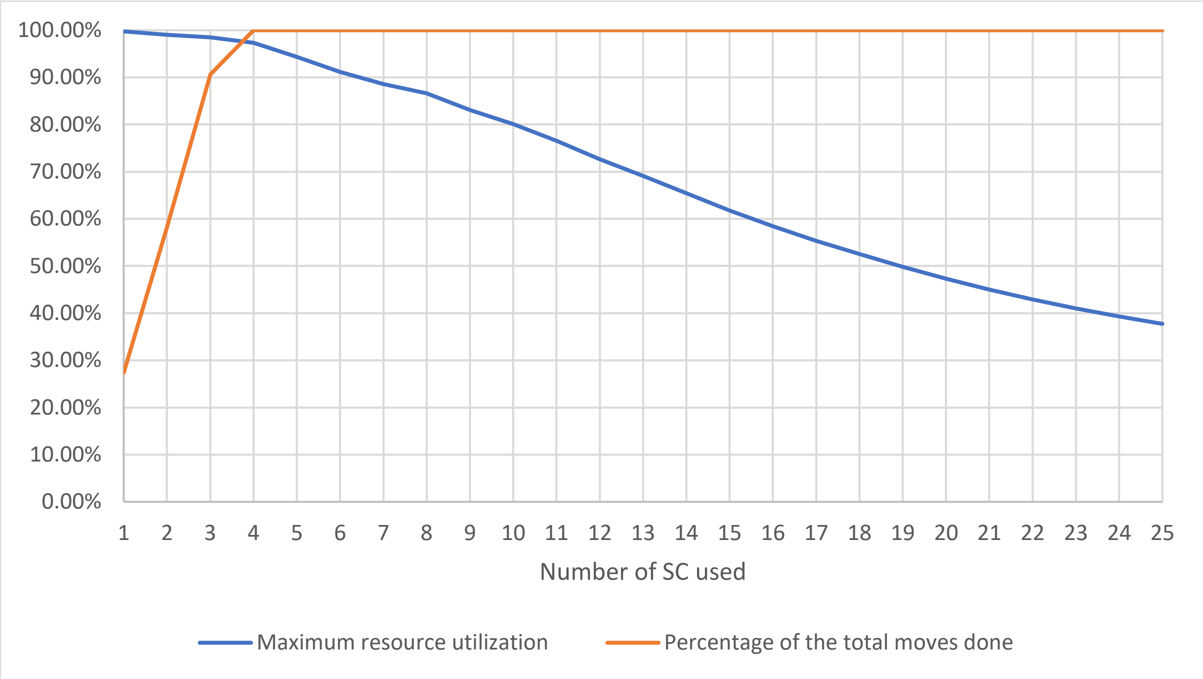


Figure 27. Real Data analysis for 09097AAT vessel

In this particular vessel, seven SCs were used and reading the graph means that an 88,58% of the time were in use. And as could be expected, the use of more SCs decreases the maximum resource utilization. That happens because for the same amount of movements more SCs are used, therefore, SCs will be idle more time.

One more thing that can be observed is that the decrease slope for maximum resource utilization is higher when the number of SCs used is low and it tends to decrease and converge with higher values of SCs used. In this particular case, at low numbers of SCs used the decreasing is around 3-4% and for high numbers of SCs tends to decrease until 1,5%.

4.5.2.3 Generated Data analysis (Gamma)

The following graph *(Figure 28)* is plotted after generating the data, merging it with the Benchmark file and running all the twenty-five simulations:

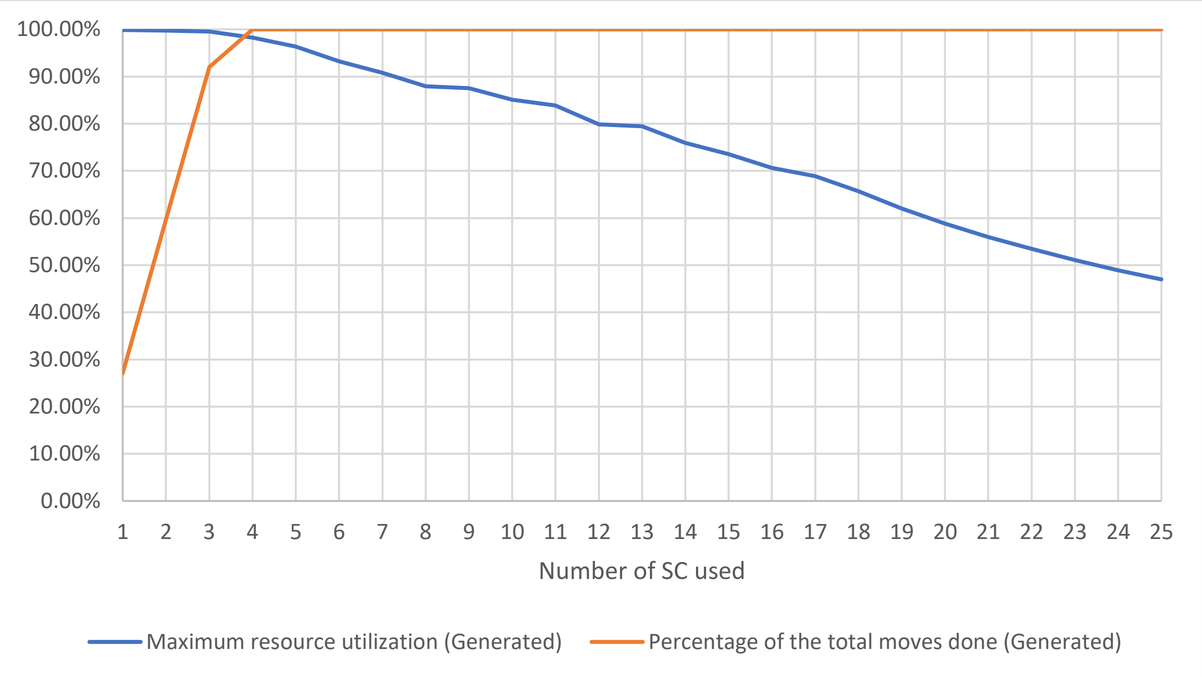


Figure 28. Real and Generated Data (Gamma) analyses for 09097AAT vessel

At first sight looking at *Figure 28,* one can think that the graph is the same as the one with real data. The *percentage of the total moves done* line increases and reaches 100% using four SCs and the *maximum resource utilization* starts from almost 100% and decreases when more SCs are used.

Nevertheless, if one look directly at the numbers a slight difference is appreciated, the *maximum resource utilization* values are higher. To see better this difference, the graph with real data and the one with generated data have been merged into one so a comparison can be made and some conclusions can be reached.

Below in *Figure 29*, both previous graphs are merged into one:

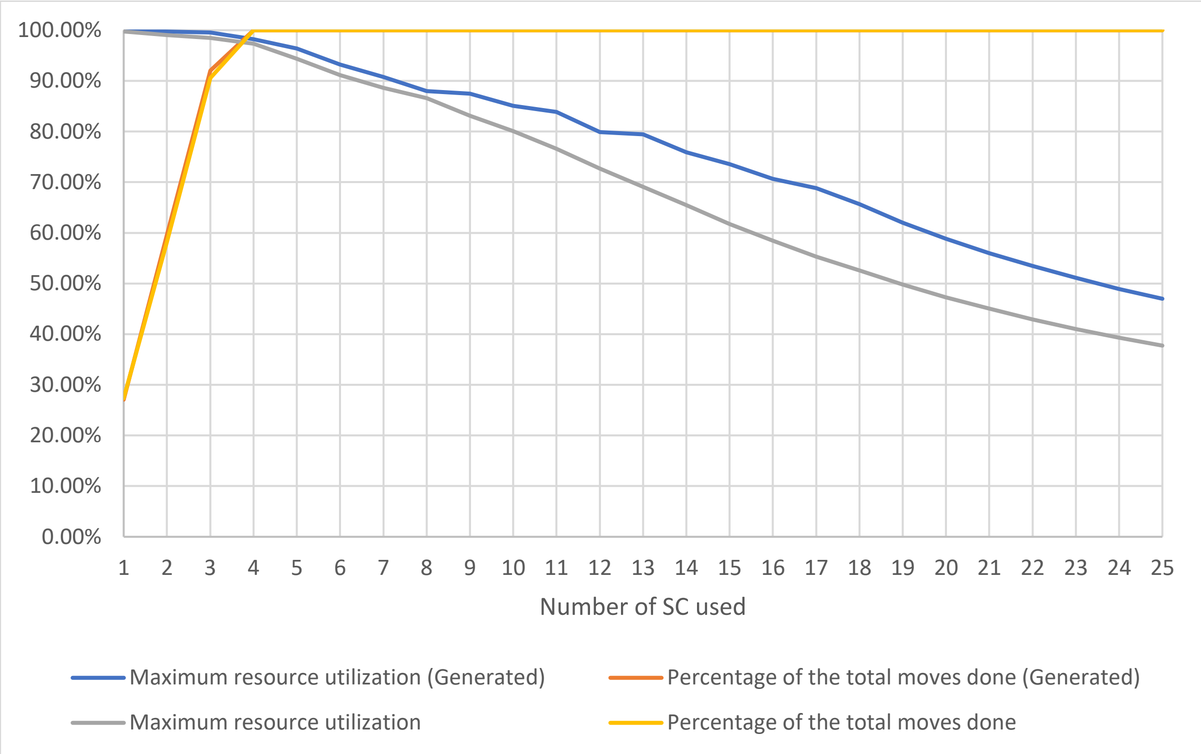


Figure 29. Real and Generated Data (Gamma) analyses for 09097AAT vessel

In *Figure 29* it can be seen the difference between both analyses. First of all, looking at the *percentage of the total moves done* it can be stated that both are practically the same but there are some minor differences. The reason why this is happening is explained in the previous *section 4.5.1.3*.

Secondly, looking at the *Maximum resource utilization* it can be stated that values in the generated one are higher than the real ones with an average of 8,09% of difference (without counting less than four SC used because all the movements are not completed). The reason of this difference is also explained in the previous *section 4.5.1.3*.

4.5.2.4 Generated Data analysis (Weibull)

The following graph *(Figure 30)* is plotted after generating the data, merging it with the Benchmark file and running all the twenty-five simulations:

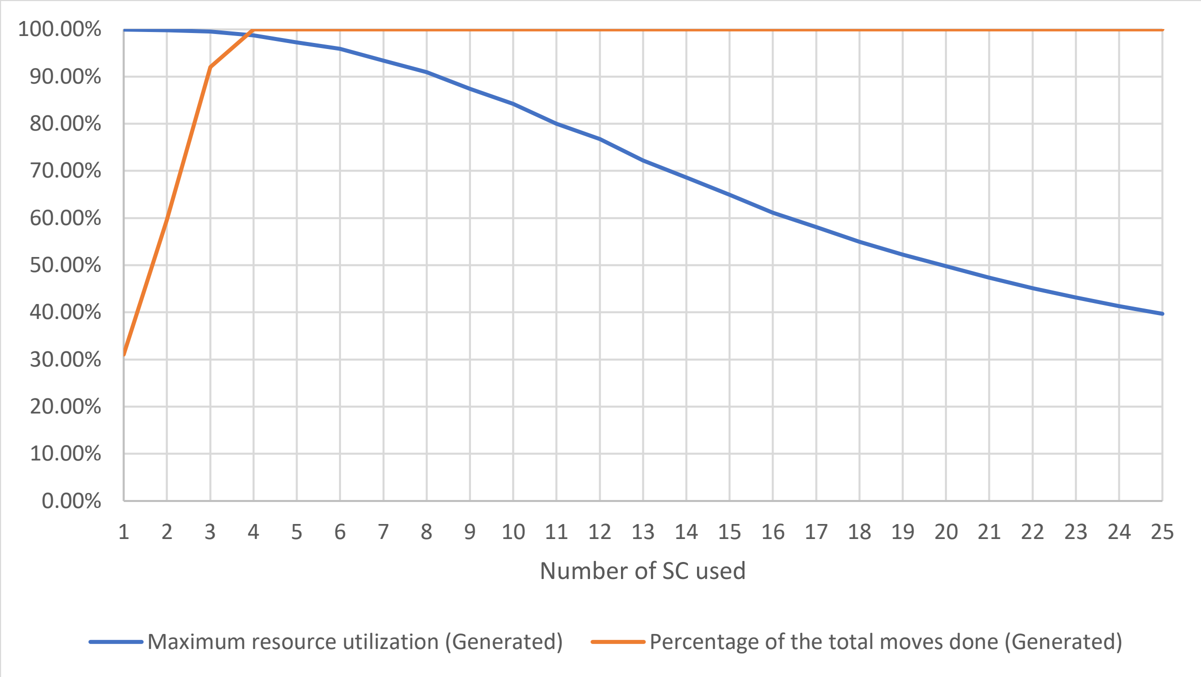


Figure 30. Generated Data (Weibull) analyses for 09097AAT vessel

At first sight looking at *Figure 30,* it happens the same as with the Gamma one, one can think that the graph is equal to the one with real data. The *percentage of the total moves done* line increases and reaches 100% using four SC and the *maximum resource utilization* starts from almost 100% and decreases when more SC are used.

Nevertheless, if one look directly at the numbers a slight difference is as well appreciated, the *maximum resource utilization values* are slightly higher. To better see this difference, the graph with real data and the one with generated data have been merged into one so a comparison can be taken and some conclusions can be determined.

Below in *Figure 31*, previous graph and the real data one are merged into one:

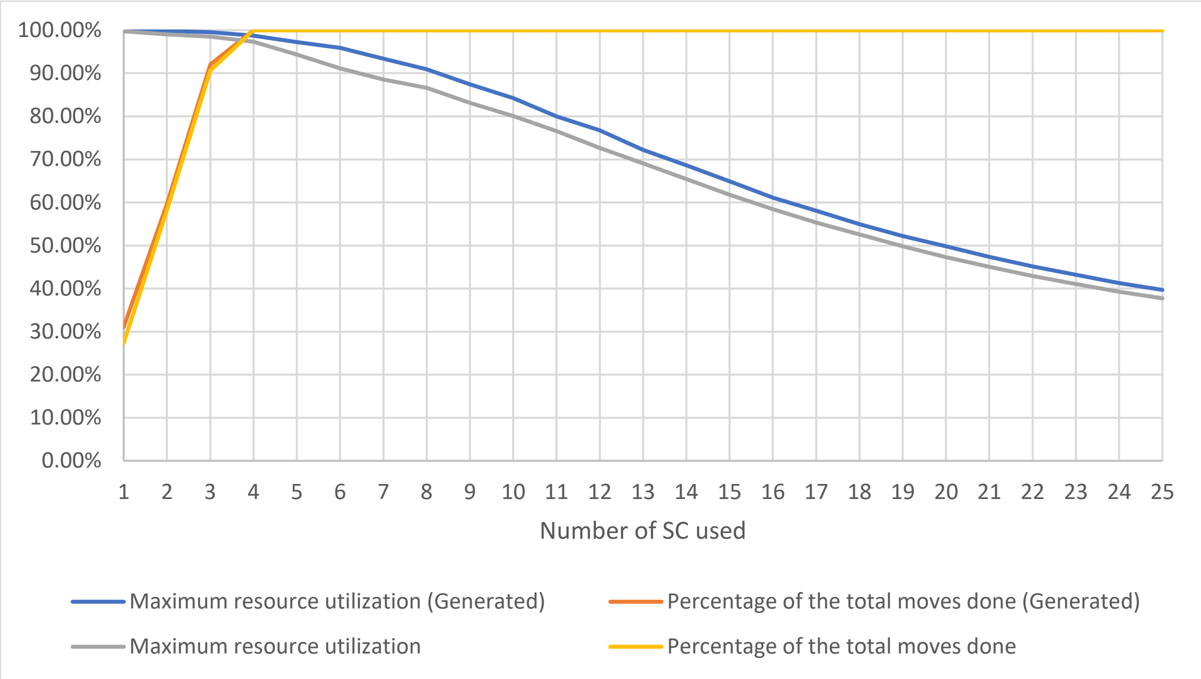


Figure 31. Real and Generated Data (Weibull) analyses for 09097AAT vessel

In *Figure 31* it can be seen the difference between both analyses. First of all, looking at the *percentage of the total moves done* it can be stated that both are practically the same but there are some minor differences. The reason why this is happening is explained in the previous section 5.5.1.3.

Secondly, looking at the *Maximum resource utilization* it can be stated that values in the generated one are higher than the real ones with an average of 3,04% of difference (without counting less than four SC used because all the movements are not completed). The reason of this difference is also explained in the previous section 5.5.1.3.

Finally, having seen both analyses with generated data and their comparison with the one with real data, it can be stated that at least for this vessel the Weibull distribution reproduces the reality better than the Gamma distribution not completely but with a fairly difference.

4.5.3 Vessel 09506AAT

4.5.3.1 Vessel Information

This vessel berth took place on Tuesday 29th of August in 2013. The operations started at 13:20 in the afternoon and finished at 20:50 in the evening. During seven hours and thirty minutes seven SC have been working to realize 455 container movements.

Berth Date		29/08/2013
Number SC used		7
Operations	Start Time	13:20:00
	Finish Time	20:50:00
	Hours	7,5
	Minutes	450
	Seconds	27000
Moves	Total	455
	After clean	455

Table 10. 09506AAT Vessel information

Note that after cleaning the data the totality of the movements remains which means that every movement will be simulated and therefore a good analysis is expected from it. Cleaning the data consists of eliminating container movements that their positions within the yard are not modelled and prepare it to run the simulation.

4.5.3.2 Real Data analysis

The following graph *(Figure 32)* is plotted with the maximum resource utilization and the percentage of the total moves done from each twenty-five simulations using real data.

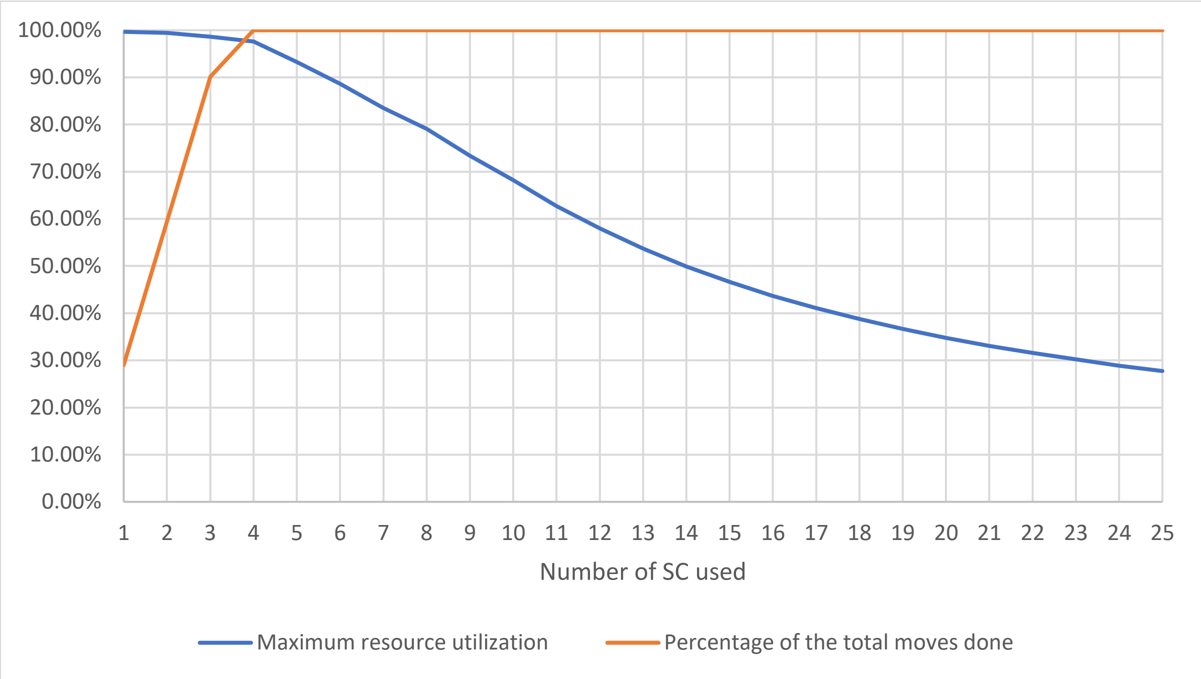


Figure 32. Real Data analysis for 09506AAT vessel

In this particular vessel, seven SCs were used and reading the graph means that an 83,52% of the time were in use. And as could be expected, the use of more SCs decreases the maximum resource utilization. That happens because for the same amount of movements more SCs are used, therefore, SCs will be idle more time.

One more thing that can be observed is that the decrease slope for maximum resource utilization is higher when the number of SCs used is low and it tends to decrease and converge with higher values of SCs used. In this particular case, at low numbers of SCs used the decreasing is around 4-6% and for high numbers of SCs tends to decrease until 1,13%.

4.5.3.3 Generated Data analysis (Gamma)

The following graph *(Figure 33)* is plotted after generating the data, merging it with the Benchmark file and running all the twenty-five simulations:

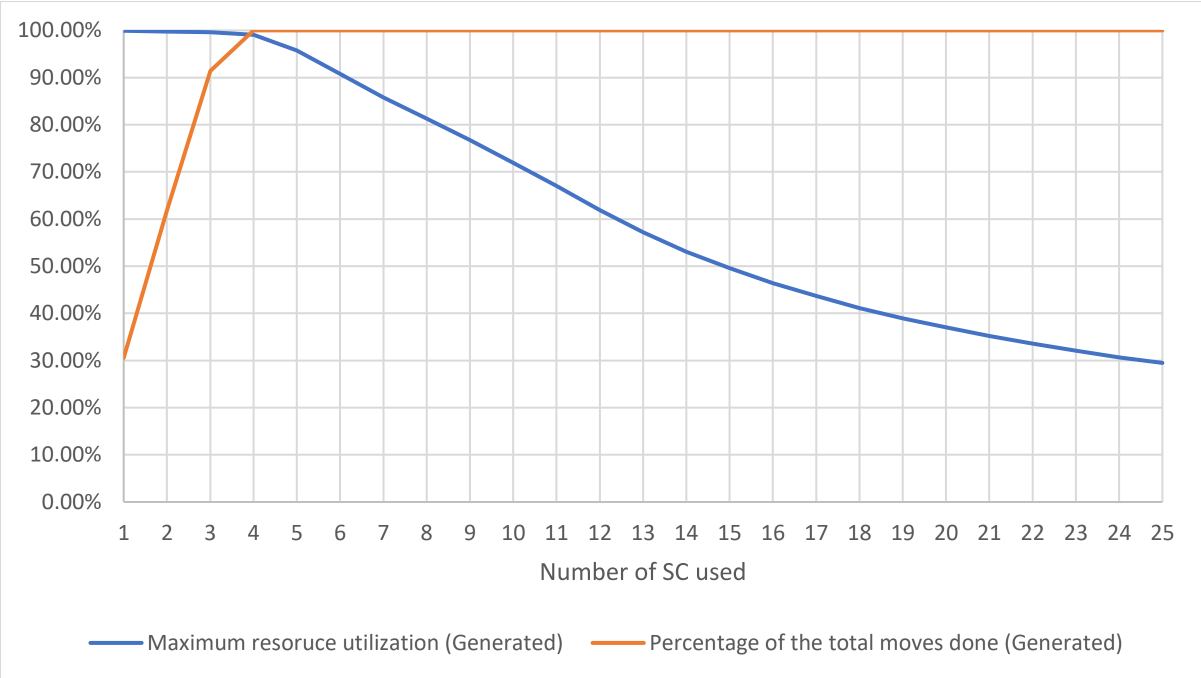


Figure 33. Generated Data (Gamma) analysis for 09506AAT vessel

At first sight looking at *Figure 33,* one can think that the graph is the same as the one with real data. The *percentage of the total moves done* line increases and reaches 100% using four SCs and the *maximum resource utilization* starts from almost 100% and decreases when more SCs are used.

Nevertheless, if one look directly at the numbers a slight difference is appreciated, the *maximum resource utilization* values are higher. To see better this difference, the graph with real data and the one with generated data have been merged into one so a comparison can be made and some conclusions can be reached.

Below in *Figure 34*, both previous graphs are merged into one:

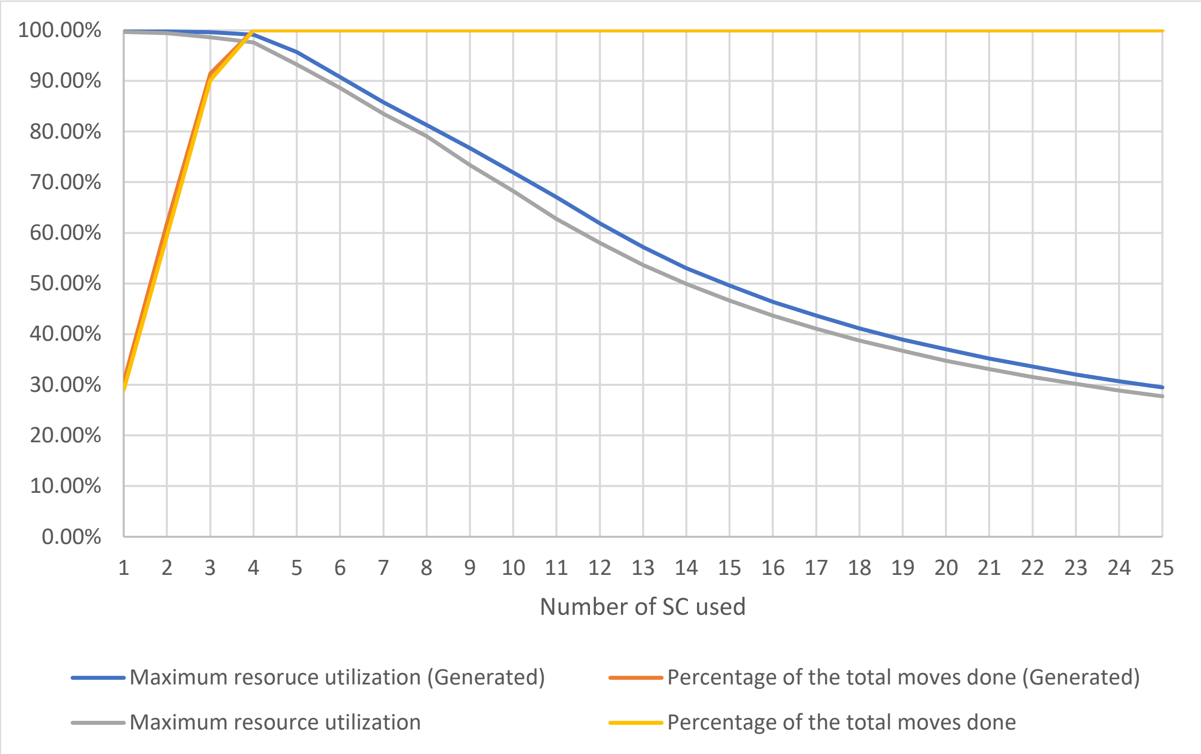


Figure 34. Real and Generated Data (Gamma) analyses for 09506AAT vessel

In *Figure 34* it can be seen the difference between both analyses. First of all, looking at the *percentage of the total moves done* it can be stated that both are practically the same but there are some minor differences. The reason why this is happening is explained in the previous *section 4.5.1.3*.

Secondly, looking at the *Maximum resource utilization* it can be stated that values in the generated one are higher than the real ones with an average of 2,60% of difference (without counting less than four SC used because all the movements are not completed). The reason of this difference is also explained in the previous *section 4.5.1.3*.

4.5.3.4 Generated Data analysis (Weibull)

The following graph *(Figure 35)* is plotted after generating the data, merging it with the Benchmark file and running all the twenty-five simulations:

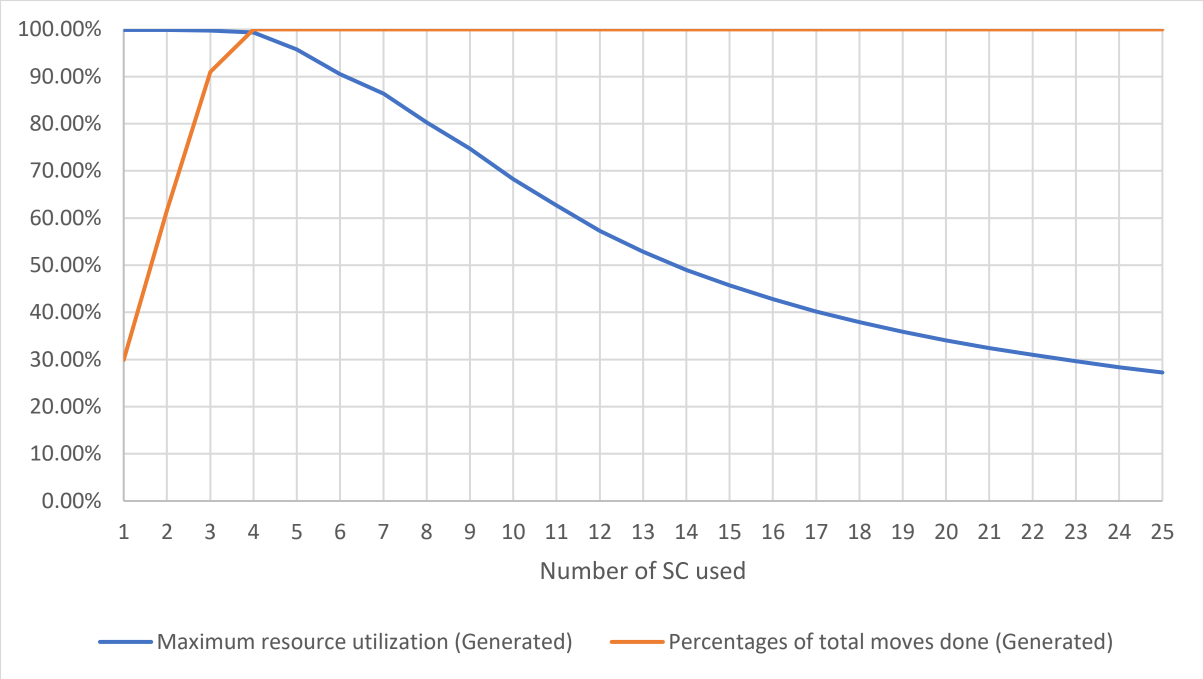


Figure 35. Generated Data (Weibull) analysis for 09506AAT vessel

At first sight looking at *Figure 35,* it happens the same as with the Gamma one, one can think that the graph is equal to the one with real data. The *percentage of the total moves done* line increases and reaches 100% using four SC and the *maximum resource utilization* starts from almost 100% and decreases when more SC are used.

Nevertheless, if one look directly at the numbers a slight difference is as well appreciated, the *maximum resource utilization values* are slightly higher. To better see this difference, the graph with real data and the one with generated data have been merged into one so a comparison can be taken and some conclusions can be determined.

Below in *Figure 36*, previous graph and the real data one are merged into one:

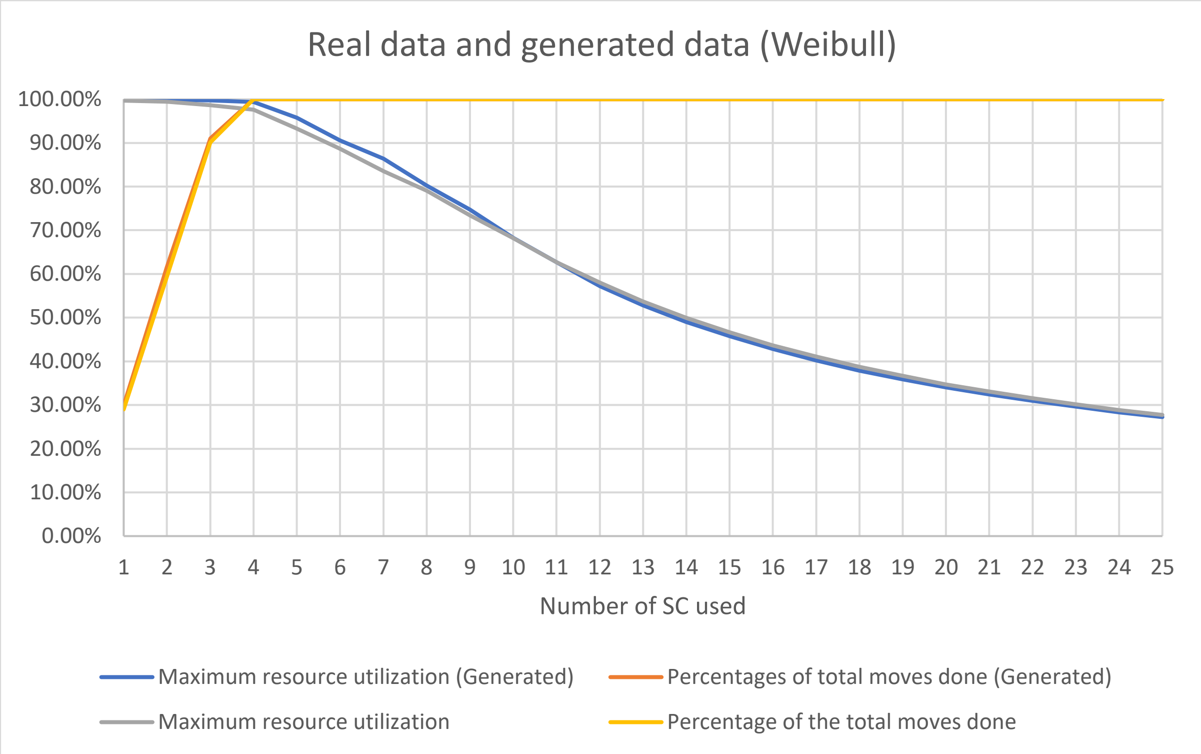


Figure 36. Real and Generated Data (Weibull) analyses for 09506AAT vessel

In *Figure 36* it can be seen the difference between both analyses. First of all, looking at the *percentage of the total moves done* it can be stated that both are practically the same but there are some minor differences. The reason why this is happening is explained in the previous section 5.5.1.3.

Secondly, looking at the m*aximum resource utilization* it can be stated that values in the generated one are slightly higher than the real ones between four and nine SCs and then both lines merge and are practically the same. However, computing the average difference between those values gives a 1,00% of average difference (without counting less than four SC used because all the movements are not completed).

Finally, having seen both analyses with generated data and their comparison with the one with real data, it can be stated that at least for this vessel the Weibull distribution reproduces the reality better than the Gamma distribution not completely but with a fairly difference. As said in the beginning of this vessel analysis, a good analysis was expected because the totality of the movements was simulated. The generated data have given very good results almost reproducing the exact reality, however the data have been generated randomly so maybe if another dataset is generated worse results could have been appeared.

4.5.4 Vessel 09505AAT

4.5.4.1 Vessel Information

This vessel berth took place on Thursday 22th of August in 2013. The operations started at 13:00 in the afternoon and finished at 21:30 in the evening. During eight hours and thirty minutes eight SC have been working to realize 520 container movements.

Berth Date		22/08/2013
Number SC used		8
Operations	Start Time	13:00:00
	Finish Time	21:30:00
	Hours	8,5
	Minutes	510
	Seconds	30600
Moves	Total	520
	After clean	493

Table 11. 09505AAT Vessel information

Note that after cleaning the data there is still a 94,8% of the movements which means that almost the totality of them will be simulated. Cleaning the data consists of eliminating container movements that their positions within the yard are not modelled and prepare it to run the simulation.

4.5.4.2 Real Data analysis

The following graph *(Figure 37)* is plotted with the maximum resource utilization and the percentage of the total moves done from each twenty-five simulations using real data.

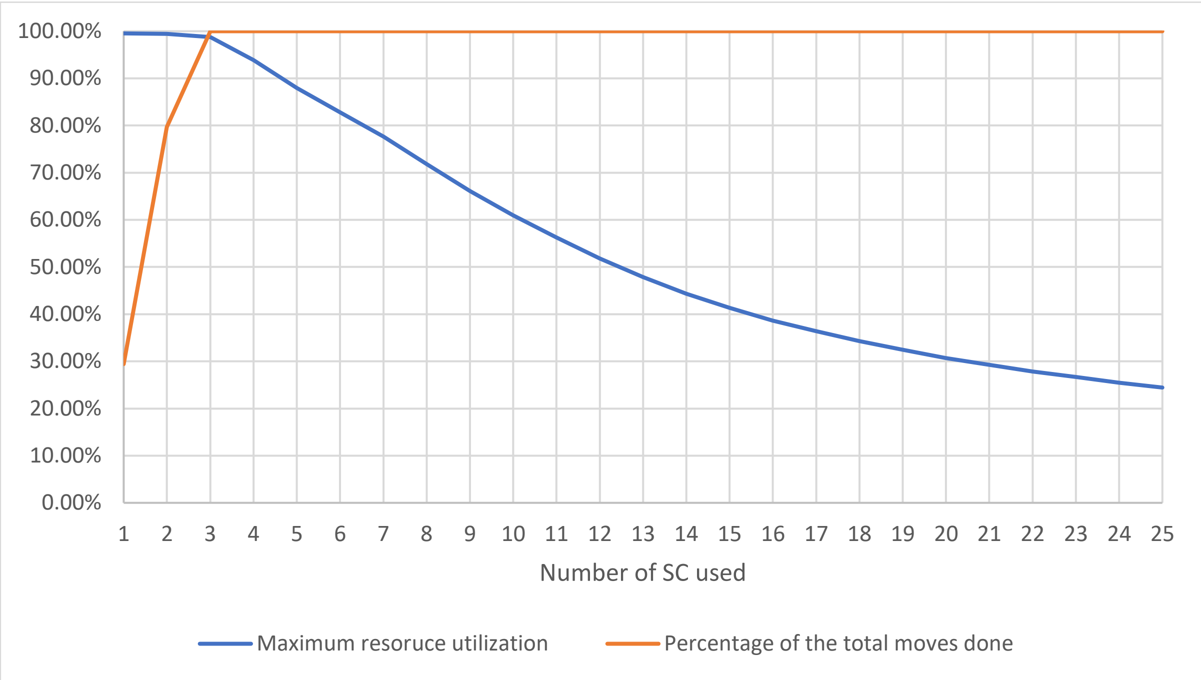


Figure 37. Real Data analysis for 09505AAT vessel

In this particular vessel, eight SCs were used and reading the graph means that a 71,84% of the time were in use. And as could be expected, the use of more SCs decreases the maximum resource utilization. That happens because for the same amount of movements more SCs are used, therefore, SCs will be idle more time.

One more thing that can be observed is that the decrease slope for maximum resource utilization is higher when the number of SCs used is low and it tends to decrease and converge with higher values of SCs used. In this particular case, at low numbers of SCs used the decreasing is around 4-6% and for high numbers of SCs tends to decrease until 1,03%.

4.5.4.3 Generated Data analysis (Gamma)

The following graph *(Figure 38)* is plotted after generating the data, merging it with the Benchmark file and running all the twenty-five simulations:

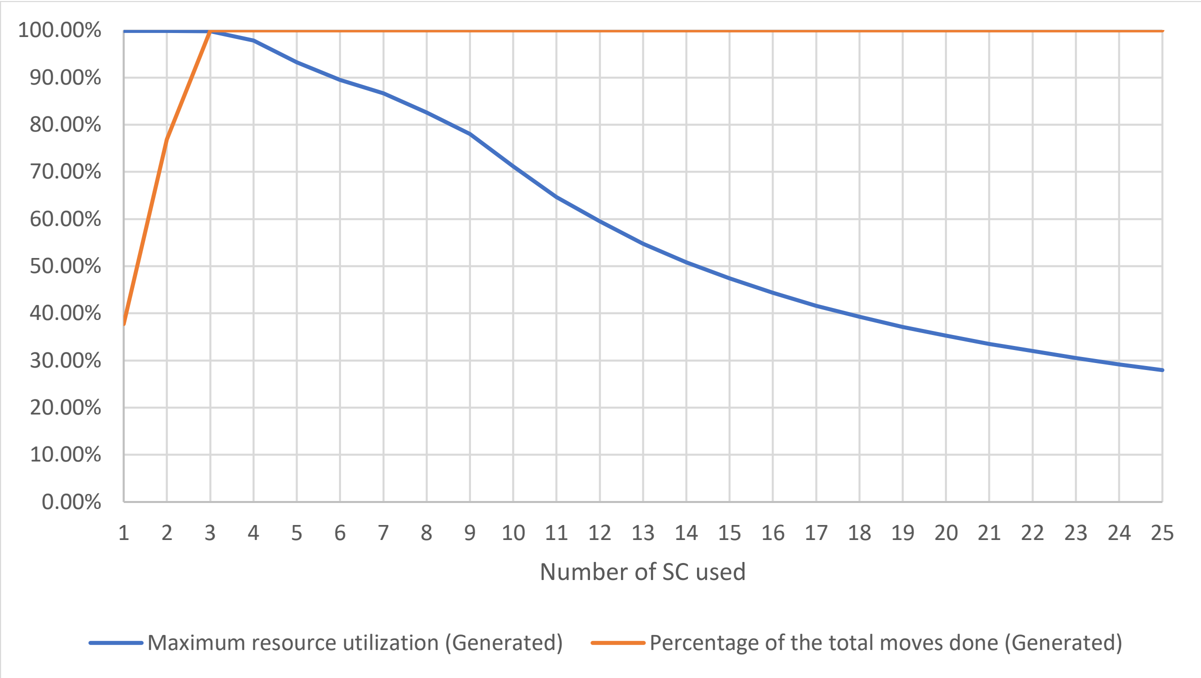


Figure 38. Generated Data (Gamma) analysis for 09505AAT Vessel

At first sight looking at *Figure 38,* one can think that the graph is the same as the one with real data. The *percentage of the total moves done* line increases and reaches 100% using three SCs and the *maximum resource utilization* starts from almost 100% and decreases when more SCs are used.

Nevertheless, if one look directly at the numbers a slight difference is appreciated, the *maximum resource utilization* values are higher. To see better this difference, the graph with real data and the one with generated data have been merged into one so a comparison can be made and some conclusions can be reached

Below in *Figure 39*, both previous graphs are merged into one:

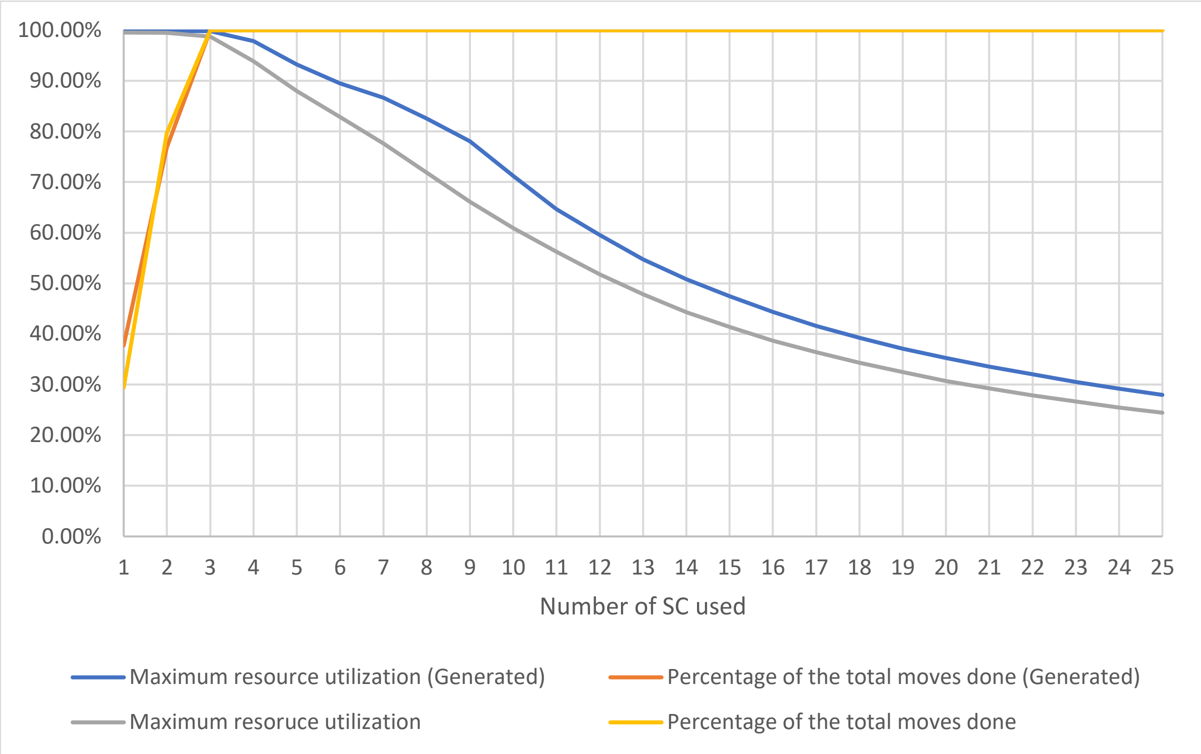


Figure 39. Real and Generated Data (Gamma) analyses for 09505AAT Vessel

In *Figure 39* it can be seen the difference between both analyses. First of all, looking at the *percentage of the total moves done* it can be stated that both are practically the same but there are some minor differences. The reason why this is happening is explained in the previous *section 4.5.1.3*.

Secondly, looking at the *Maximum resource utilization* it can be stated that values in the generated one are higher than the real ones with an average of 6,04% of difference (without counting less than four SC used because all the movements are not completed). The reason of this difference is also explained in the previous *section 4.5.1.3*.

4.5.4.4 Generated Data analysis (Weibull)

The following graph *(Figure 40)* is plotted after generating the data, merging it with the Benchmark file and running all the twenty-five simulations:

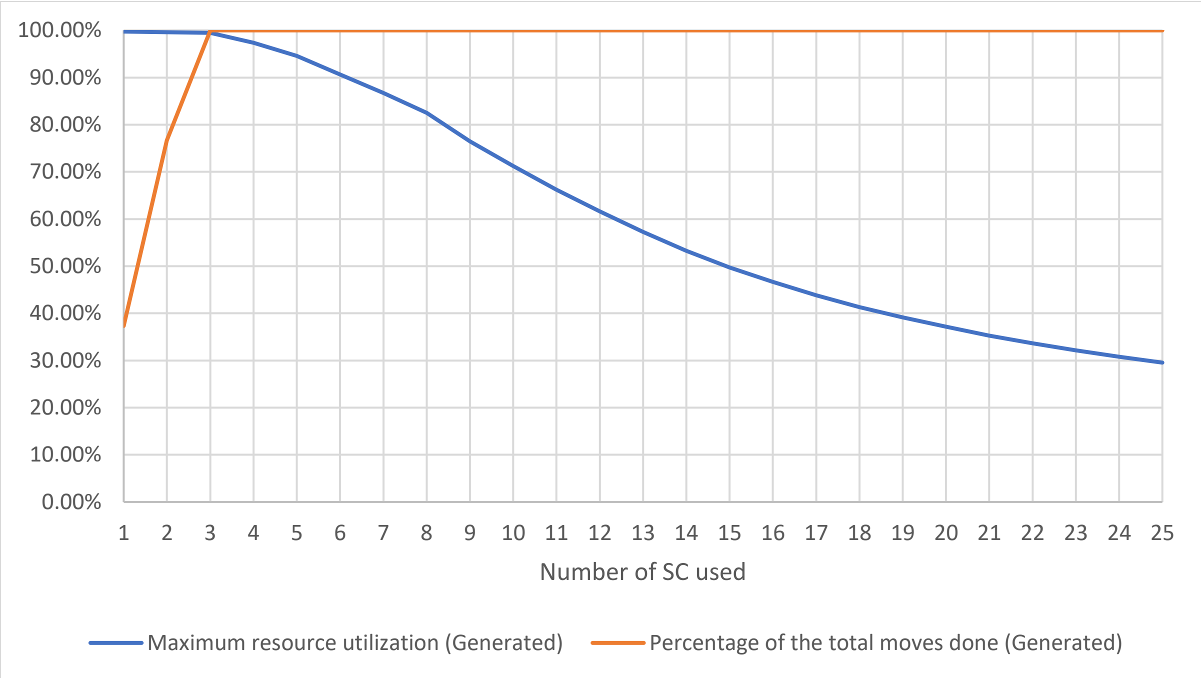


Figure 40. Generated Data (Weibull) analysis for 09505AAT Vessel

At first sight looking at *Figure 40*, it happens the same as with the Gamma one, one can think that the graph is equal to the one with real data. The *percentage of the total moves done* line increases and reaches 100% using three SC and the *maximum resource utilization* starts from almost 100% and decreases when more SC are used.

In this case, if one look directly at the numbers they are basically the same. To better see if they are pretty much the same, the graph with real data and the one with generated data have been merged into one so a comparison can be taken and some conclusions can be determined.

Below in *Figure 41*, previous graph and the real data one are merged into one:

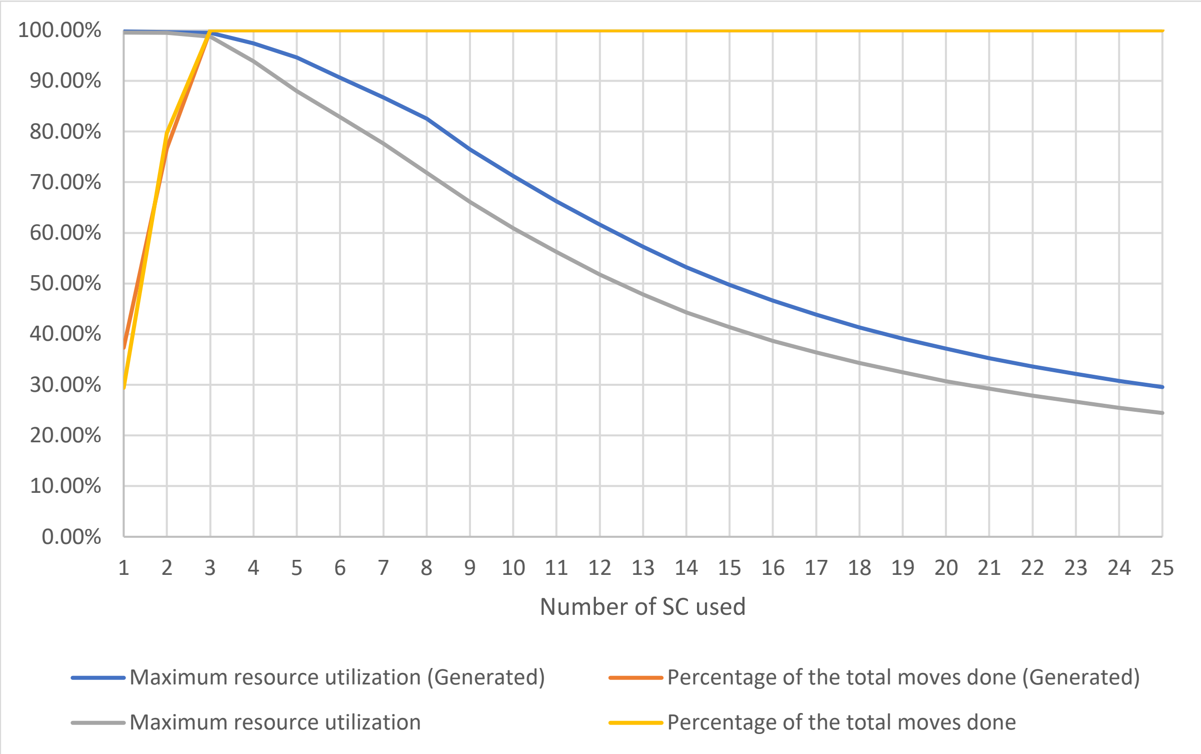


Figure 41. Real and Generated Data (Weibull) analyses for 09505AAT Vessel

In *Figure 41* it can be seen the difference between both analyses. First of all, looking at the *percentage of the total moves done* it can be stated that both are practically the same but there are some minor differences. The reason why this is happening is explained in the previous section 5.5.1.3.

Secondly, looking at the *Maximum resource utilization* it can be stated that values in the generated one are slightly higher than the real ones with an average of 7,34% of difference (without counting less than four SC used because all the movements are not completed).

Finally, having seen both analyses with generated data and their comparison with the one with real data, it can be stated that at least for this vessel the Gamma distribution reproduces the reality better than the Weibull distribution not completely but with a fairly difference.

4.6 Conclusions of the analysis

Similar results for the four vessels studied have been obtained after performing the analyses. All the points to highlight will be presented below:

- The graphs obtained with the real data and the ones obtained with generated data are similar in form. First, in the four cases the *percentage of the total moves done* always reach the hundred percent with the same amount of SCs used. Second, in the four cases the *maximum resource utilization* starts from almost hundred percent and decreases with a steady slope in the beginning and then start to converge at the end.
- The part of the graph that is relevant is after the *percentage of the total moves done* reaches the 100%, because before that not all the movements are completed, therefore, using that amount of SCs makes no sense.
- When both graphs, the one with real data and the one with generated data, are merged a difference can be seen in the *maximum resource utilization* line. In the four cases studied the *maximum resource utilization* with generated data is higher. See below the average difference (note that the average has been calculated with the number of SCs that already reached the 100% of the total moves):

Vessel	Average difference	
	Gamma	Weibull
09710AAT	15,53%	13,19%
09097AAT	8,09%	3,04%
09506AAT	2,60%	1,00%
09505AAT	6,04%	7,34%

Table 12. Average difference in maximum resource utilization

The reason of this difference can be explained by the following. Breaks taken by SC and possible incidents that may affect during the berth are not contemplated while fitting the data. That is because those times are normally large values in the dataset and while validating it they appear as outliers and many of them have been removed to be able to validate the dataset. Consequently, more movements are done in a smaller period of time meaning that the utilization of the resources will be higher.

- Looking at the average difference in *Table 12*, it can be said that in most of the cases the Weibull probabilistic distribution has generated data more similar to the reality than the Gamma.
- When both graphs, the one with real data and the one with generated data, are merged a slightly difference can be seen in the *percentage of the total moves done* line. The reason of this difference can be explained by the following. As explained in *section 4.4.6.2*, a QC normally has a number of bays assigned and it perform all the movements from one bay before moving to the next one. But it can be seen in the real data that this is not always happening. Because of this, when new times are generated and merged with the Benchmark file, the order of the movements may differ and consequently some movements can take more time to perform or less time because of its container position within the yard.
- In these four cases it can be stated that Weibull probability distribution generate values similar to those in the reality.

5. Recommendations and perspective

In this chapter, some recommendations and perspective for future work will be presented.

First of all, working on the limitations of the model could be a good start. I strongly recommend trying to get a more complete license of the software AnyLogic because it could definitely help. Complex process flows, with more blocks on them, could be created because with the student license a limitation of two hundred blocks is present. Adding all the other zones of the yard zone that are not currently in the model would also help to get more accurate results and be able to study those vessels with high number of containers going in that zones. Trying to add the reshuffling movements in the process would also be a good improvement because big vessels could be studied. Finally, another improvement in the model would be positioning the QCs in function of the vessel berthed, therefore the error caused by fixing the QCs positions would be mitigated.

Unloading process flow uses too many blocks because basically the pick-up process of the container has to be duplicated for each QC in the model. I strongly believe that with a bit more of investigation in AnyLogic software and with more knowledge in Java (AnyLogic is based in Java) a simplification can be done and fully take advantage of Anylogic.

The process of data cleaning and creation of the databases needed to run the simulations is, at this moment, mostly manual. That is because not many vessels have been studied and therefore was not worth spending many hours to automatize it. Automatizing this process would lead to a fastest way to perform more simulations in less time and therefore improve this tool for the future.

Finally, I suggest to generate more databases for each case studied and perform the analyses. Since the times are generated randomly following the probabilistic distributions found, performing the same analysis with more datasets would lead to a more robust analysis and therefore reliable conclusions and maybe new ones could be reached. For example, it could be decided if the Gamma probabilistic function generated values more similar to the reality than the Weibull.

6. Bibliography

1. Borshchev, A. (2013). *The big book of simulation modeling.* [Place of publication not identified]: AnyLogic North America.
2. Grigoryev, I. (2014). *AnyLogic 7 in Three Days - A quick course in simulation modeling.* 3rd ed. The AnyLogic Company.
3. Ivanov, D. (2017). *Operations and supply chain simulation with AnyLogic. Decision-oriented introductory notes for master students.* 2nd ed. Berlin School of Economics and Law.
4. Code of Conduct. (2018). The Hague: APM Terminals.
5. Company profile. (2018). The Hague: APM Terminals.
6. Fact Sheet. (2018). The Hague: APM Terminals.
7. Meersmans P J M, Dekker R (2001) Operations research supports container handling. Technical Report EI 2001-22, Erasmus University Rotterdam, Econometric Institute.
8. Daganzo C F (1989) The crane scheduling problem. Transportation Research-B 23B(3): 159-175
9. Avriel, M., Penn, M. and Shpirer, N. (2000). Container ship stowage problem: complexity and connection to the coloring of circle graphs. *Discrete Applied Mathematics*, 103(1-3), pp.271-279.
10. Avriel, M., Penn, M., Shpirer, N. and Witteboon, S. (1998). *Annals of Operations Research*, 76, pp.55-71.
11. Dubrovsky, O., Levitin, G. and Penn, M. (2002). *Journal of Heuristics*, 8(6), pp.585-599.
12. Evers, J. and Koppers, S. (1996). Automated guided vehicle traffic control at a container terminal. *Transportation Research Part A: Policy and Practice*, 30(1), pp.21-34.
13. Imai, A., Nishimura, E. and Papadimitriou, S. (2003). Berth allocation with service priority. *Transportation Research Part B: Methodological*, 37(5), pp.437-457.
14. Kap Hwan Kim (1997). Evaluation of the number of rehandles in container yards. *Computers & Industrial Engineering*, 32(4), pp.701-711.
15. Kim, K. and Kim, K. (1999). Routing straddle carriers for the loading operation of containers using a beam search algorithm. *Computers & Industrial Engineering*, 36(1), pp.109-136.
16. Legato, P. and Mazza, R. (2001). Berth planning and resources optimisation at a container terminal via discrete event simulation. *European Journal of Operational Research*, 133(3), pp.537-547.

17. Meiswinkel, S. (2018). Straddle Carrier Routing at Container Ports with Quay Crane Buffers. *On Combinatorial Optimization and Mechanism Design Problems Arising at Container Ports*, pp.77-107.
18. Murty, K., Liu, J., Wan, Y. and Linn, R. (2005). A decision support system for operations in a container terminal. *Decision Support Systems*, 39(3), pp.309-332.
19. Bruzzone, A., Longo, F., Nicoletti, L., Bottani, E. and Montanari, R. (2012). SIMULATION, ANALYSIS AND OPTIMIZATION OF CONTAINER TERMINALS PROCESSES. *International Journal of Modeling, Simulation, and Scientific Computing*, 03(04), p.1240006.
20. Lau, Y. and Lee, M. (2007). Simulation Study of Port Container Terminal Quay Side Traffic. *Communications in Computer and Information Science*, pp.227-236.
21. Sgouridis, S., Makris, D. and Angelides, D. (2003). Simulation Analysis for Midterm Yard Planning in Container Terminal. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 129(4), pp.178-187.
22. Kondratyev, M. (2015). An object-oriented approach to port activity simulation. *International Journal of Simulation and Process Modelling*, 10(1), p.1.
23. Longo, F., Huerta, A. and Nicoletti, L. (2013). Performance Analysis of a Southern Mediterranean Seaport via Discrete-Event Simulation. *Strojniški vestnik – Journal of Mechanical Engineering*, 9(59), pp.517-525.
24. 2019, P. (2018). *Global container market - demand growth 2019 | Statistic.* [online] Statista. Available at: https://www.statista.com/statistics/253931/global-container-market-demand-growth/ [Accessed 18 Jun. 2018].
25. Aarhushavn.dk. (2018). *Container terminal.* [online] Available at: https://www.aarhushavn.dk/en/terminals/container_terminal/container_terminal.htm [Accessed 18 Jun. 2018].
26. Facts, C. (2018). *Topic: Container Shipping.* [online] www.statista.com. Available at: https://www.statista.com/topics/1367/container-shipping/ [Accessed 18 Jun. 2018].
27. Mathworld.wolfram.com. (2018). *Absolute Error -- from Wolfram MathWorld.* [online] Available at: http://mathworld.wolfram.com/AbsoluteError.html [Accessed 18 Jun. 2018].
28. Mathworld.wolfram.com. (2018). *Relative Error -- from Wolfram MathWorld.* [online] Available at: http://mathworld.wolfram.com/RelativeError.html [Accessed 18 Jun. 2018].

29. Statistics How To. (2018). *Relative Error: Definition, Formula, Examples.* [online] Available at: http://www.statisticshowto.com/relative-error/ [Accessed 18 Jun. 2018].
30. Www-eio.upc.edu. (2018). *Anàlisi de Dades per Transport i Logìstica (ADTL-MSCTM).* [online] Available at: http://www-eio.upc.edu/teaching/adtl/indexcat.html [Accessed 18 Jun. 2018].