AN ALGORITHM FOR SCHEDULING CONTAINERS ON BARGES

public version

Simon Pruijn
s1598937

An algorithm for scheduling containers on barges

Author:	Simon Pruijn s1598937 sepruijn@gmail.com
Supervisor University of Twente:	Dr. Ir. M.R.K. Mes
Supervisor NexusZ:	MSc. M. Glandrup

Date:	March, 1st 2018

EXECUTIVE SUMMARY

INTRODUCTION

Multimodal Container Service (MCS) transports containers using barges between five inland terminals and the Port of Rotterdam. Currently, human planners at MCS need to manually decide which container gets assigned to which barge. Each barge sails according to a predetermined fixed schedule between one of these five inland terminals and the Port of Rotterdam. The Port of Rotterdam contains multiple terminals. It differs every time which terminals are visited to load and unload containers within the Port of Rotterdam.

The current schedules are not optimal, meaning that the barges could move more efficiently. Human planners could benefit from a decision support system. Currently, no scheduling algorithm is available to provide the planners at MCS decision support. In this research we therefore solve the core problem *"Planners have insufficient decision support"* by designing an algorithm and proof of concept to provide the planners at MCS better decision support.

APPROACH

To design our own algorithm, we conducted a literature review, performed a data-analysis, held interviews and used our own expertise.

In the data-analysis we found the following:

1) Large discrepancies exist for the number of containers loaded and unloaded; both within and between terminals.
2) Barges arrive on average 5 hours and 17 minutes earlier at the terminal than they are planned to.
3) Handling time can be forecasted in a linear model with a constant initialization time of 15 minutes and 48 seconds and a variable time of 2 minutes and 26 seconds per container.

The current scheduling process consists of four steps:

1) Use date provided by customer
2) Choose feasible modality
3) Make sure capacity is not exceeded
4) Optimise schedule

Whereby the quality of step 4 depends upon the quality of the human planner and differs largely.

THE ALGORITHM

In our algorithm, the sail schedules between the inland terminal and the Port of Rotterdam are a hard constraint and therefore always preserved. Also, the capacity of the barge is used as a constraint. For our algorithm we define the following five steps:

1) Generating priority matrices
2) Filling the barge
3) Repeat step one and two for the next voyage
4) Optimize the schedule
5) Sequencing

The steps are implemented into a proof of concept. The proof of concept is applied to five pseudo-randomly chosen voyages. In our evaluation we found that different terminals are visited by the human planners than in the schedule generated by the algorithm. Some terminals are visited by the algorithm that are not visited by the human planners and vice versa. This is mainly due to a difference in the input data. We found out that some containers do not show up in Excel when searching on the voyage code that do show up in NLink. This makes it hard to draw a decisive conclusion on the performance of step 1 and 2 of our proof of concept.

When comparing the sequences generated in step 5, we found that planners visit inlets in another, better, order than assumed for the development of step 5. This can easily be implemented by changing the order of the terminals in the list of step 5. Furthermore, we saw that human planners take small numbers of import containers if a barge is already nearby. This can be implemented by adding a rule that allows for earlier pickup of small amounts. Furthermore, we saw some variations with no traceable reason. In conclusion: step 5 of the algorithm can be valuable in decision support with the proposed improvements.

CONCLUSIONS AND RECOMMENDATIONS

With the proof of concept, we proved that a schedule can be generated that is comparable to a schedule made by human planners. However, we did not have access to the same data that is available to the human planners.

We cannot say with certainty whether the developed algorithm is good enough to be implemented. It should be tested in a fair test where human planners and the algorithm have the same input data. However, we did test step 5 elaborately. This step can be implemented immediately in the decision support tool, with the made recommendations.

The proof of concept has only been tested on a sample set of five distinct voyages between Meppel and the Port of Rotterdam. To test dependency effects, series of voyages should be tested. To test if the algorithm can also be applied to other inland terminals than Meppel.

From our research the following recommendations follow:

1) It is important to research how the data integrity can be improved. To improve output quality, it is most important that time windows are registered correctly, and old data is removed from the main database. Furthermore, data should be validated before entering the database.
2) A standard input form can be considered. Now, customers often provide times, locations, weights, using their own forms. This makes it easier to skip or forget data. A standardized form can help improve data integrity.

PREFACE

This research contains the bachelor's thesis "*an algorithm for scheduling containers on barges*" to complete my bachelors program Industrial Engineering and Management at the University of Twente.

I want to thank my company supervisor, Maurice Glandrup, and my supervisor at the University of Twente, Martijn Mes, for their patience on my journey and great feedback on concept versions of this report. I would like to thank Jacob Boorsma, manager barge and truck planning at MCS, for the great insights he provided in the interview. Furthermore, I would like to thank Saskia Hidding for her support on writing and grammar.

You are reading the public version. For questions the author can be reached on sepruijn@gmail.com.

Simon Pruijn
Enschede, March, 1st 2018

CONTENTS

Executive summary ... iii
Preface ... v
List of definitions ... vii
1 Introduction ... 1
2 Systematic literature review ... 7
3 Context analysis ... 15
4 Solution design ... 29
5 Evaluation Proof of concept ... 41
6 Conclusions, recommendations and discussion ... 47
7 References ... 51
Appendix A: Lists of barges and terminals ... 53
Appendix B: Problem identification and problem cluster ... 54
Appendix C: Lists of Key Concepts ... 55
Appendix D: Search strategy literature review ... 56
Appendix E: Summary of content articles literature review ... 57
Appendix F: Interview form ... 59
Appendix G: Analysis waiting times, handling times and difference actual time and planned time ... 60
Appendix H: Regression ... 62
Appendix I: Epoch times ... 66
Appendix J: Input list for terminal sequencing ... 67
Appendix K: Comparison of the algorithms ... 68
Appendix L: Schedule sequence comparison ... 73
Appendix M: Step 5 sequence comparison ... 78
Appendix N: Barge utilization ... 83

LIST OF DEFINITIONS

Barge Ship used for transporting containers at inland waterways.

Call An order of a customer containing information about how many containers should be shipped at a certain moment from point A to B.

Handling Time The time a barge is placed under the crane up until the barge leaves the terminal. In this time containers are loaded and unloaded.

Heuristic A rule or set of rules to find a solution to a problem. A heuristic is an approach to solve the problem with a practical methodology, but does not guarantee to find the optimal solution.

Median The middle value in a row of values. If the amount of data in the sample set is an even number, the median is the average of the middle two values. For example, in the dataset {2, 2, 4, 5, 12}, 4 represents the median.

Mean The mean is the average value of a sample set. In the case of the sample set {2, 2, 4, 5, 12}, the mean is (2 + 2 + 4 + 5 + 12) / 5 = 5.

Modality 1.Way of transportation, e.g. barge, truck or train.
2. Software MCS uses to register calls and corresponding data.

NLink Name of the web interface containing information about containers, barges and schedules for the voyages containers make.

onTerminal Name of the digital field containing the time at which an export container should be delivered or an import container should be picked up.

Planner Person who schedules the containers.

Planning *(verb)* Long term assignment of resources.

Offline planning The planning that takes place before a vehicle has started the motion (Shiller, 2015).

Online planning Planning that takes place after a vehicle has started its motion (Shiller, 2015).

Sail schedule The schedule for barges, containing when which barge leaves and enters the inland terminal, and when which barge leaves and enters the Port of Rotterdam. The sail schedule is a timetable comparable to that of a bus line.

Scheduling The process of determining the sequential order of activities, assigning planned duration and determining the start and finish times of each activity.

Terminal A place where containers are loaded, unloaded and stored.

TEU Twenty-feet Equivalent Unit, the size of a container with a length of 20 feet.

Time Window Defined period of time in which an action can happen. For example the pick-up of delivery of a container.

VBA Abbreviation for Visual Basics for Applications. A programming language developed by Microsoft to be able to program macros for programs such as Excel.

Vessel Ship used for carrying freight on sea. Opposed to a barge, which is used for inland transport, a vessel is used for intercontinental transport.

Voyage Trip of a barge from an inland terminal to the Port of Rotterdam (also called export voyage) or trip of a barge from the Port of Rotterdam to an inland terminal (also called import voyage).

Waiting Time The time between a barge arrival and the barge is placed under a crane. In this time a barge waits for loading and unloading.

1 INTRODUCTION

In this chapter we introduce the problems with scheduling containers on barges. First, we explain the current situation in Section 1.1 in a qualitative way. In Section 1.3 we name our core problem and explain the path towards solving it. In Section 1.4 we name the means and currently available material for this research. To conclude in Section 1.5 we lay out the research objective, research questions and approach, providing a road map for this research and an overview of the questions per chapter.

1.1 CURRENT SITUATION

This assignment is commissioned by *NexusZ*, a company that makes client-driven software for transport companies. One of the companies that uses the software of NexusZ is *Multimodal Container Service (MCS)*. MCS calls itself Full Service, meaning that they provide the entire routing solution between the Port of Rotterdam and the customer and decide themselves what vehicles they use for transport, while respecting the customer's wishes. MCS uses trucks and barges to transport containers. Large parts of the planning process are done manually by five human planners who take care of the customers' wishes.

To delimit this bachelor assignment, the scope is limited to barge planning. A barge involves a major capital investment and the daily operating costs are high. This means that improving barge utilization can be translated into significant financial improvements. Improving barge utilization also means that there are less barge voyages or less trucks needed to ship all the containers. This potential reduction in transport operations can reduce the damage to the environment.

MCS transports, according to their website, around 100,000 containers per year. MCS uses barges to ship these containers between Meppel, Groningen (Westerbroek), Leeuwarden, Harlingen, Kampen and Rotterdam. The assignment of these containers to barges is done by the human planners. Currently, NexusZ is designing a web interface named *NLink* to provide an overview and support decision making. Based on this information the planners can decide when to ship which container on which barge. The quality of the generated schedules completely relies on the experience of the planners. Every barge has a capacity of 156 TEU, which is equal to 156 20-feet containers.

NLink offers various interfaces comparable to dashboards to monitor the container transport. These interfaces are accessible via the menu in the upper left corner. One of these interfaces is the *shipper manifest*. Figure 1 shows the shipper manifest interface. On the left side of the window a table containing all terminals that are scheduled to load and unload containers for the selected voyage is visible. Under '*Laden en Lossen*', the schedule for the Port of Rotterdam is shown. In the first column the terminals are displayed. Behind these terminals columns are displayed with the arrival time, the leave time (in green), the planned and actual numbers of unload and load containers. In the last column notes from the shipper are displayed. If the terminal name is clicked, also the planned time for that terminal shows up on screen. A map on the right shows the current location of the barge.

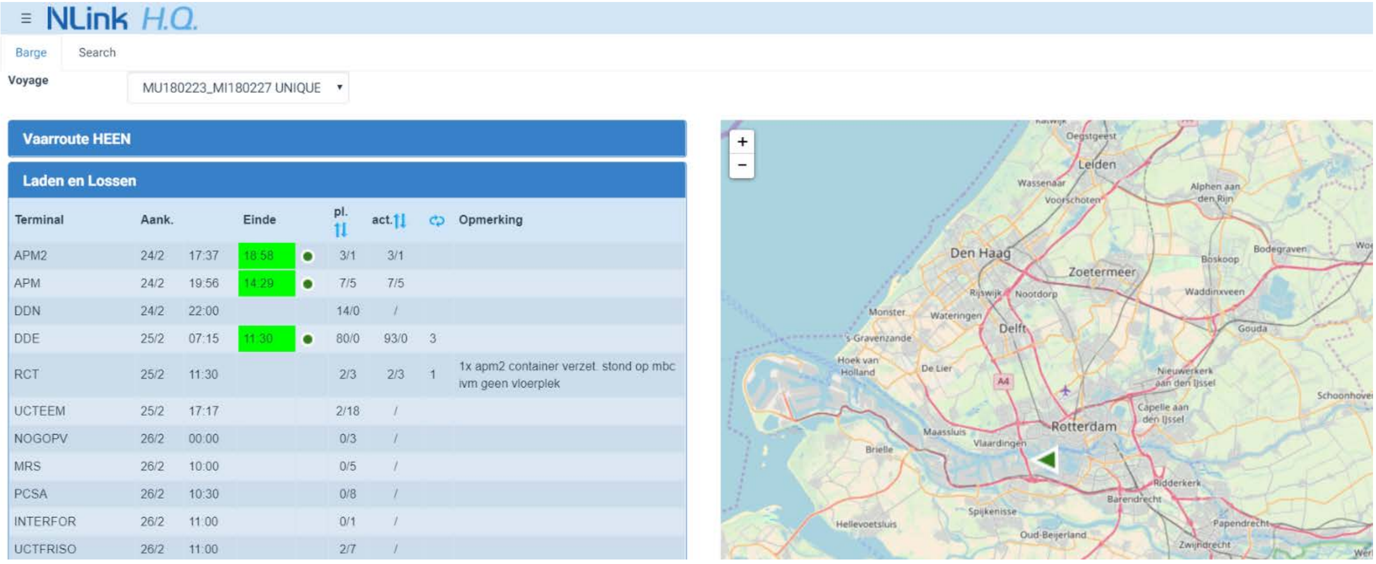


Figure 1: The NLink interface

MCS uses 13 barges to transport the containers. These containers are transported between Rotterdam and the five inland terminals owned by MCS and close partners. Transportation also takes place between the several terminals within the Port of Rotterdam. A barge usually does not sail between multiple inland terminals. From the inland terminals, the freight can be transported further to the Netherlands or to other countries. Figure 2 shows the inland terminals indicated in blue and the outland terminals indicated in red. Appendix A contains two lists; one with the names of the barges and one with the terminals.

Figure 2: terminals MCS uses

Currently, human planners need to decide which container gets assigned to which barge. They plan barges to unload and load their freight at a terminal at a specific time and assign those containers to these barges. The barges sail according to a predetermined fixed schedule between one of the inland terminals and the Port of Rotterdam. The Port of Rotterdam contains multiple terminals and it differs every time which terminals a barge visits within the Port of Rotterdam. The planners need to consider many aspects. For example, to pick up a container, the container should be present, and the timeframe should also match the wishes of the customer. Furthermore, there are many factors that influence the realised transportation time, such as waterway congestion, waiting times at terminals and weather conditions.

A combination of online and offline planning takes place in the planning process. Offline planning is the planning that takes place before a vehicle has started the motion, online planning is the planning while the vehicle is making its cycle (Shiller, 2015).

The current schedules are not optimal, meaning that the barges could move more efficiently. This inefficient transport includes unnecessary delays, extra resources needed, excessive truck transport and poor utilization of the resources. Currently, no scheduling algorithm is available to provide the planners at MCS decision support.

1.2 PRACTICAL INFORMATION

In this section we explain various concepts that are important throughout this document. One of the concepts used in this research is voyage. With voyage an import or an export voyage can be meant, or a combination of those. An import voyage is a barge sailing from the Port of Rotterdam towards the inland terminal for importing containers. An export voyage is a barge sailing from the inland terminal towards the Port of Rotterdam for exporting containers. Each has a voyage code containing an indication for the inland terminal, an indication for importing or exporting and the date the barge enters or leaves the inland terminal.

A customer can order a single trip or a round trip. A round trip means that the emptied container is also returned to the terminal it was picked up. A round trip is double the costs of a single trip and is put separately into the database. Therefore, we do not distinguish between round trips and single trips in this research and treat everything as a single trip.

A customer provides a time and date they want the container(s) to be picked up. This is put in the database as the $onTerminal$ time.

1.3 PROBLEM DISCRIPTION

A company can have several problems. A problem cluster is a helpful tool to map problems and their relationships (Heerkens and Van Winden, 2012).

At MCS the planning process does not have a decision support system that can generate schedules automatically. The planning process is complex. Many possible solutions and much information can be used in the planning process. At this moment, more information is available than human planners can handle. There is a trend that the availability of cargo information goes to *Just in Time (JIT)*. There are also more aspects in cargo and infrastructure to consider than before, online as well as offline. Aspects to consider are, amongst others, the congestion on waterways, vessels that are too late, bad weather that prohibits a smooth handing of containers in the harbour, strikes, a vessel that decides to go to another harbour and reserved timeslots on terminals which are changed. Since humans cannot incorporate all these factors within reasonable time, the plans for transport are not optimal, which means that there is inefficient transport, leading to a competitive disadvantage for MCS. Appendix B contains a list with all the problems identified for the construction of the problem cluster.

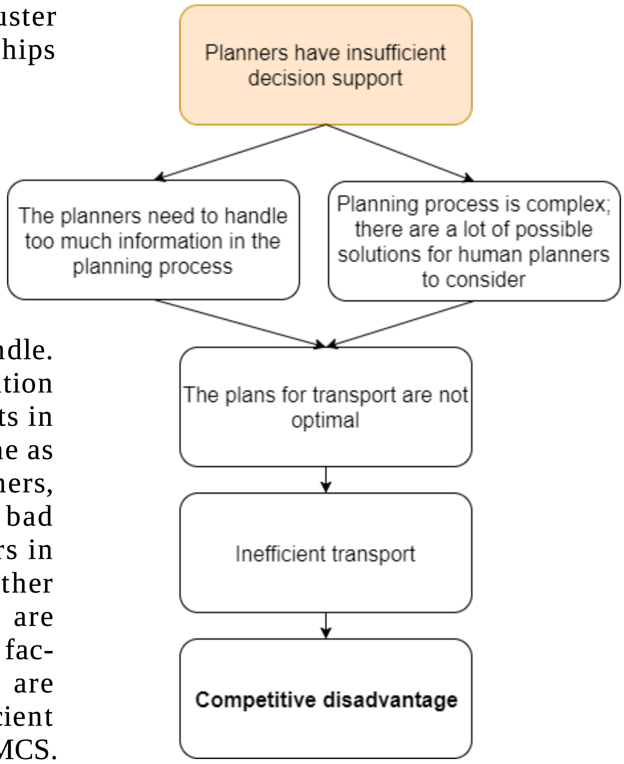


Figure 3: the problem cluster, core problem is indicated in salmon colour.

Figure 3 shows the problem cluster, which is mainly drawn up in consultation with the company supervisor and university supervisor. As derivable from Figure 3, the core problem is defined as follows:

> "Planners have insufficient decision support."

A decision support system can support planners in doing their tasks. It can calculate many possible solutions and can take a lot of information into account while doing that. Therefore, the main question of this research is:

> "In what way can planners use decision support and to what performances can this lead?"

1.4 MEANS AND CURRENTLY AVAILABLE MATERIAL

Several means are available to complete this research successfully. One of the means is the current software NLink, which has a test account to see how the software works and how the available data is interfaced towards the planners. Besides, every hour an Excel sheet is generated with the current schedule containing all information that is available at that moment in the system, including, but not exclusively, the voyage number, terminals, times wished by the customers, and planned times. The system also generates an Excel sheet with the current orders on an hourly basis. These sheets provide insights in how human planners plan and think, since every hour a new sheet is generated. Furthermore, the Excel sheets can be used to build and test algorithms of which the logic can be implemented in NLink.

Information can also be obtained using conversations on demand with the company supervisor, expertise knowledge from the UT-supervisor and conversations at MCS, in consultation with and via the supervisor at NexusZ.

The optimal solution would be one that provides an integrated solution with the current web interface that provides useful information and is able to provide planning proposals. However, due to the time constraint, this projects' prototype will be programmed in Excel. Parts of the logic and algorithm can be converted later to the current web interface, since this research bears upon the logic behind the code and not upon the code itself.

Existing bachelor and master theses that are made earlier for NexusZ contain proposals for useful heuristics. We will identify these heuristics and analyse their strengths and weaknesses. Furthermore, it is important to look at which constraints exist in barge scheduling.

1.5 RESEARCH OBJECTIVE AND QUESTIONS, KNOWLEDGE PROBLEM AND APPROACH

As we recall from Section 1.3, the main question in this research concerns: *"In what way can planners use decision support and to what performances can this lead?"*. The current version of NLink already offers a lot of data and data visualisation. Therefore, to solve the core problem *"Planners have insufficient decision support"*, we develop an algorithm that helps the planners in faster scheduling of containers on barges by automating (parts of) the process. We formulate the main research objective as follows:

> "Design an algorithm and proof of concept to provide the planners at MCS better decision support."

KNOWLEDGE PROBLEM

A knowledge problem is a question that needs to be answered to solve the core problem. To recall our core problem is that planners have insufficient decision support. To solve this, we are designing an algorithm and proof of concept to provide the planners at MCS better decision support.

To be able to design this algorithm, a heuristic is needed. A heuristic consists of a series of steps that can be followed to generate a feasible solution. Heuristics search for a local optimum. This is faster than generating all possible solutions and selecting the best one. A heuristic does not necessarily find a global optimum. We explain optima in depth in Section 2.3.

Because it would take too much time to program all possible heuristics, a selection of potential interesting heuristics that can be used in an automated planning algorithm is made.

The knowledge problem is formulated as the following question:

> "What is a good heuristic to plan the containers on barges?"

To answer this question, the following sub questions are formulated:

- What procedure is used by planners in their manual planning?
- Which heuristics are available in literature that might be applicable?
- What are the strengths and weaknesses of these heuristics?
- What are the ideal conditions under which the heuristic can be applied?

"Good" is a widely interpretable word. In this research question, it will be measured on the following aspects:

- Expected performance with respect to call size, number of terminals visited and lateness.
- Time it takes for the algorithm to create a planning (not too high).

The aspects researched possibly influence each other. A heuristic that optimizes on call size would probably lead to more lateness, since it considers call size as more important than lateness. A heuristic trying to find minimal lateness, however, would probably lead to less lateness but to a higher call size. Hence a trade-off should be made between outcomes.

RESEARCH QUESTIONS

Various research questions follow from the research objective. We divide each question into sub research questions and assign those to a chapter. This division gives us the following overview of this research:

Chapter 2: What are possible solutions for the barge scheduling problem that can be found in literature?
2.1 What are key theoretical concepts within barge scheduling?
2.2 Which of these concepts are applicable to the container scheduling problem at MCS?

We answer these questions by performing a literature study and studying theses in the same research field.

Chapter 3: How can an automated scheduling algorithm contribute to better decision support at MCS?
3.1 How are schedules currently made and what steps are taken?
3.2 At what points in the process are decisions made?
3.3 What output should the solution provide?
3.4 Which requirements should the solution fulfil?

We answer these questions by gaining knowledge from literature and interviews at MCS and NexusZ.

Chapter 4: How should the automated scheduling algorithm work?
4.1 Which approach can be used to come to the desired output?
4.2 Which simplifications and assumptions need to be made?
4.3 What should the model look like?

Based on knowledge gained from literature and interviews at MCS and NexusZ, we design a suitable scheduling algorithm.

Chapter 5: What performance can be expected when implementing the algorithm?
5.1 How can we test whether the developed algorithm generates an acceptable solution?
5.2 What are the advantages and disadvantages of the developed algorithm?

We answer these questions by testing the developed algorithm on various datasets, evaluating the outcomes and comparing them (numerically) with real-world schedules.

2 SYSTEMATIC LITERATURE REVIEW

This chapter contains a systematic literature review. A systematic literature review is a scientific repeatable way of gaining knowledge on previous research. As stated in Section 1.5, we need to develop a good heuristic to be able to make an automated planning. Therefore, the knowledge question is formulated as:

> "What is a good heuristic to plan the containers on barges?"

To partially provide an answer to this research question and the accompanying sub questions mentioned in Section 1.5, we identify heuristics that are available in literature that can be applied to planning containers on barges.

This chapter focuses on literature about scheduling. In Sections 2.1 to 2.3 we conduct a systematic literature review. In Section 2.4 we discuss literature gathered in an unsystematic way. The following sub questions as mentioned in Section 1.5 will be answered:

- Which heuristics are available in literature that might be applicable?
- What are the strengths and weaknesses of these heuristics?
- What are the ideal conditions under which the heuristic can be applied?

We do this in the order or our research questions:

- What are key theoretical concepts within barge scheduling?
- Which of these concepts are applicable to the container scheduling problem at MCS?

First we define the key theoretical concepts, then we search literature about these concepts and discuss how it can be applied to barge scheduling.

Section 3.2 in the next chapter provides more information about how planners work in practice and what heuristics planners currently use to answer the sub question of the knowledge question: "*What procedure is used by planners in their manual planning?*".

2.1 KEY THEORETICAL CONCEPTS IN BARGE SCHEDULING

To conduct a systematic literature review, it is necessary to start with defining the key theoretical concepts. To gather general knowledge about barge planning, several articles have been consulted. For its high Field-Weighted Citation Impact of 43.69, the article of M. Christiansen (2004) has been consulted. Additionally, the master thesis of L. Baranowski (2013) contains a list with key concepts related to planning and scheduling at a company comparable to MCS. Furthermore, Winston (2004) has been consulted to review any missing key concepts. The terms in these lists can be found in *Appendix C: Lists of Key Concepts*.

The found key theoretical concepts can be grouped into categories. By looking at similar key concepts, we distinguish the following list of categories, which are relevant to this literature review:

- Scheduling
- Planning
- Mathematical Programming
- Mathematical Model
- Simulation
- Constructive heuristic
- Meta-heuristic

2.2 SEARCH STRATEGY

We use the database *Scopus* for this search. Because of the limited time available for this research, if more than 100 results are found, only the first 100 results will be considered while search results are sorted on highest citation rate. For the search through the database, we formulate inclusion and exclusion criteria (listed in Appendix D, Table 11).

Based on the inclusion and exclusion criteria, we generate a search protocol to search through Scopus to find literature about heuristics to schedule containers on barges. The performed bibliographic search is depicted in Appendix D, Table 12.

2.3 EVALUATION

The articles found using the method described in Section 2.2 are assessed in a systematic way. Table 1 on page 9 shows an overview containing the *used method*, *main concepts* and an *evaluation* of each article. The main concepts are numbered, and a cross indicates that the article discusses *(1) scheduling*, *(2) usage*, *(3) call size*, *(4) number of terminals visited* and/or *(5) timeliness*. Furthermore, it shows the chosen performance indicators if a quantitative analysis has been performed in the article.

ALGORITHMS FOUND

A wide variety of algorithm types are found in this literature research. In barge planning, not one specific heuristic seems to be widely applied. Many authors use their own developed heuristics or provide a summary of research done before. Appendix E shows per article the found content. Below we provide a general overview.

The algorithms found can be divided in two main categories: constructive heuristics and improvement heuristics. The construction heuristics described are several variations of insertion heuristics and linear programming. Examples of improvement heuristics are k-opt, simulated annealing and tabu search. Authors who wrote about simulated annealing or tabu search generated good solutions in a large variety of fields, both in terms of time as in terms of optimality.

Author(s)	Title	Year	Used Method	Concepts (*) 1	2	3	4	5	Evaluation	Chosen performance indicators
Solomon Marius M.	ALGORITHMS FOR THE VEHICLE ROUTING AND SCHEDULING PROBLEMS WITH TIME WINDOW CONSTRAINTS.	1987	Savings, savings waiting time limit, nearest neighbour, sweep heuristics. 3 different insertion heuristics	x				x	mainly taking into account time windows. Insertion criterion 1 performed best.	Schedule time
Potts C.N., Kovalyov M.Y.	Scheduling with batching: a review	2000	Machine-scheduling (EDD, batch methods)	x					Does not seem to be applicable to barge planning	Total lateness
Solomon Marius M., Derosiers Jacques	TIME WINDOW CONSTRAINED ROUTING AND SCHEDULING PROBLEMS.	1988	k-opt heuristic	x					mainly a summary of research done before and gives a development over time	
Cheng T.C.E., Gupta M.C.	Survey of scheduling research involving due date determination decisions	1989	CON, RAN, TWK, SLK, NOP, JIQ, JIS, PPW	x	x			x	This paper is taking the due date as a decision variable and choosing the best way to fit it	Due date, Usage, lateness
Magnanti T.L., Wong R.T.	NETWORK DESIGN AND TRANSPORTATION PLANNING: MODELS AND ALGORITHMS.	1984	Linear programming				x		Mainly focussed on routing problems and designing the network. Knap sack, TSP and LP are mentioned	
Applegate David, Cook William	Computational study of the job-shop scheduling problem	1991	Own heuristic procedure for finding schedules with adding a shuffle	x					Explaining their way of solving a job shop problem, no comparison with other algorithms	computation time, difference heuristic and optimal value
Osman I., Potts C.	Simulated annealing for permutation flow-shop scheduling	1989	Different variations on simulated annealing compared to the NEH heuristic	x					There are different types of simulated annealing (ordered search and unordered search, switch neighbour and insert neighbour). In this case an SA(Switch, Random Search) gave the best results	best solution
Dell'Amico M., Trubian M.	Applying tabu search to the job-shop scheduling problem	1993	tabu search based algorithm	x					They made an algorithm and ran it on a set of 53 benchmark problem instances. In many cases they found the (near) optimal solution. Not a direct link towards barge scheduling.	Closeness to optimum, computation time compared to global optimum
Widmer M., Hertz A.	A new heuristic method for the flow shop sequencing problem	1989	Their own developed SPIRIT method	x					A method that can be used for quick but not precise estimation	Closeness to optimum, computation time

Table 1: Evaluation of the literature review. (*) Concepts are numbered to save space. 1. Scheduling, 2. Usage, 3. Call size, 4. Number of terminals visited, 5. Timeliness

Local and global optima

To further illustrate how the optimization techniques simulated annealing and tabu search work, a reader should first be aware of the difference in local and global optima. An optimum is the best possible result. In terms of money this often concerns the minimization of costs or the maximization of revenue. Other examples of optima applied to barge scheduling are the maximization of barge utilization or minimization of terminal visits.

A global optimum is the best of all possible solutions. In the case of revenue, the global optimum would be the highest peak in Figure 4. A local optimum is a point where all neighbours provide a worse output than that local optimum. A neighbour is an input of almost the same settings. In our example in Figure 4, we see that if we move slightly left or right from the red arrow, we end with a worse solution. A local optimum can be, but is not necessarily, a global optimum.

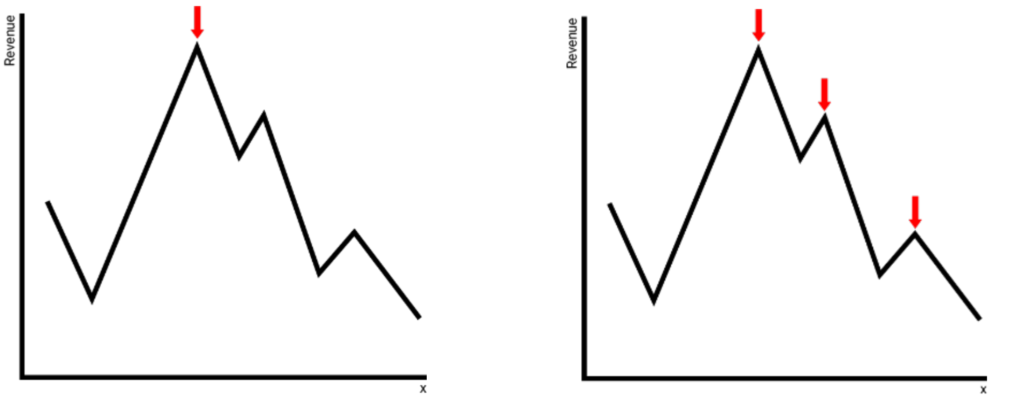


Figure 4 Left: the red arrow indicates the global maximum, Right: the red arrows indicate local optima.

Several optimization techniques exist to find the optima. Figure 5 shows the basic idea for local search methods. We start with the red arrow in the figure on the left. By moving to the left we move to higher revenue until we find a point where both neighbours are lower than the current point. By using this local search method we find a local optimum.

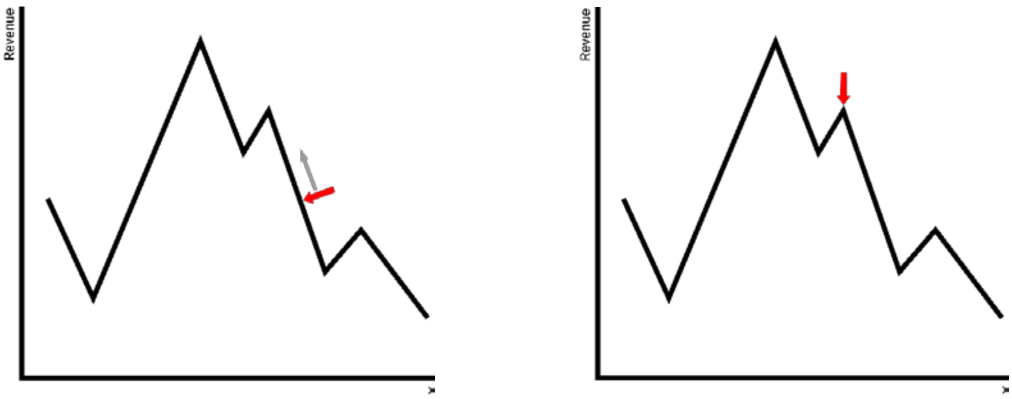


Figure 5 Local search method, adjusted from Krul (2016).

Simulated annealing

One approach that is often used to escape local optima is simulated annealing. In this way simulated annealing can get closer to a global optimum but does not necessarily find it. Simulated annealing is a method to reach a global optimum in a probabilistic way.

> "The idea [of simulated annealing] is that solutions leading to a worse objective value compared to the current solution, should not be denied in all cases, but are sometimes accepted to find a better local optimum, or even the global optimum."
>
> (p. 29, Krul, 2016)

Simulated annealing has its analogic origin in the cooling of steel. Simulated annealing consists of two steps. First a random but feasible solution is generated. Then this solution is optimized. This optimization is done using a so-called temperature. This temperature is high at the beginning and decreases every step. With the decrease of this temperature, also the chance of accepting a worse solution and thereby escaping a local optimum is decreased. In the beginning the acceptance probability is close to 1, while at the end the acceptance probability of accepting a worse solution reaches almost 0.

A more extensive explanation on simulated annealing is given by Krul (2016).

Tabu search

Tabu search is like simulated annealing another method to escape local optima. It was introduced in 1986 by Fred Glover. The basic principle of tabu search is to pursue the search whenever a local optimum is encountered by allowing non-improving moves. (p. 169, Gendreau and Potvin, 2005)

Tabu search chooses from all its neighbours the best solution that is not the Tabu list. The tabu list contains solutions that have been evaluated previously. The choice for another solution that is not in the tabu list is made, even if this solution is worse than the current solution. The algorithm can stop after a fixed number of iterations or after several iterations without improvement. The computer then takes the best solution found. A tabu search can be commonly noted in the following way (adapted from Gendreau and Potvin (2005)):

Notation:

- S the current solution,
- S^* the best-known solution,
- f^* value of S^*,
- $N(S)$ the neighbourhoof of S,
- $\tilde{N}(S)$ *the 'admissable' subset of* $N(S)$ (i.e. non-tabu or allowed by aspiration)
- T tabu list.

Initialization:
Construct an initial solution S_0.
Set $S := S_0$, $f^* := f(S_0)$, $T := \emptyset$.

Search:

```
While termination criterion not satisfied Do
       select S in argmax[f(S')];  //S' is part of Ñ(S)
       If f(S) < f* Then
              f* = f(S);
              S* = S;
              record tabu for the current move T (delete oldest entry if necessary);
       End If
End While
```

In this algorithm, argmax returns the subset of solutions in $\tilde{N}(S)$ that maximizes f.

EXPLANATION OF SEARCH RESULTS

In our systematic literature research, we did not find many articles about algorithms applied to barge planning or scheduling. One of the reasons therefor could be the following:

> "Barge planning is originally done by employees with great naval experience, who are not open to technological advancements. With the change in the sector of coming in extra people with an academic background this seems to change, but at this point in time it seems that we are not there yet."
>
> (p. 13, Christiansen, 2004)

Another reason could be that this research area mainly takes place at companies where this is part of the core business. Therefore, much research into the exact algorithms and heuristics is confidential. Furthermore, another reason could be that we filtered on high citation rates, while applied research is cited less than abstract research.

During our research, we also found literature in an unsystematic way. In the next section we briefly discuss this literature and their relevance to our research.

2.4 SIMILAR LITERATURE IN THIS FIELD

In addition to the systematic literature review, it is important to also look at literature similar to this research. This comparable literature can be applied more directly to this thesis's field of research. In our systematic literature review we found mostly abstract concepts for algorithms, because the more abstract theories are cited more often. However, the abstract theories are hard to apply directly to our problem. Therefore, we try in this section to supplement the abstract theories with more applied research.

Three theses have been made for NexusZ in the near past. One of these is written by David de Meij (2014) who analyses what data can be used and what data needs to be collected for future use in barge scheduling at *Combi Terminal Twente (CTT)*. CTT is a multimodal transport company, meaning they use barges, trucks and trains to transport containers. In this way, CTT is like MCS, with the difference that MCS does not offer transportation by train.

The thesis of De Meij does not focus on an algorithm, but focuses more on data gathering, data integrity and data visualisation. De Meij also conducted a literature study and concludes that similar problems to barge scheduling can be found in the Traveling Salesman Problem as described by Lin & Kerninghan (1973) and in the Vehicle Routing Problem as described by Christofides (1976).

Another thesis is written by Inge Krul (2015). The goal of this thesis is to "*give support to the truck planners at CTT with scheduling to improve the performance of container transport*". Krul researched which Key Performance Indicators (KPI) are relevant in container transport. She finds the following KPIs important for customers:

Not in time
Number of containers that are not loaded at the loading/discharge time.

Time too late
The amount of time containers are too late in the KPI mentioned above.

Time not in time window
Lateness outside an assumed soft time window. Customers provide a hard loading/discharge time, but a certain amount of time outside this hard time provided by customers is considered as on time in this KPI.

Not in time window
Total number of containers delivered outside the soft time window. No distinction is made between 2 minutes or 2 hours too late.

Krul considers the following KPI's to be important for transport companies:

- Total number of trucks
- Travel time
- Waiting time
- Number of detours (moves where a truck does not transport a container)
- Total time of detours

Krul develops in the *Plant Simulation* software from Siemens a simulation model to test several experimental factors, under which the start- and stop temperatures and cooling factors for simulated annealing. She also experiments with the probability of choosing an operator (crossing, moving, or swapping a depot) and with the number of jobs that are swapped or moved with that operator.

The thesis of Lina Baranowski (2013) introduces a model that assigns containers to barges in a priority-based system. She prioritizes barges based on how many days in the time window are left and whether the barge is entering or leaving the Port of Rotterdam. Then she proposes several filters and different ways of assigning containers to barges. Since she found the filter containing a priority and distance-based algorithm performed best, we will focus on that approach. The problem Baranowski solves with this algorithm shows many similarities with the problem described in this thesis. However, the model she built contains an objective function containing only the minimization of late arrivals and pickups, while we also want to optimize upon other factors. Furthermore, she formulated her priorities in terms of days, making it harder to apply to voyages with irregular intervals. In our algorithm design we develop an algorithm dealing with these weaknesses.

All theses mentioned in this section are publicly available to read at http://essay.utwente.nl.

2.5 CONCLUSION

The goal of this literature research chapter was to find an answer to the following question:

> "What are possible solutions for the barge scheduling problem that can be found in literature?"

Several construction heuristics are available. The most relevant construction heuristic we found in literature is an insertion heuristic. This heuristics' strength is that it provides a feasible result in a reasonable amount of time. However, the weakness is that the result is not proven to be (close to) optimal, so therefore it is advisable to use it in combination with an improvement heuristic. The classical case to apply this heuristic is in truck-routing problems without very narrow time windows, so it is debatable how useful this heuristic is in the barge planning problem.

Two main improvement heuristics can be found in literature that can be applied to optimization for barge scheduling: simulated annealing and tabu search. Both algorithms can escape from local optima. The main difference is that tabu search holds a list with solutions that already have been evaluated, and therefore does not come back to the same point as long as it is on the tabu list.

We also evaluated three former written theses. In these theses we found that L. Baranowski developed a priority-based algorithm. We use this algorithm as inspiration in Chapter 4 at developing our algorithm.

In conclusion, we did not find a decisive answer in this literature research to our question. Since no decisive answer exists in literature, we formulate the construction algorithm based on some aspects found in this literature review, the experience within the company and the university, insights we gain from observing the data and common sense.

3 CONTEXT ANALYSIS

In this chapter we answer the question *How can an automated scheduling algorithm contribute to better decision support at MCS?* To provide an overview of how barge scheduling works, several interviews have taken place with the company supervisor at NexusZ and the manager barge and truck planning at MCS. The structured interview form at MCS (in Dutch) can be found in Appendix F.

First, we look at the quantitative aspect of scheduling containers on barges. We analyse data gathered over several months, to gain insights in container amounts, waiting times and handling times. In the second part of this chapter we look qualitatively at how human planners currently come up with a feasible schedule.

Note: in the public version some terminal names, tables and graphs have been censored. The terminal names are censored per paragraph, and do not have a common letter.

3.1 DATA ANALYSIS

In this section we perform a quantitative analysis upon data from the database. First, we look at averages and spread for the containers per terminal for each voyage, then we want to know more about the number of containers per day, and finally we analyse waiting and handling times at terminals.

CONTAINERS PER TERMINAL

To analyse how many containers MCS transports and how they are distributed, we look at a shipper manifest Excel-sheet. This sheet contains data of actual voyages, alongside with planned times and loading and unloading amounts. We look at *StartDates* (the time a barge arrives at the terminal and starts the queue) between 26-09-2017 and 20-11-2017 and Planned times between 1-10-2017 and 22-11-2017 after clean-up. These data ranges are chosen for the pragmatic reason that this is all data that is available at hand. In the clean-up of the sheet we removed all rows that did not have a moment registered for the actual unload or actual load time. Furthermore, all rows without a *StartDate* are removed. Then we created a pivot table from the data to come to the following analysis.

Table 2 shows the average number of containers that is loaded and unloaded at each terminal stop in a voyage. It shows the planned numbers and the actual numbers for the considered period. It also shows the standard deviation and the maximum and the minimum numbers entered in our dataset. We see that, on average, more containers are actually loaded and unloaded than planned.

	Containers Unload Planned	Containers Load Planned	Containers Load Actual	Containers Unload Actual
Average	7.89	8.87	8.65	9.25
Standard Deviation	12.82	15.00	13.90	15.25
Maximum	106.00	137.00	124.00	137.00
Minimum	0.00	0.00	0.00	0.00

Table 2 containers loaded and unloaded per terminal at a voyage.

Table 3 shows the average amount of containers to unload and load which are planned, and the average amount of containers to unload and load that are actually loaded and unloaded. As we see, there are large differences between terminals. Some unload a small number of containers, such as the terminal WHT, but then load on average a relatively high number of containers. In the case of Terminal A this is 11.07. There are also terminals that are more balanced in the loading and unloading amounts, such as Terminal B.

Terminal	AVG Unload Planned	AVG Load Planned	AVG Unload Actual	AVG Load Actual
[redacted]				

Table 3 Average Load and Unload containers per terminal

Figure 6 shows the actual numbers of loaded and unloaded containers per terminal in a graphical representation. As stated before, we see that terminals rely often heavily on either loading or unloading. Terminal C is an exception to the rule, here we see two bars of around the same length, showing that here loading and unloading is almost equal. The reason for this is that Terminal C actually covers the function of depot, meaning that containers can be delivered and picked up for storage.

Figure 6 Unloaded and Loaded actual number of containers average per voyage

CONTAINERS PER DAY

We also want to know how many containers are transported each day by barge. Table 4 shows how many containers are planned, and actually unloaded and loaded. From the data set we derive that each day around 190 containers are unloaded and 201 containers are loaded. To readers it might seem illogical that the number of containers loaded is not equal to the number of containers unloaded. However, the dataset does not contain all containers since we removed rows with insufficient data for proper analysis. Furthermore, the dataset only covers data from a limited time horizon, which might also cause a shift. All these factors make that a difference in loading and unloading can exist.

The average of 201 containers per day is fewer than the 100,000 containers MCS states to ship yearly in Chapter 1, since 201 x 365 = 73,365. However, the dataset we considered in this chapter has been modified by removing rows with missing data. Also, the sample is from voyages planned to load and unload between 1-10-2017 and 22-11-2017. There might be a fluctuation between months that we are not able to research using this data. Furthermore, the number stated in Chapter 1 also includes containers transported by trucks, while we consider trucks outside the scope of this research.

	Unload Planned	Load Planned	Unload Actual	Load Actual
Total Dataset	9101	10224	9858	10448
Average per Day	175	197	190	201
Average per Voyage	80	90	86	92

Table 4 totals of containers and containers per day

HANDLING AND WAITING TIMES AT TERMINALS

To analyse waiting times, handling times and the difference between planned and actual arrival times, we use the same Shipper Manifest data sheet, but clean it up in a slightly different way. We define waiting time to be the time between a barge arrival and the barge is placed under a crane. In this time a barge waits for loading and unloading. We define handling time as the time a barge is placed under the crane up until the barge leaves the terminal. In this time containers are loaded and unloaded.

In Excel we filter the data by removing all rows without a *crane start date* (the moment the barge comes under a crane to start unloading and/or loading) and by removing all rows without an *end date* (the moment the barge leaves the terminal).

For this data set, the planned dates reach from 1 October 2017 to 22 November 2017. The waiting time is calculated by subtracting the start date (the time a barge arrives) from the crane start date. The handling time is calculated by subtracting the crane start date from the end date. The difference between actual time and planned time is calculated by subtracting the planned time from the start date.

If enough voyages to make a sufficiently confident analysis exist, an analysis of that terminal is made. We consider ten voyages as minimum in this analysis. For our analysis we create a pivot table containing per terminal the voyages with average waiting time, average handling time and average difference between the actual time and the planned time. The average is taken because only insignificant differences, probably due to the registration speed of the system, exist.

The analysis for each terminal can be found in Appendix G (confidential). This Appendix also provides an explanation on how to read box-and-whisker plots as we will discuss below. For readers unfamiliar with reading box-and-whisker plots, we suggest reading Appendix G before continuing this section.

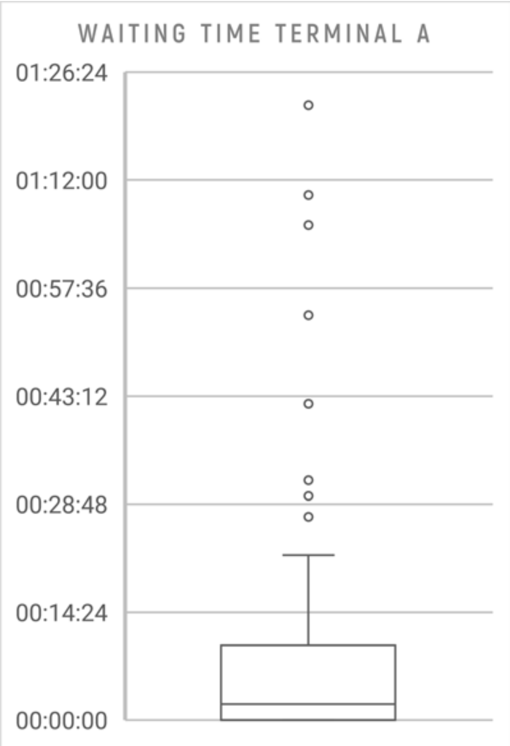


Figure 7a Box-and-whisker plot waiting time at BCW

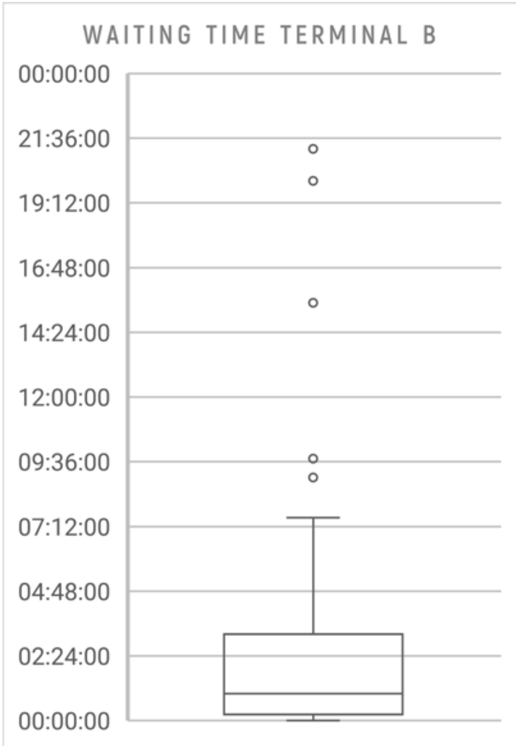


Figure 7b Box-and-whisker plot waiting time at DDE

Our terminal analysis shows large differences in waiting times at terminals. To compare two extreme cases: the waiting time is for 75% of the journeys at Terminal A under 10:00 minutes, while at Terminal B 75% of the waiting time lays under 3 hours and 12 minutes. The median at Terminal A is 2:08 minutes, while the median at Terminal B lays much higher at 1 hour. The box plots not only show us that there is big difference in median between terminals, but also in spread.

Another way to look at the waiting and handling times is by looking at the averages. For every terminal the average waiting time and the average handling time are plotted in Figure 8.

We see that a large difference in average waiting times exists. For example, Terminal A offers an average waiting time of 12 minutes, but Terminal B has a waiting time of over 2 hours and

33 minutes. We also see large differences in handling time. Terminal C takes 2 hours and 10 minutes per voyage, while Terminal D has an average of 7 minutes.

However, this discrepancy in handling times can be explained by the number of containers loaded and unloaded. We only load and unload 22 containers on average at Terminal C, while we load and unload almost 70 containers at Terminal D. The handling time, therefore, seems to be largely dependent upon the number of containers.

Censored

[redacted]

■ Waiting Time ■ Handling Time

Figure 8 Waiting and handling times of the terminals

To further investigate this phenomenon, we define a new performance indicator: the average handling time per container. The average handling time per container is defined as follows:

$$\frac{average\ handling\ time\ per\ voyage\ stop\ at\ terminal}{actual\ number\ of\ containers\ unloaded + actual\ number\ of\ containers\ loaded}$$

Figure 9 on the next page shows the handling time per terminal as dark grey bars. The left y-axis depicts the handling time. The handling time per container is represented as a light grey line. The corresponding times can be found on the right y-axis. We see no correlation between the total handling time and the handling time per container. People might expect that terminals with a faster handling time also have a faster handling time per container. However, we did not find a clear relation in Figure 9. The chart does suggest, however, that the total handling time is largely dependent on the number of containers and the handling time per container.

[redacted]

Figure 9 Bar and line graph of handling time and handling time per container

To further investigate this, we predict the handling time (dependent variable) with the number of containers (independent variable) in the statistical analysis software SPSS using linear regression. Appendix H explicates all details on this linear regression analysis.

We can, as already suggested earlier, based on this regression analysis state that the chance of no correlation is 0%. In other words, a correlation exists. We found that the handling time can be forecasted with 2 minutes and 26 seconds per container and a constant initialization time of 15 minutes and 48 seconds.

However, the data points in the scatter plot shown in Figure 10 on the next page are spread broadly from the trend line, making that predictions still contain large level of uncertainty. Also, the upper and lower bound of the 95% confidence interval for both the constant as the variable are far from the estimated values. In our analysis in Appendix H we found that 64% of the handling time can be predicted by the number of containers loaded and unloaded. To make more reliable predictions, import and export containers could be analysed individually. A model per terminal instead of one for all terminals could improve the precision too.

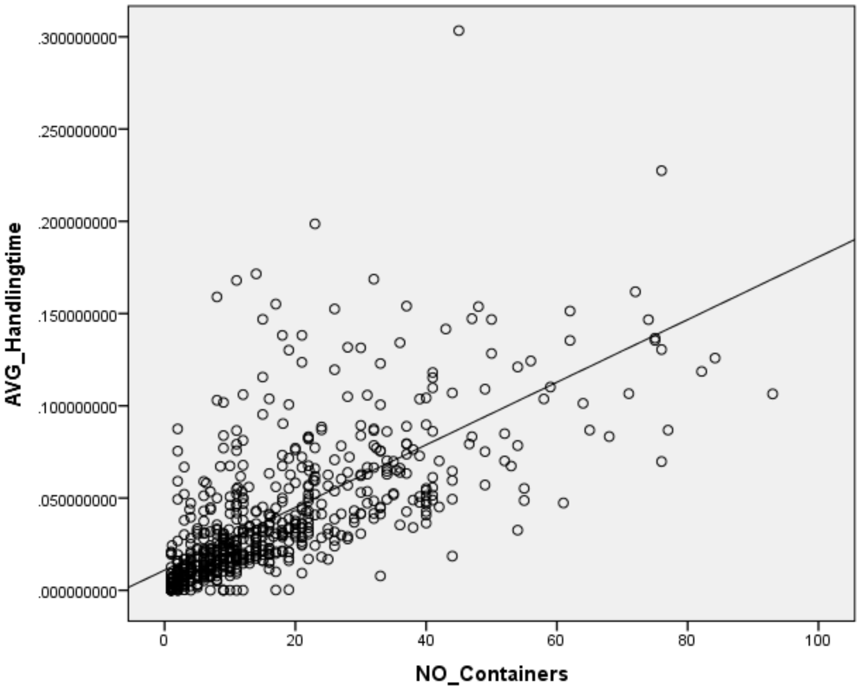


Figure 10 correlation plot of the average handling time (in days) and the number of containers with trend line at y = 0.0016969x + .010983623

It seems that the more containers loaded and unloaded, the larger the difference between the handling time and the expected handling time. These are represented respectively by dots and the line in Figure 10.

The bar graph in Figure 11 shows the average difference between the handling time and expected handling time in days on the y-axis. The difference is calculated by taking the distance of each point to the line for every sample. Then we cluster these distances in groups and calculate the average. We see that for groups containing less than 40 containers, this average difference is lower than 0.025 days, and for 1-10 containers even 0.012 days. For groups with more than 41 containers, this difference seems to be higher. Therefore, the model is more usable with less than 40 containers than with higher numbers.

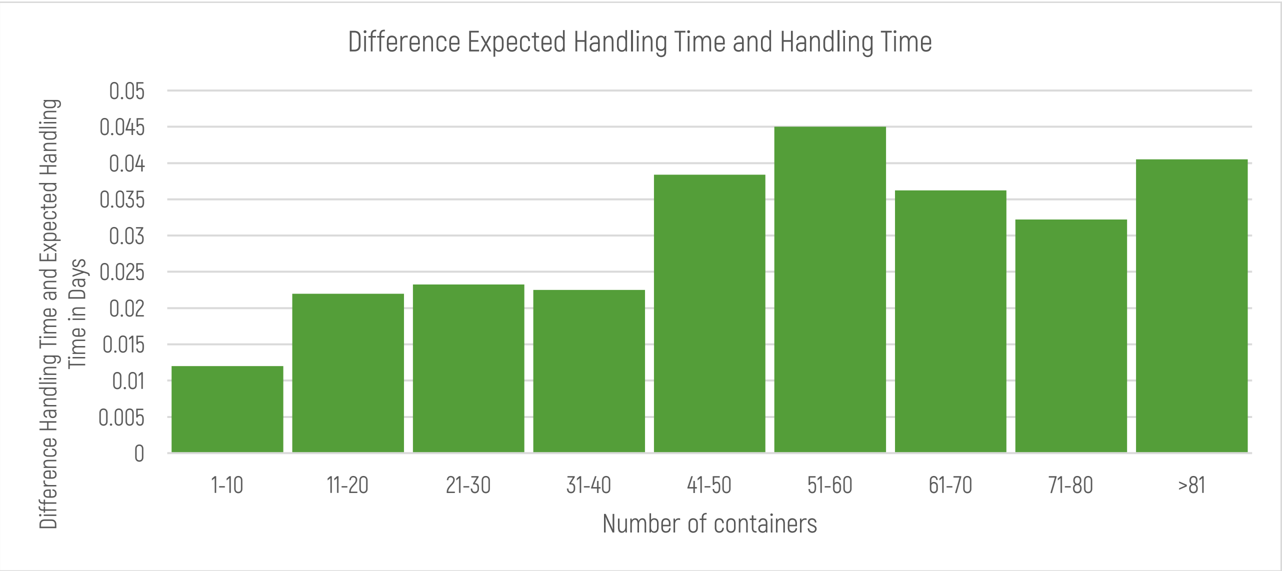


Figure 11 Bar graph difference expected handling time and handling time

Terminal	Waiting Time	Handling Time	Handling Time Per Container	Planned vs Actual

Censored

[redacted]

Table 5 Terminals with Average Waiting Time, Average Handling Time and Difference between Planned and Actual arrival in days

Table 5 shows the average waiting time, average handling time, average handling time per container and the average difference between the time a barge was planned to arrive and the actual arrival in days. The difference between planned and actual time is negative if the arrival date is earlier than the planned date. The difference is positive if the arrival date is later than the planned date.

As we noticed earlier, high differences in average waiting and handling times at terminals exist, but not so much in handling times per container. We would expect that the barges are, on average, exactly on time in a perfect schedule. However, the average over terminals shows that barges arrive 0.22 days (5 hours and 17 minutes) earlier at the terminal than they are planned to. A possible explanation is that the planned moment is not clearly defined as whether it is the moment a barge should leave the terminal or whether, as we interpreted it, it is the moment a barge enters the terminal. Another possible explanation could be that planners like to have some extra time in their planning. The average of the difference in planned and actual times is largely influenced by Euromax. Euromax has a value of -2.10, which has some voyages with a difference in planned and actual arrival of over 4 days. This might be a registration error or a barge staying for multiple days.

3.2 BARGE SCHEDULING

Next to data analysis, it is important to look at the current scheduling method as it is performed currently by human planners. To improve the chances of acceptation, the algorithm should meet the wishes of the planners. Currently, barges sail at a fixed sailing schedule between an inland terminal and the Port of Rotterdam, comparable to a bus line. As we recall from Section 1.1, MCS ships from and towards five inland terminals. Each inland terminal has its own sailing schedule between the inland terminal and the Port of Rotterdam. For illustration purposes, we show one of those inland terminals (Leeuwarden) which has the following sail schedule:

Leeuwarden even weeks

	Mo	Tu	We	Th	Fr	Sa	Su
Barge	Barge A		Barge B	Barge A			
Arrival	07.00h		07.00h	10.00h			
Departure	15.00h		15.00h	17.00h			

Table 6 sail schedule Leeuwarden even weeks

Leeuwarden odd weeks

	Mo	Tu	We	Th	Fr	Sa	Su
Barge	Barge A	Barge B		Barge A	Barge B		
Arrival	07.00h	7.00h		10.00h	10.00h		
Depature	15.00h	15.00h		17.00h	18.00h		

Table 7 sail schedule Leeuwarden odd weeks

Table 6 and Table 7 show the two barges that sail between Leeuwarden and Rotterdam. In the first row the days of the week are represented. Odd weeks differ from even weeks; therefore, two tables are shown. Each barge has a time that it arrives in Leeuwarden and a time it departs from Leeuwarden.

Whereas Table 6 and Table 7 show when barges arrive at and depart from the inland terminal Leeuwarden, Table 8 and Table 9 show when these barges arrive at and depart from the Port of Rotterdam. In between the times in these tables, a barge is sailing between the Port of Rotterdam and Leeuwarden.

Table 8 and Table 9 show the arrival and departure times of the barges that are sailing between Leeuwarden and the Port of Rotterdam. Also here a different sail schedule exists for even and odd weeks. MCS keeps distinct times a planner can schedule the barge to arrive for loading and unloading in a specific area in the Port of Rotterdam. The first time in Table 8 and Table 9 is the starting moment delivery can take place at Eemshaven, Waalhaven and Botlek. The second time is the starting moment delivery could take place at Maasvlakte I and II.

For example, in even weeks barge A departs on Monday at 15.00h and can start transferring containers from 16.00h on Tuesday in Eemshaven, Waalhaven and Botlek, and from 21.00h on Tuesday in Maasvlakte I and II. At 16.00h on Wednesday Barge A leaves the Port of Rotterdam and arrives at 10.00h again in Leeuwarden

Rotterdam (From Leeuwarden) even weeks

	Mo	Tu	We	Th	Fr	Sa	Su
Barge A Arrival in Rotterdam		16.00 h / 21.00 h			18.00 h / 21.00 h		
Barge A Departure from Rotterdam			16.00 h				7.00 h
Barge B Arrival in Rotterdam				16.00 h / 21.00 h			
Barge B Departure from Rotterdam	15.00 h					12.00 h	

Table 8 sail schedule Rotterdam (From Leeuwarden) even weeks

Rotterdam (From Leeuwarden) odd weeks

	Mo	Tu	We	Th	Fr	Sa	Su
Barge A Arrival in Rotterdam		16.00 h / 21.00 h			18.00 h / 21.00 h		
Barge A Departure from Rotterdam			16.00 h				7.00 h
Barge B Arrival in Rotterdam			16.00 h / 21.00 h			20.00 h / 22.00 h	
Barge B Departure from Rotterdam	9.00 h			16.00 h			

Table 9 sail schedule Rotterdam (From Leeuwarden) odd weeks

These sail schedules for inland terminals and the Port of Rotterdam, which are provided by MCS, are used as input for the proof of concept in Chapter 5.

CURRENT SCHEDULING METHOD

To make a schedule, orders are used which can be found in the database software. Orders arrive in various ways. Possible ways are by phone, via e-mail or in person. These orders are put via a (partly automated) system in the database software. Customers can provide a fixed date or a freely adjustable date for the containers to be picked up and/or delivered.

If an order comes in more than three weeks before it needs to be shipped, it is put into the planning system three weeks in advance. MCS has a customer policy that states that customers can provide their order up to one week before the container needs to be shipped. Therefore, orders are entered into the system between one and three weeks in advance. Often information on orders is still updated as time progresses. The order needs to be final the day before pickup at 16.00h.

Scheduling takes place all day. An order is moved to human planners once it is complete. The planners then add it towards a voyage. The sequence in which orders are assigned by human planners to voyages is not specified.

THE STEPS OF SCHEDULING CONTAINERS ON BARGES

If an order is complete, and it is not too early (around 3 weeks in advance) the planners can start with scheduling the order. Orders can be planned in random order. A planner focuses on one order at a time. Planners currently take the following steps when creating a schedule:

1) The planner looks at the date fields and the date range submitted by the customer.
2) Depending on the provided date range, the planner chooses whether to transport by barge or by truck.

3) The planner makes sure that the available capacity of the vehicle at the chosen trip is not exceeded.
4) The planner optimizes the call size of the trips and number of terminals a barge visits, with the following questions in mind:
    a. Can the container be switched to another incoming or outgoing voyage within the time range provided by the customer?
    b. Can containers be shipped to another terminal?
    c. Can the customer agree on the container being delivered outside their initially desired time range?

Some of these steps, for example 4b and 4c, require the planners to call the customers. MCS wants to try to avoid this and prefers to shuffle within the provided time ranges as much as possible. The quality at which step 4 is performed, differs largely between human planners.

In practice we see that the quality of the optimization mentioned in step 4 differs between human planners. In a large part of the analysed schedules, only two or three terminals of the 25 within the Port of Rotterdam remain unvisited. In two of the five cases, a terminal was even visited twice in the same voyage. Planners also sometimes plan terminals with the same point of time, which is physically impossible to accomplish.

DECISION POINT

For the algorithm it is important to know at what point(s) in the process a decision is made. In this process one main decision is taken: a container *is* or *is not* placed on a voyage. For example, a container can be placed on voyage j and on voyage $j+1$. It is up to the planner to decide on which voyage the container should go. From this follows the routing. The routing depends on which containers are chosen, since this leads to a terminal being visited or not.

The main point of decision in barge scheduling can also be formulated as the following question: *On which trip do I take the container, where no trip is also an option?* If no trip is chosen, the container will be transported by truck. This decision is made by the human planner in the scheduling process when orders are added to voyages.

The manager barge and truck planning at MCS sees a good decision as a decision that preserves the barge sail schedule as much as possible, visits the least number of terminals in Rotterdam and has the biggest possible call size. A bad decision is a decision that exceeds the date ranges provided by the customers. To improve the chances of acceptation of the algorithm, we incorporate these decisions as much as possible. However, sometimes it could be better to change the sail schedule if this yields lower costs, or chose to transport a container later if the fine and reputation loss do not outweigh the extra transportation costs.

3.3 THE ALGORITHM

REQUIREMENTS

To be able to automate (parts of) the process, an algorithm is needed. For a to be developed algorithm the following requirements are formulated:

1. **It must be built within the period for the bachelor assignment**

There is limited time available for this assignment.

2. **Time windows provided by customers must be met as much as possible**

MCS considers time windows for picking up and delivering containers set by customers as very important, since customers expect an on-time delivery.

3. **The call size must be high enough**

Barges are expensive to operate. Therefore, it is important that there is a high utilization of the barge. A large call size means that there are many containers on the barge, which means that the utilization is high.

4. **Where possible it must be adaptable to other companies and situations**

NexusZ serves many types of customers. These customers are interested in similar decision support solutions. If the algorithm is adaptable, these wishes can be fulfilled.

IMPORTANT FACTORS

Next to the requirements for the algorithm, important factors exist that the algorithm should be able to make a trade off in. The following factors are important to NexusZ and MCS:

- Preservation of barge sailing schedules
- Dates and times provided by the customers
- Opening times of terminals at Rotterdam
- Call size optimality: the number of containers that is being transported from terminal A to B (as high as possible)
- Number of terminals visited within the Port of Rotterdam (as low as possible to improve timeliness)

Using these requirements and factors, an algorithm will be developed to provide the planners at MCS more decision support on the scheduling of containers on barges.

SOLUTION OUTPUT

The algorithm should provide a schedule for several upcoming voyages. Furthermore, decision support should provide feedback on how to improve existing schedules. This indication can be given such that planners know where large benefit can be gained by contacting customers for example to ask if the time window can be changed.

3.4 CONCLUSION

In this chapter we answered the question *What should an automated scheduling algorithm be able to do?* by looking at the current planning process.

We first saw in Section 3.1 that, on average, 9 containers are unloaded and 9 containers are loaded per terminal. Large differences in loading and unloading exist. We also noted that on average every day 190 containers are unloaded, and 201 containers are loaded.

In Section 3.1 we also looked at different terminals' waiting and handling times and the difference between the planning and execution. We noted that large differences in both waiting times and in handling times between terminals exist. However, the handling time per container seems to be constant. Furthermore, we found that, on average, terminals are visited earlier than planned.

In Section 3.2 we explained that the current planning process is mainly based on the experience of the planners. They work mainly based on rules of thumb. The current planning process consists of four steps:

1) Look at date field
2) Choose feasible modality

3) Make sure capacity is not exceeded
4) Optimise schedule

Whereby the quality of step 4 depends upon the quality of the human planner and differs largely. The decision that is made, is whether a container is placed on a specific trip. The algorithm should provide a schedule for several upcoming voyages and the decision support should provide suggestions for improvement.

4 SOLUTION DESIGN

In this chapter we look at the algorithm and what the implementation of the desired algorithm should look like. Recall that our research objective is to design an algorithm and proof of concept to provide the planners at MCS better decision support. This chapter answers the question *How should the automated scheduling algorithm work?*

To do this, we build upon the work of L. Baranowski, since this is the most relevant applied work available in this field of research. However, the objective function Baranowski only minimizes the late arrival and late pick-up of containers. It contains no trade-off between other factors. The ideas and concepts behind her study are useful as inspiration. Therefore, we chose to use much of her work as an inspiration for our algorithm instead of using it as foundation.

This chapter is divided in two parts: the algorithm (Section 4.1) and the proof of concept (Section 4.2). In the section of the algorithm we describe a general applicable algorithm that can be applied to our case, but also to similar situations. In the section about the proof of concept we discuss our implementation of the algorithm for testing purposes to be able to value our algorithm. We end this chapter with a conclusion on both the algorithm and the proof of concept.

4.1 ALGORITHM

This section describes our developed algorithm to schedule containers on barges. It contains the approach that is used to create the algorithm. Then the steps for our developed algorithm are described. We end with the underlying assumptions for this algorithm to work.

4.1.1 APPROACH

The goal of the tool is to provide a series of schedules for the planner. Preferably, the tool should also show suggestions for improvement. Recall from Chapter 3 that a barge transports via a fixed sail schedule between an inland terminal and the Port of Rotterdam; a time table like a bus line. Planners need to adhere as much to those set times as possible.

The main decision concerns which containers within the Port of Rotterdam are picked up and delivered. For the Port of Rotterdam, the planners decide which terminals are visited in which sequence and how many containers are picked up and delivered. Figure 12 shows that five separate areas or groups within the Port of Rotterdam can be distinguished. L. Baranowski distinguished four groups. However, the Maasvlakte II did not exist when she defined the areas. A barge arrives at an *Arrival Day* in the Port of Rotterdam and leaves at a fixed date that we call the *Leave Day*. In between these two dates, the barge can pick-up and deliver containers at multiple terminals. Each barge has a maximum capacity for transporting containers. Every terminal is part of only one group. We describe the different areas in the Port of Rotterdam by a group with a number. Group 1 contains terminals belonging to the Waalhaven, Group 2 contains terminals belonging to the Eemhaven, Group 3 contains terminals belonging to the Botlek, Group 4 contains terminals belonging to the Maasvlakte and Group 5 contains terminals belonging to the Maasvlakte II. Table 10 provides an overview of the groups and their corresponding terminals. Figure 12 shows the location of the groups on a map of the Port of Rotterdam.

Group	Terminals
1	BCW, MRS, PROGECO3, UP7, WHT
2	ITERFOR, KRAREE, MBCROT, PCSA, PROGECO, RSTNOORD, RSTZUID, UCTEEM, UCTFRISO
3	CETEM, PCTRO, WBT
4	APM, DCS, RCT, DDE, DDN, EUROMAX
5	APM2, RWG

Table 10 Grouping of terminals Port of Rotterdam

Figure 12 grouping of terminals in Port of Rotterdam

The algorithm will focus on timeliness, since MCS appoints this as the most important factor. In the optimization step we focus on other factors which can be optimized upon, such as terminal visits or visits within the same group. However, in practice we see that the latter two seem to be of less importance to the human planners than the timeliness. In our algorithm, the sail schedules are a hard constraint and therefore always preserved. Also, the capacity should be put in as a constraint and is therefore preserved.

For every container two options exist in the algorithm: to go with a barge today or not. If this is the last planned voyage on which a container can be transported and it is not assigned to the barge, the container is transported by truck. If this is not the last voyage a container can be shipped, it will remain in the list to be shipped with one of the next voyages.

Since new containers can come in as time progresses, it is important to schedule as many containers as early in the voyage schedules as possible. We do this by filling the barge as much towards the capacity constraint as possible. We initially do not pay too much attention to the number of terminals visited. In the optimization step (step 4), the allocation can be optimized on reducing terminal visits too.

For our algorithm we define five steps: *(1) generating priority matrices, (2) filling the barge, (3) repeat step one and two for the next voyage, (4) optimize the schedule* and *(5) sequencing*. These five steps are further explicated below.

STEP 1 – GENERATING PRIORITY MATRICES

To be able to quickly generate a schedule, we want to transpose our lists with all container data and our barge sail schedule into a matrix. We want to create a matrix like the example in Table 11. This matrix contains for each terminal within the Port of Rotterdam the number of import or export containers, split up into priorities.

	BCW	PCTRO	WBT	APM	DCS	RCT	EUROMAX	...
priority 1	15	0	0	3	0	3	9	...
priority 2	0	0	0	1	0	4	26	...
priority 3	1	0	2	4	0	9	5	...
priority 4	13	0	1	0	0	2	7	...

Table 11 example of an unplanned import or export matrix

One of the most important aspects in the algorithm is the determination of priorities for containers. Since MCS states that it is important that containers are delivered and picked up at the date the customer wishes and that barges keep their schedule, we choose to assign priorities based on delivery and pick-up dates and times.

The idea of generating import and export matrices can be found in the work of L. Baranowski (2013). L. Baranowski uses a priority system based on *Arrival Days* and *Leave Days* of barges, and days a container can be picked up or delivered. This system works best when used in a situation where one barge per day arrives.

Since it is more generically applicable if we define priorities based on if it is possible to assign a container to a voyage we use another definition of priorities than Baranowski does. We define the following priorities:

Priority 1 if the container only fits on voyage 1 and not on another voyage.
Priority 2 if the container fits on voyage 1 and voyage 2, and no other voyages.
Priority 3 if the container fits on voyage 1, 2 and 3, and no other voyages.
Priority 4 if the container fits on voyage 1, 2, 3 and 4, and no other voyages
…

Whereby 1 is the first upcoming voyage, 2 is the next one, etcetera. The lower the number, the more important that priority is. So priority 1 containers are most important to transport.

More generically speaking, the priority system with priority n can be described for voyage j as:

Priority n if the container fits on voyage j, … , j + (n-1) and no other voyages.

Two matrices are generated for voyage j: an unplanned import matrix containing all containers to import and an unplanned export matrix containing all containers to export.

STEP 2 – FILLING THE BARGE

In step 2 we assign the containers from the matrix to the voyages. Since we want to minimize the number of containers transported by truck instead of barge, we assign in order of priority. The containers with the highest priority can become the largest problem. For example, if extra orders between voyage 1 and voyage 2 come in, we can always postpone containers with priority 3, but we cannot postpone containers with priority 2.

Therefore, we first assign containers to the barge with priority 1 containers as much as possible. If not all priority 1 containers can be placed, the remaining containers need to be transported by truck. We allocate containers to the voyage from priority 1 first, then of priority 2, then of priority 3 etcetera.

We want to minimize terminal visits by barge. This makes us want trucks to go to more terminals. We accomplish this by leaving terminals with small numbers of containers and assigning terminals with large numbers of containers to barge voyages. Therefore, we assign containers to the barge in order of terminal with most containers to terminal with least containers when allocating within each priority.

We mainly take the containers in the order of priority, because priority 1 containers become our first problem. If those are not taken, those need to be transported by truck. Priority 2 containers become a problem if those are not taken on the next voyage, so those can become a problem very fast. Although a saving could be made by combining terminals that are already being visited if containers with other priorities are chosen, we see in NLink that in practice not many terminals are skipped. Furthermore, these savings can be obtained in step 4.

All containers that are on the list that need to be planned and are planned now are removed from the unplanned list. All containers that are priority 1 and not transported by a barge are moved to the truck list.

STEP 3 – REPEAT STEP ONE AND TWO FOR THE NEXT VOYAGE

In step 3 we restart the algorithm for the next voyage. This also means that priorities will shift. So, priority 2 becomes priority 1, priority 3 becomes priority 2 etc. For the last priority, the number of containers needs to be pulled from the input list again. See Figure 13 for an example of a shift.

	BCW	MRS	PROGECO3	UP7	...
PRIORITY 1	0	0	0	0	...
PRIORITY 2	7	6	0	3	...
PRIORITY 3	2	1	2	4	...
PRIORITY 4	8	1	0	10	....

	BCW	MRS	PROGECO3	UP7	...
PRIORITY 1	7	6	0	3	...
PRIORITY 2	2	1	2	4	...
PRIORITY 3	8	1	0	10	...
PRIORITY 4	?	?	?	?	....

Figure 13 example shift of priorities

However, the new matrix does possibly not contain all containers if only is shifted. Containers that are available later than the current voyage can be assigned to a next voyage.

To illustrate this, we created a case depicted in Figure 14. In this example we depicted the pickup availability of containers A, B, and C as blue bars. All containers are available at the same terminal, which we call for convenience terminal X. The moments barge 1, 2 and 3 are available in the Port of Rotterdam to pick up containers are depicted as green bars. Container A can only be transported by barge 1. Container B can be transported by barge 1 and 2. Container C can only be transported by barge 2.

This means that barge 1 has one priority 1 container (container A) and one priority 2 container (container B) at terminal X. If Barge 1 does not take container B, we see that barge 2 has two priority 1 containers (B has become a priority 1 container, and container C is added) and 0 priority 2 containers. Opposed to the result of when only shifting would have been applied. Since container C is than not considered, there would only be one priority 1 container in the unplanned matrix. In this example, not only containers are added to the last priority for the next barge voyage, but also to higher priorities.

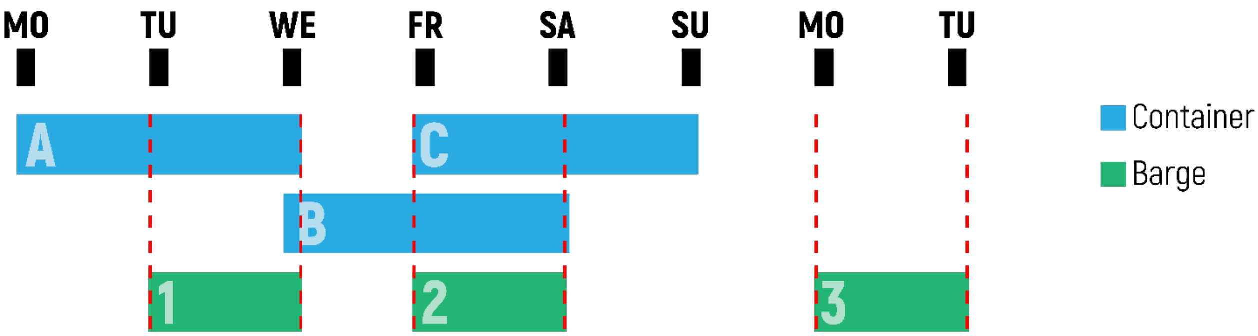


Figure 14 Example: prioritization for terminal X depicted in a timeline

It is preferable to look at all input data, updated with the removal of already planned containers. In this way we make sure to also incorporate containers that are not eligible for previous voyages in the current schedule. This method can cost a little more computation time than shifting, but we do not see computation time as a bottle neck. Therefore, at step 3, we not only shift, but fully reperform step 1 and 2 with updated input data.

STEP 4 – OPTIMIZE THE SCHEDULES

To maximize revenue the schedule should be further optimized. In this step we optimize on a trade-off between the number of containers transported by barge and the number of terminals visited by barge using mutations. Costs can be defined for visiting a terminal with a barge or truck to express this trade-off. Costs can be used to express several factors such as distance, weight, fixed costs for visiting a terminal and many others. Costs can also reflect the decision of choosing to visit terminals within the same group or not. Optimization takes place over all planned voyages after the allocation for a certain number of voyages in step 1 to 3 has been performed.

Not all orders of all voyages are already known, but an estimation or stochastic process can be used to simulate unknown orders. For example, if we expect voyage 3 to contain 20 more priority 1 containers than already known in the system, we can add them before optimizing to simulate real orders that come in later.

Optimizing can be done in several ways. We can use move or swap operations to reach an improvement of the schedules. These operations can be performed a certain number of times, or via a special technique such as simulated annealing or tabu search. This step can be added to improve the generated schedules. However, for a feasible schedule it is not necessary to execute an optimization.

STEP 5 – SEQUENCING

As a result of step one to four, matrices with import and matrices with export containers are generated for a number of voyages. From these matrices we know which terminals are visited and how many containers are picked up and delivered. However, we do not know yet in which sequence the terminals are visited. In step 5, our last step, we determine the sequence of terminal visits.

In practice, barges visit terminals in the order of the groups we distinguished in Section 4.1.1. We derived this from the data available in NLink. So, a barge starts at group 1 (Waalhaven) then sails to group 2 (Eemhaven), to 3 (Botlek), 4 (Maasvlakte I), 5 (Maasvlakte II) and then back to 4, 3, 2, and 1. This principle is comparable to a line bus that drives from bus stop A to E and stops at bus stops B, C and D.

Figure 15 shows a map with the route a barge takes if all terminals are visited in the Port of Rotterdam. On the right of the map the inland terminal is located, on the left group 5 (Maasvlakte II). As discussed earlier, if the line is followed starting at the right, the areas are visited in the order 1, 2, 3, 4, 5, 4, 3, 2, 1.

Figure 15 The simplest route a barge can take while visiting all terminals

However, a terminal only needs to be visited once in a voyage. Terminals can be visited on the way up to Maasvlakte II or on the way back to the inland terminal. Figure 16 shows a schematic overview of when a barge visits a terminal. A barge stops on the way up at a terminal if the number of containers to unload is higher than the number of containers to load. In this way a barge always ends with fewer containers than that it started with and capacity constraints can therefore never be exceeded. A barge stops at a terminal on its way back if the number of containers to unload is less than or equal to the number of containers to load.

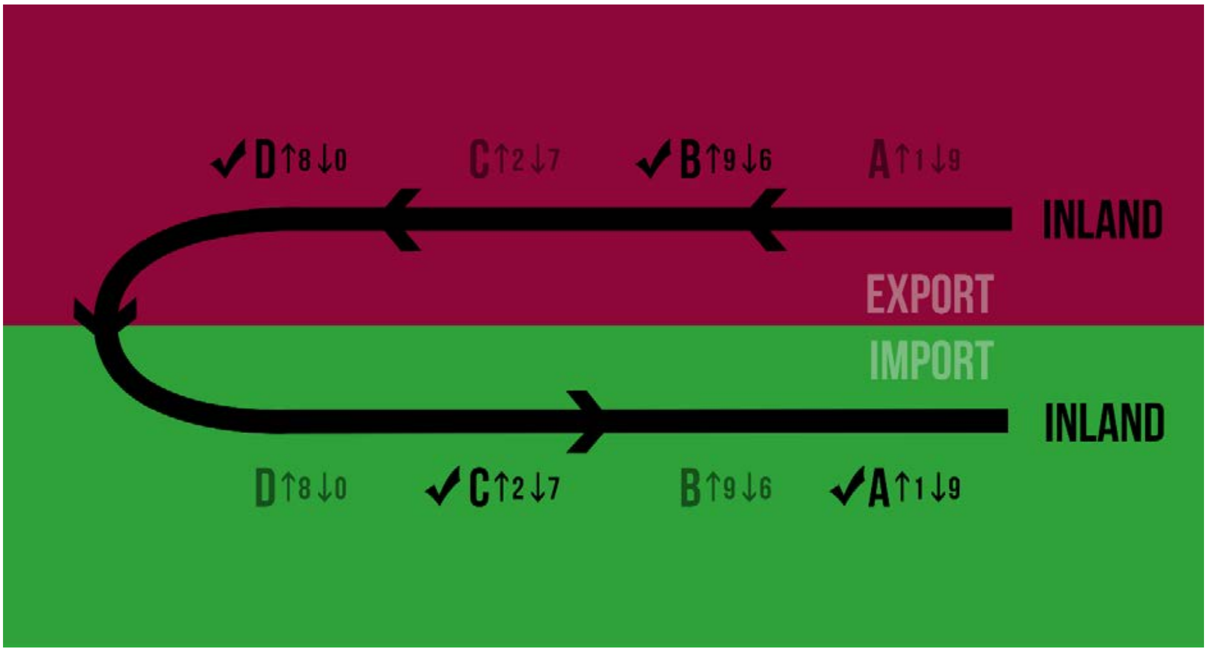


Figure 16 A barge goes to a terminal on the route from inland to Maasvlakte II if Unload > Load. A barge goes to a terminal on the route from Maasvlakte II to the inland if Load >= Unload.

With these rules we determine if a terminal is visited on the way from the hinterland to Maasvlakte II or on the way from Maasvlakte II to the hinterland. Using these rules we generate a list with the terminals in sequence and the number of containers loaded and unloaded.

In reality, the routing depends on many factors. Some of these are whether a terminal is open or closed, if high waiting times are expected and other factors. This is hard to implement in the algorithm since we do not have access to this live data. However, the proposed method for step 5 already provides quite adequate results which we discuss in Section 5.4.

4.1.2 ASSUMPTIONS ALGORITHM

To model reality, some simplifications and assumptions are made. The algorithm is developed under the following assumptions:

- All *priority 1* containers not transported by barge are transported by truck. All *other priority (2, 3, …, n)* containers can be transported by the next voyage.
- Barges have a fixed capacity.
- A certain time window exists wherein a container can be picked up and delivered without having to deal with penalties.
- Terminals, locks and bridges have a 24/7 service.
- All containers in the input data are going from or towards the inland terminal, there are no containers moved within the Port of Rotterdam.

4.2 PROOF OF CONCEPT

To analyse the performance of our developed algorithm, we implemented step 1, 2 and 5 in Excel with Visual Basics for Applications (VBA). In this section we explain per step how this is done. Furthermore, we provide a list with all assumptions added to the algorithm in our proof of concept to make it programmable within a feasible time.

ADDITIONAL ASSUMPTIONS PROOF OF CONCEPT

The assumptions made for our proof of concept include the assumptions of the algorithm which can be found in Section 4.1.2. The following *extra* assumptions are added:

- The proof of concept only covers containers that are shipped to or from Meppel.
- The fixed capacity is 80 containers.
- A container can be delivered up to two days before its delivery date at the destination terminal without penalties.
- A container can be picked-up up to two days after its pick-up date at the pick-up terminal without penalties.
- The barge always leaves the next day. In the sailing schedule of Meppel this is true in reality for 3 out of 4 voyages and we think that for the barge that waits an extra day that this is a rest day.

INPUT

For the input of the model we use real historic data. Every hour, two sheets are generated. One sheet contains all the calls with containers that are not yet planned. The other sheet contains the containers that are planned. For our proof of concept, we decided to work with the sheet containing the containers that are planned, since the information in the unplanned sheet is not complete enough to make a schedule. This is because not all information is provided by the customers, and as soon as it is, it almost immediately gets scheduled.

For our prototype, we do not use the data regarding the human made schedule but only use the customer input to make our own schedule. Further explanation on the experimental set-up can be found in Section 5.1.

4.2.1 STEPS

Recall from Section 4.1 that our algorithm consists of five steps. For our proof of concept, we also need to clean up the Excel data sheet for missing or incorrect data. We explain this under step 0. For steps 3 and 4, which we do not perform, we explain why they are not performed in our proof of concept.

An algorithm is a series of sequential steps. Recall that the algorithm performs five steps. With step 0 added, our proof of concept runs the following six steps:

- Step 0: Clean up the data sheet
- Step 1: Generate priority matrices
- Step 2: Assign containers to barges based on priority
- Step 3: Repeat step one and two for the next voyage (skipped)
- Step 4: Optimize the schedule (skipped)
- Step 5: Sequencing

STEP 0: CLEANING UP THE SHEET

To make the data easier to work with, we first clean up the retrieved Excel sheet. MCS operates with five main inland terminals. The generation of schedules happens mutually independent for each terminal. Therefore, we decide to look at one inland terminal for the proof of concept. We consider transportation between Meppel and the Port of Rotterdam. Meppel is the smallest terminal in the data set and has a barge sail schedule that is the same for odd weeks and even weeks. Other inland terminals have sail schedules differing for odd and even weeks. This makes Meppel easier and faster, and therefore more suitable, to test within the given timespan.

The generated sheets contain unnecessary and incomplete data. First, this step duplicates the original worksheet and renames it to *ModelInput*, so we can always restore the original sheet. Then, all containers that do not have a MU or MI (Meppel Export or Meppel Import) code are removed, so that only containers that are from or towards Meppel remain. Next, we convert the Epoch times to Excel times to improve compatibility with Excel's functionalities. More information on Epoch times can be found in Appendix I.

STEP 1: GENERATE PRIORITY MATRICES

In our algorithm description we described that we want to create a matrix with the number of containers per priority for each terminal. We do this for import containers and export containers separately. The priorities are defined as follows:

Priority 1 if the container only fits on voyage 1 and not on another voyage.
Priority 2 if the container fits on voyage 1 and voyage 2, and no other voyages.
Priority 3 if the container fits on voyage 1, 2 and 3, and no other voyages.
Priority 4 if the container fits on voyage 1, 2, 3 and 4, and no other voyages.
…

We define the containers to have a time window *s* of two days. This time window means that if an export container has an *onTerminal* time of day *d*, that we can start delivering it at day *d-2*. It also means that if have an import container has an onTerminal time of day *d*, that we can start delivering it at day *d+2*. Since A barge stays for 2 days in the Port of Rotterdam and barges arrive at least one day apart, we define priority 1, 2, 3 and 4, and no further priorities in our proof of concept.

The proof of concept first looks at export containers. It creates an empty unplanned sheet with all headers initialised for both import and export containers. Then the proof of concept loops over the terminals and if the name of the terminal on the unplanned sheet corresponds to the ModelInput sheet and the corresponding container is an export container, the container gets assigned to priority 1 if the ArrivalDate is earlier than the onTerminal time and if the onTerminal time s is smaller than the Leave date for voyage j. If this also holds for the next voyage, priority 2 is assigned, etc.

Figure 17 shows a graphical representation of when a barge takes a container. The timespan of an available container is represented in blue. Barges are represented in green, where A is the Arrival date and L is the LeaveDate. Barges that are eligible to take the containers are red outlined.

For import containers happens the same, only the on terminal time noted here is equal to the time a container can be picked up. We can pick it up in the time window from onTerminal to $onTerminal + s$. This changes our pseudo code to:

```
If ArrivalDate < onTerminal + s And on Terminal < LeaveDate Then
       a container is eligible to go on a barge
End If
```

The feasibility of loading an import container is graphically represented in Figure 18, in the same way as Figure 17 does.

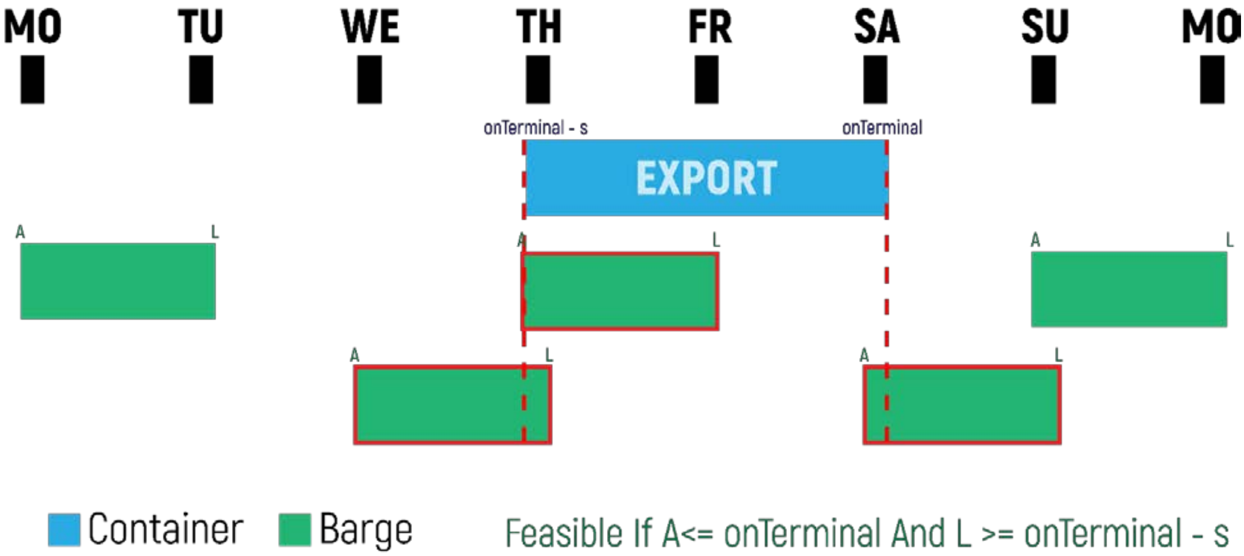


Figure 17 Example of when export containers can be taken onto a barge. In this example, the priority would be 3.

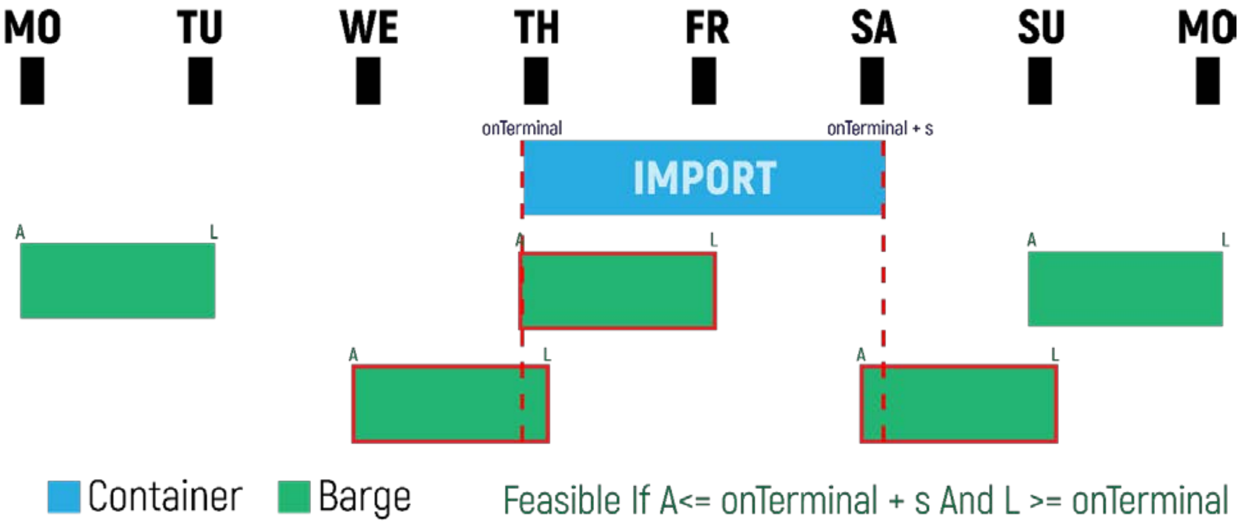


Figure 18 Example of when import containers can be taken onto a barge. In this example, the priority would be 3.

STEP 2: ASSIGN CONTAINERS TO BARGES BASED ON PRIORITY

First, matrices for the planned sheets are initialized. If the name of the terminal is the same in the unplanned sheet as the terminal in the planned sheet we are planning for, and this terminal contains the highest value in that row (most containers), it is added to the barge and removed from the unplanned sheet. This happens per priority, first for terminals that contain planned containers and then for all terminals and first for all export containers, and then all import containers within that priority. The pseudo code looks as follows:

```
'PSEUDO CODE FOR STEP 2

For priority 1 to 4

'EXPORT CONTAINERS THAT ARE ALREADY PLANNED IN
        For terminal 1 to 25
        If terminalsVisited(terminal) = true Then

                Maximum = 0 'initialize maximum
                For n 1 to 25
                        If value at that priority at terminal n > Maximum Then
                                Maximum = value
                        End If
                Next n

                For m 1 to 25
                        If terminal value at that priority = Maximum Then
                        If the capacity constrained is not exceeded when adding this container
Then
                                Add container to planned sheet and remove from unplanned sheet
                        End If
                        End If
        Next m
        End If

'ALL EXPORT CONTAINERS
        For terminal 1 to 25
        Maximum = arbitrary large number
        Do While Maximum > 0
        Maximum = maximum for row of priority we are looking at
        For m 1 to 25
                If terminal value at that priority = Maximum then
                If the capacity constraint is not exceeded when adding this container Then
                        Add container to planned sheet
                        Set TerminalsVisited(m) = 1
                End If
                Remove value from unplanned sheet, even if capacity constraint is exceeded when
                adding this container
                End If
        Next m
        Loop While

'IMPORT
Same as Export. The terminalsVisited List is shared, the capacity is separated in variables.


Next priority
```

Figure 19 Pseudo code for step 2

This pseudo code can be used to see the structure of a potential implementation.

STEP 3: REPEAT STEP ONE AND TWO FOR THE NEXT VOYAGE (SKIPPED)

We only perform the scheduling for one voyage. This is because the data for the next voyage can be retrieved by taking the Excel sheet for the next voyage. Furthermore, a schedule can be changed even when a barge sails, therefore, scheduling for multiple days would be nice for humans to see and can be functional if we try to optimize the schedule as a whole, but since we skip step 4 as well, it is not of added value to implement step 3 into our proof of concept.

STEP 4: OPTIMIZE THE SCHEDULE (SKIPPED)

In this step we would optimize the scheme by using mutations. However, for the evaluation of the core of this algorithm it is not necessary to evaluate this step and due to the limited time, we decided not to test this step in our proof of concept.

STEP 5: SEQUENCING

In our proof of concept, we implement the rule of step 5 extensively explained in Section 4.1. This happens by looping forward and back over a list of terminals in sequence of route. The computer checks if the terminal is allocated to unload and/or load containers and, if this is the case, assigns the terminal visit to the way towards Maasvlakte II if the number of containers unloaded is more than the number of containers loaded, and else assigns the visit to the way backwards to the inland terminal. The list of the route can be found in Appendix J. Step 5 is executed in a separate workbook for pragmatic reasons.

OUTPUT

The output is now two matrices. One with the terminals and how many containers of each priority are loaded and one with the terminals and how many containers of each priority are unloaded.

4.3 CONCLUSION

In this chapter we developed an algorithm for scheduling containers on barges, to provide the human planners at MCS better decision support. First, we attributed priorities to containers for voyage j. Instead of the priorities being time dependent, we made them dependent on the number of voyages it can be placed on. Making this algorithm more broadly applicable than L. Baranowski's algorithm. Then the containers are placed on barges until the capacity constraint is reached in the order of priority, terminal visited or not, and largest to smallest number of containers. When the containers are assigned to a planned matrix, the algorithm repeats step 1 and 2 for the next voyage. An optimization step takes place and finally a sequence is generated based on the number of containers to load and to unload.

In our proof of concept, we implemented these steps, except for the repetition over multiple days and the improvement step, since it is important to analyse the plain algorithm first and make sure that is performs well, before expanding it.

In the next chapter we analyse the performance of our proof of concept.

5 EVALUATION PROOF OF CONCEPT

To evaluate the performance of the algorithm developed in Chapter 4, we perform experiments with our proof of concept. In this chapter, we first explain our experimental setup in Section 5.1. In Section 5.2 we compare the outcomes in terms of visited terminals and number of containers loaded and unloaded. In Section 5.3 we compare the number of terminals visited. In Section Comparison 5.4 we look at the visiting order. In Section 5.5 we analyse how these schedules affect barge utilization. The evaluation of these aspects leads to our conclusion about the evaluation of the proof of concept in Section 5.6.

5.1 EXPERIMENTAL SETUP

Experimentation happens over the planned Excel sheets of the days of several executed voyages and is compared to the schedules found in the web interface NLink. Recall that we do not execute step 3 and 4 of the algorithm in our proof of concept due to the limited time for this research.

Our experimental setup consists of a sample of 5 dates chosen pseudo randomly, such that they are at least 2 weeks apart to minimize influence upon each other. Furthermore, the chosen arrival dates are mostly not at the same day of the week: Tuesday, Wednesday, Friday or Saturday. Every week 4 voyages take place, so a double weekday is unavoidable for our sample. For testing a planned sheet is used that is generated between 10.30 and 11.30 hour the morning of departure since this is the moment a human schedule is marked definitive before execution. The only exception is October 31st, where we work with data from 2-11-2017 due to the availability of data. The results from testing can be used to examine where the algorithm differs from what human planners do.

The chosen 5 schedule arrival dates are the following, with the corresponding voyage codes from NLink between are put between brackets:

Tuesday, 31 October 2017 (MU171030_MI171102)
Friday, 17 November 2017 (MU171116_MI171120)
Wednesday, 6 December 2017 (MU171205_MI171208)
Saturday, 23 December 2017 (MU171222_MI171228)
Tuesday, 9 January 2018 (MU180108_MI180111)

We only compare separate voyages, and no series of voyages, because if a human planner decides for voyage *j* otherwise than the algorithm does for voyage *j*, the input for voyage *j+1* will also be different. This makes it hard to compare schedules since MCS does not have clearly defined quantitative KPIs. Furthermore, new containers can be added up until the moment of departure from Meppel. It is hard to incorporate this stochastic element since we have not researched the stochastic process of entered containers nor is it known at the company.

In this chapter, we often use the terms algorithm, planned and actual when comparing outcomes. In the context of this chapter, *Algorithm* means results as generated by the algorithm programmed in (parts of) the proof of concept based on input data from the Excel sheets. With *Planned* we mean the planned visited terminals as allocated by human planners. With *Actual* we mean the voyage as it has been registered by the barge operator.

5.2 ALLOCATION COMPARISON

In this section we compare the allocation of containers of the algorithm generated schedule, the human planned schedule and the actual executed voyage. The container allocation is the output as generated after step 2 of the proof of concept. The quality of this allocation is important, since this largely determines our resulting schedule.

For each voyage in our sample set, an import and an export matrix for the algorithm, the planned schedule and the actual executed schedule is made. The matrices contain for each terminal the number of containers transported that voyage by barge. Appendix K shows these matrices. Below we evaluate the differences we found in these matrices.

On Tuesday, October 31st, the largest discrepancy can be found at the import containers. The algorithm transports only 57 containers, while the planned and actual transportation covers 88 and 86 containers, respectively. The main explanation for this difference, is that the schedule in NLink does contain containers that we did not find back in the Excel sheet. This might be due to a problem in data conversion to the supplied Excel sheets. However, for terminals that are in both our input sheet and in our planned and actual sheet we do see overlap between decisions made by the algorithm and decisions made by human planners. Furthermore, the number of terminals visited for export containers are the same for the algorithm, planned and actual schedules.

In the allocations of Friday, 17 November, again, terminals are differently chosen due to a discrepancy in input. This means that again other terminals are chosen to be visited by the algorithm than by the human planners. The total number of containers exported is for the algorithm 80, which is exactly the capacity constraint we chose. The total number of containers exported in the actual and planned schedule is 103, which is much higher than our chosen capacity constraint.

For Wednesday 6 December and Saturday 23 December, we encounter the same points as for 31 October and 17 November. For 9 January, the planned voyage and algorithm match quite closely, however we notice a difference due to difference in input.

In conclusion, we see large discrepancies in the decision for which terminals are visited between human planners and the proof of concept. The main cause seems to be that the input is different for the human planners than for the proof of concept. It appeared to us that there is a problem in the data conversion from the main database to the Excel files. In Excel, around half of the containers show up for the same voyage code as in NLink. However, the exact problem with this data conversion could not be found.

Also, the number of containers transported is higher in every actual voyage than our capacity constraint. While our algorithm transports at most 80 containers, the average of import and export containers is in the actual voyages while the average of import and export containers in real life lays at 94.1 for our sample.

5.3 NUMBER OF TERMINALS VISITED

One of the performance indicators is the number of terminals visited. MCS wants to minimize the number of terminals visited, to decrease the chance a voyage takes longer than expected. Therefore, we measure how many terminals are visited in the schedule in the Port of Rotterdam.

Figure 20 shows a clustered bar graph containing, for each voyage from the sample set and the average of these voyages, the number of terminals visited in the algorithm, planned schedule and actual voyage. If a terminal is combined to pick up and deliver containers, we only count it as one visit. If a visit was not combined by the human planners in their schedule, or if it is executed twice, then we also count the visit as double.

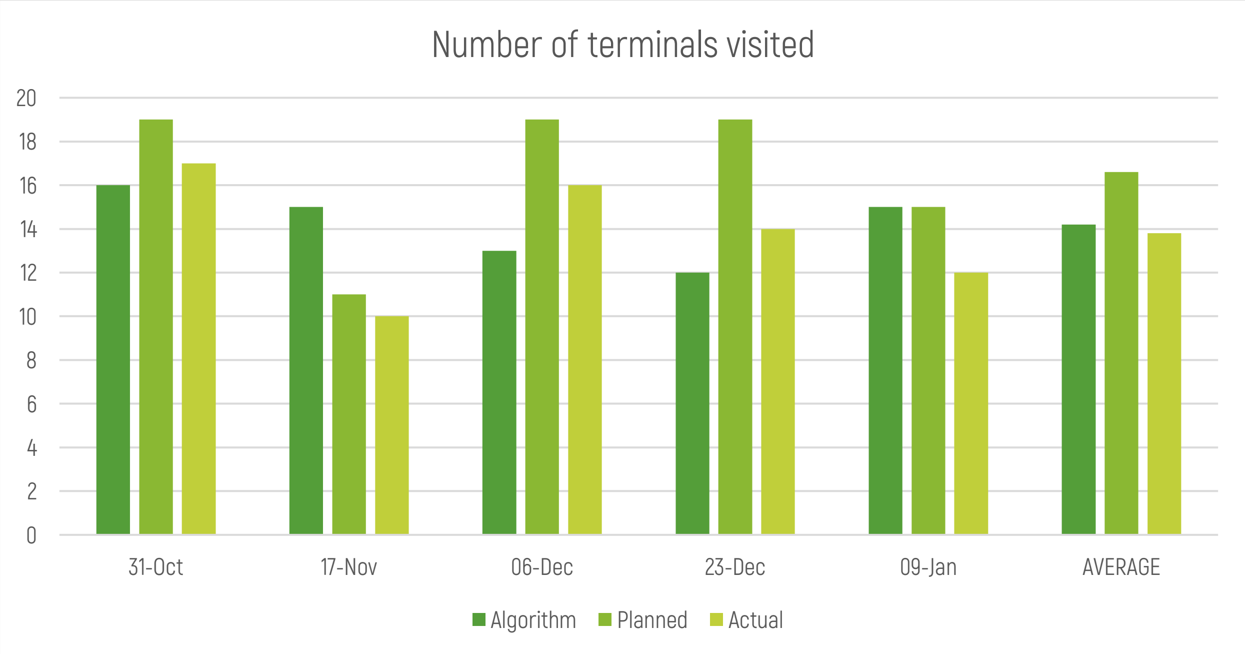


Figure 20 Number of terminals visited for each voyage in the sample set for algorithm, planned and actual voyage

We see that, in every voyage from our sample set, the human planners have scheduled more terminals to visit than the barge actually visits. In the database we see comments such as *'X containers from terminal Y loaded via terminal M'*, meaning that sometimes containers can be loaded or unloaded at another terminal than originally planned.

On average, the algorithm matches the number of actually visited terminals more closely than the human made schedule does. However, we concluded in Section 5.2 that the input data from Excel differs with the data in NLink. This makes it hard to say how the algorithm would perform on the number on terminals visited with the same input as the human planners.

5.4 COMPARISON SEQUENCES

For each voyage we also perform step 5 of our algorithm: the sequencing of the terminal visits. The schedules of the algorithm compared to the planned and the actual schedule can be found in Appendix L. The sequences are given in the form of a table that can be read from up to down, to see the order in which the terminals are being visited. In the second and third column, the number of containers imported and exported can be found.

Since many discrepancies in terminals visited by the algorithm and terminals visited in the planned and actual voyages exist, it is hard to compare these schedules. To make it easier to compare step 5 from the algorithm solely, we also run step 5 on the data from NLink. In NLink we can find the schedules as they are made by the human planners. We converted the data manually into a table similar to the output of step 2 from the algorithm. In that way, we lose the sequencing done by human planners and are able to compare the sequencing step from the algorithm with the sequencing done by human planners. We refer to the schedules generated with the logic from step 5 of the algorithm as *generated*, and to human schedules of voyages as *NLink*. These comparisons can be found in Appendix M. The following three valuable insights are gained from all five comparisons:

THE DEAD-END PRINCIPLE

While step 5 of the algorithm considers the route as a whole and draws within an inlet one clockwise line to determine the order of terminal visits, the human planners seem to consider an inlet to be more like a dead-end road. Where they, when exporting containers, first take the nearest terminal, and then the farthest. On the way back to the inland terminal, when there are more containers to import than to export, human planners first go to the furthest terminal and then to the nearest terminal.

Figure 21 shows how the proof of concept currently has implemented the order of terminal visits within inlets: as a clockwise loop visiting first terminals on the right side of the inlet, then at the bottom, and then at the left. Figure 22 shows how planners see an inlet: A barge that differs from the main route and enters a dead-end road. For example, for the voyage of 9 January, WHT and UP7 are switched in order. For both terminals holds that more containers are imported than exported, so the terminals are visited when sailing from Maasvlakte II to the hinterland. In the computer-generated sequence, UP7 is visited first and then WHT. In the human generated schedule, WHT is visited first and then UP7. This makes that the barge can sail the red distance depicted in Figure 21 with fewer containers in the human made schedule than in the computer-generated schedule.

Figure 21 The inlet sequencing method currently used in step 5.

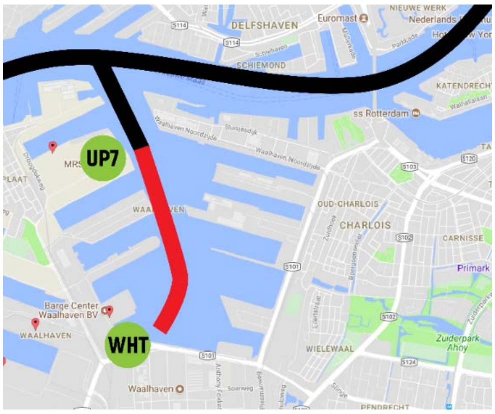


Figure 22 How Human planners see the route, more as a dead end road.

This problem can be easily fixed by changing the order of the terminals in the list used for the step 5 sequencing.

PLANNERS TAKE SMALL NUMBERS OF IMPORT CONTAINERS IF THE BARGE IS ALREADY NEARBY

If a barge is already nearby a terminal for exporting containers on its way towards Maasvlakte II, and only a small number of containers is imported (around 1 or 2), human planners try to take them then even if less containers are exported than imported. This often results in that a terminal group does not have to be visited twice within the same voyage.

In step 5 of the algorithm, this behaviour can be simulated by adding an extra rule for small number of import containers and checking if this prevents the need to visit on the way back to the inland terminal.

TERMINALS PLACED EARLIER FOR UNKNOWN REASONS

Sometimes, planners sequence one to three terminal visits earlier than the algorithm. This might be due to terminal opening times, or (expected) busyness at terminals. These reasons are unknown to us and are hard to model within the algorithm. However, this only happens at most once at each voyage from the sample set and therefore the computer-generated sequence is still useable as solution for human planners to use in a decision support.

CONCLUSION

Step 5 of the algorithm already performs closely to what the human planners do. An easy improvement step can be made by changing the order of the list to match better what human planners do in an inlet. Also adding a rule for small numbers of import containers would help step 5 closer towards the thinking pattern of human planners and make it useable in practice.

5.5 BARGE UTILIZATION

In this section we compare the utilization of barges in the sequenced schedules, as barge utilization greatly affects revenue. We generate graphs representing the utilization of the barge over time using these schedules and the numbers of loaded and unloaded containers. Appendix N contains an overview of the (planned) utilization of the barge for all voyages.

The height of the bar in the graph represents the number of containers on the barge before loading and unloading at the corresponding terminal on the x-axis. Since we assume that containers are only transported between the Port of Rotterdam and the inland terminal and not within the Port of Rotterdam, the first bar represents all export containers and the last bar from the bar graph represents all import containers of that voyage. A higher utilization at these points means that less containers need to be transported by truck.

Generally speaking, we see that the minimum numbers of containers on the barge at the human generated schedules are higher than the minima at the computer-generated schedules. Figure 23 shows the utilization at 6 December of the computer generated planned schedule and the human generated barge load. We clearly see that the minimum at the computer-generated schedule is much lower at 23 containers than the 54 containers in the human made schedule.

The difference in constantness of barge utilization is due to the fact that human planners tend to include small number of import containers earlier in the schedule than the computer does. However, the longer a barge sails with import containers, the more fuel is consumed. Since in the human schedules the utilization is more constant, the barge also lays with the same depth in the water. This could be a possible practical reason for wanting the utilization to be more constant.

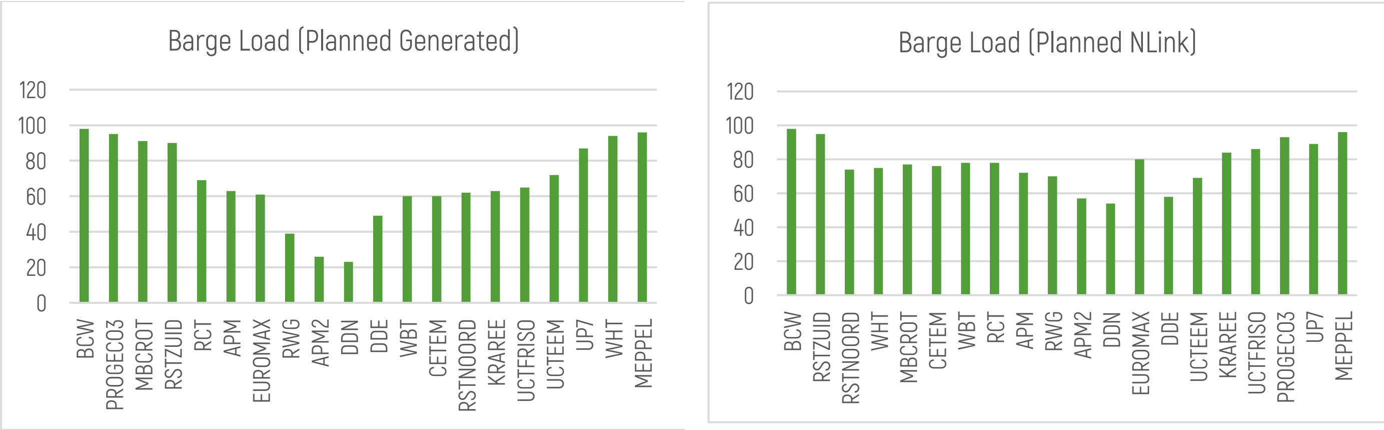


Figure 23 Barge load voyage MU171205_MI171208: step 5 of the algorithm compared to the actual barge load

5.6 CONCLUSION

In this chapter we tried to answer the question "*What performance can be expected when implementing the algorithm?*" by applying proof of concept upon five pseudo-randomly chosen voyages.

In Section 5.2 we found that there are different terminals visited by the human planners than in the human generated schedule. This is mainly due to a difference in the input data. We found out that some containers do not show up in Excel when searching on the voyage code that do show up in NLink. This makes it hard to make a decisive conclusion upon the performance of step 1 and 2 of our proof of concept.

We did see however, that human planners do transport on average more containers than the capacity constraint in the proof of concept allows. Furthermore, we saw that the number of terminals visited by human planners is higher in their planned schedules than in the execution. Generally speaking, the algorithm visits fewer terminals than the human made schedule does.

When comparing the sequences, we found that planners visit inlets in another order than assumed for the development of step 5. This can be easily implemented by changing the order of the terminals in the list of step 5. Human planners take small numbers of import containers if a barge is already nearby too. This can be implemented by adding a rule that allows for earlier pickup of small amounts. Furthermore, we saw some variations with no traceable reason. In conclusion: step 5 of the algorithm can output similar results to those of human planners if the improvements of the dead-end principle and taking small numbers of export containers earlier are implemented. In this way, step 5 be very useable in decision support.

6 CONCLUSIONS, RECOMMENDATIONS AND DISCUSSION

In this chapter we share our conclusions on our research in Section 6.1. In Section 6.2 we provide recommendations. Lastly, in Section 6.3 we discuss the algorithm and name the limitations.

6.1 CONCLUSIONS

This research answers the research question *"In what way can planners use decision support and to what performances can this lead?"* by executing the research objective: *"Design an algorithm and proof of concept to provide the planners at MCS better decision support."*

Chapter 2 answers sub question one: *"What are possible solutions for the barge scheduling problem that can be found in literature?"*. We found in a systematic literature research that insertion heuristics are applied in routing problems with time windows. We also found that popular improvement heuristics are simulated annealing and tabu search. Three former theses are also assessed. In one of these, Baranowski (2013) develops a model based on the Pipes and Filters architecture for the barge scheduling problem. She works with priorities and several filters, such as priority and distance based. Inspired by this algorithm, we developed our own algorithm in Chapter 4.

Chapter 3 answers sub question two: *"How can an automated scheduling algorithm contribute to better decision support at MCS?"*. In this chapter we researched the current situation at MCS and the wishes for decision support. We analysed the average numbers of containers at each terminal and found large discrepancies between loading and unloading for individual terminals. We also analysed the waiting times in queues and handling times at the terminals in Rotterdam, and found that large differences between terminals exist. We found that the handling time can be easily predicted if the number of containers to load and unload is known using a linear model. The current scheduling process is mainly based upon the experience of the human planners. They make a schedule based on a fixed barge sailing schedule. The decision that is made is whether a container is placed on a specific trip. The algorithm should provide schedules and possibly suggestions for improvement that help the planners to make more efficient schedules.

Chapter 4 answers sub question three: *"How should the automated scheduling algorithm work?"*. In this chapter we developed an algorithm inspired by the methods of Baranowski (2013). Our algorithm, however, does not only minimize lateness but also contains an improvement step to improve schedules where a trade-off can be made between expected lateness and number of terminal visits. Furthermore, where Baranowski defined priorities in terms of days, our priority system uses voyages. Making it easier to apply for more situations.

Our algorithm consists of the following five steps:

- Step 1: Generating priority matrices
- Step 2: Filling barge
- Step 3: Repeat step one and two for the next voyage
- Step 4: Optimize the schedules (optional)
- Step 5: Sequencing

We implemented step 1, 2 and 5 into a proof of concept which we used to test the algorithm in Chapter 5.

Chapter 5 answers sub question four: *"What performance can be expected when implementing the algorithm?"*. We found that the algorithm visits different terminals than human planners do. We also found that the algorithm visits fewer terminals than the human planners' schedule, but more than is actually executed. We also found that the sequencing step 5 comes close to the human planners, but would benefit from some improvements. These suggested improvements are repeated under 'recommendations'. We found that barge utilization is different for the algorithm and human schedules, but we cannot make a decisive conclusion about which is better. We also found a difference in the input data between Excel and NLink. We were unable to solve this conversion error, which makes it hard to determine the performance of step 1 and 2 of the proof of concept.

Now we can answer the main research question: *"In what way can planners use decision support and to what performances can this lead?"*. With the proof of concept, we proved that a schedule can be generated comparable to a schedule made by human planners. However, we did not have access to the same data that is available to the human planners, making it hard to provide a decisive conclusion on how it exactly would differ.

We cannot say with certainty whether the developed algorithm is good enough to be implemented. It should be tested in a fair test where human planners and the algorithm have the same input data. However, we did test step 5 properly. This step can be immediately used in the decision support tool, with the recommendations made implemented.

The proof of concept is only tested on a sample set of five distinct voyages between Meppel and the Port of Rotterdam. To test dependency effects, series of voyages should be tested. To test if the algorithm can also be applied to other inland terminals, such as Groningen or Kampen, the algorithm should be tested on this data too.

6.2 RECOMMENDATIONS

Based on the results of this research, we provide the following recommendations.

First, it is important to research how the data integrity can be improved. During the development of the conceptual model, we found a lot of missing data. There was a discrepancy between the data in the Excel sheets and the data in NLink. Also, we found data that was obviously incorrect. It is important for an algorithm to have valid input, as the quality of input directly affects the quality of output. To improve output quality, it is most important that time windows are registered correctly, and old data is removed from the main database. Furthermore, data should be validated before entering the database. For example, in one of the datasets a negative number is found in the field of *ContainersLoadActual*. This should not be possible. Furthermore, the database contains unnecessary and double data entries. This makes it hard to find the right entry, leading to using the wrong entry and creating larger database than needed.

A standard form can be considered. Now, customers often provide times, locations, weights, using their own forms. This makes it easier to skip or forget data. A standardized form can help improve upon the data integrity. A standardized form can also be used to specify time windows, whereas customers now provide only one time point and the margin is unknown to the planner. Customers could be rewarded for using this form by offering a discount.

An interesting path for further research in creating algorithms for barge scheduling could be machine learning. Machine learning provides computers the ability to learn without being explicitly programmed. A typical application of machine learning is when the computer uses a provided input and output to figure out what algorithm leads from that input to that output. A starting point to read more about Machine Learning is Samuel (1959).

6.3 DISCUSSION AND LIMITATIONS

As in every model, also in our research limitations exist. The first is that the proof of concept only covers containers shipped towards or from Meppel. We assume that no containers in the input data are moved within the Port of Rotterdam. This assumption is made because only a small percentage of the containers is shipped within the Port of Rotterdam. Unfortunately, it is not known how high this percentage exactly is.

Also, we assumed fixed time windows for every container in our proof of concept while in practice the time windows differ per call. In the proof of concept, we take a time window of two days to the *onTerminal* time in the database, but it is better to work with time windows that customers can provide. To be able to incorporate this, time windows should be asked from the customers and entered into the database instead of a single time.

Currently, our proof of concept only adds containers based on priority, which makes that little trade off takes place. This could lead to a barge visiting a terminal, while it could be better to use a truck. This effect can be reduced in the optimization step, but further research and testing should be done.

We also did not evaluate the run time in depth. Although the run time of the proof of concept stays within reasonable limits (5 seconds for the Excel VBA), it can become much higher when adding optimization steps and expanding the model to more voyages and terminals.

The current proof of concept has a fixed capacity of 80 containers, while in real life, there is a distinction between several types of containers such as 20ft containers and 40ft containers. By making the capacity taking care of what type of containers are transported, a much more precise estimation could be reached than in the proof of concept. Also, the real capacity constraint should be researched. In our proof of concept, we did not consider the size, transport distance and weights of containers. These aspects could be used for optimization in the optimization step.

The proof of concept does not keep track of what information is discarded. The final implementation of the decision support should provide a clear indication of orders with insufficient data to plan. Also, the proof of concept does not make a list with the containers that are going to be transported by truck, whereas the implementation of the algorithm should.

For our algorithm we assumed that all priority 1 containers not transported by barge are transported by truck, while in real life a planner might contact the customer to check whether a container can be transported on a later voyage than the date provided by the customer.

Furthermore, we assumed containers to have a certain time window, while in real life the time windows are less strictly defined. The customer provides a time, and planners must guess what the real time window is. For this algorithm to work optimally, an effort should be made to make these time windows hard time windows.

The current algorithm does only contain five steps and ends up with a sequence, while for a schedule time information is also necessary. Based on the research done in Chapter 3 on handling times and waiting times, and by knowing an average sailing speed or time between terminals, an expected arrival time could be easily generated. In this way a step 6 for adding times can be added to the algorithm. Also, we assumed terminals, locks and bridges to have a 24/7 service, while in real life this certainly is not the case. We did not research if this lead to infeasible solutions.

7 REFERENCES

Applegate David, Cook William (1991). Computational study of the job-shop scheduling problem. ORSA Journal on Computing.

Baranowski, L. (2013). Barge loading problems. Netherlands: University of Twente

Cardoen, B., Demeulemeester E. & Beliën, J. (2010). Operating room planning and scheduling: A literature review. European Journal of Operational Research

Cheng, T., & Gupta, M. (1989). Survey of scheduling research involving due date determination decisions. European Journal of Operational Research, 38(2), 156-166.

Christiansen, M., Fagerholt, K., & Ronen, D. (2004). Ship routing and scheduling: Status and perspectives. Transportation Science, 38(1), 1-18.

Cooper, D.R., & Schindler, P.S. (2014). Business Research Methods (12th ed.). New York: McGraw-Hill/Irwin

Dell'Amico M., Trubian M. (1993). Applying tabu search to the job-shop scheduling problem. Annals of Operations Research.

Genreau, M., Potvin, J. (2005). Search methodologies: Introductory tutorials in optimization and decision support techniques. New York: Springer.

Heerkens, H., Van Winden, A. (2012). Geen Probleem. Nederland: Van Winden Communicatie.

Krul, I. (2015). Decision support for container transport scheduling
A case study at Combi Terminal Twente. Netherlands: University of Twente.

Magnanti T.L., Wong R.T. (1984). Network design and transportation planning: Models and algorithms. Transportation Science.

Meij, D. (2014). Data analyse bij Combi Terminal Twente. University of Twente.

Löwik, S. https://blackboard.utwente.nl/bbcswebdav/pid-1028481-dt-content-rid-2434349_2/courses/2016-201500021-2A/Presentation%20on%20literature%20review-BB.pdf retrieved on 13/05/2017

Osman, I., & Potts, C. (1989). Simulated annealing for permutation flow-shop scheduling. Omega, 17(6), 551-557.

Potts, C., & Kovalyov, M. (2000). Scheduling with batching: A review. European Journal of Operational Research, 120(2), 228-249.

Robinson, S. (2014). Simulation: The practice of model development and use (Second edition. ed.). Houndmills, Basingstoke, Hampshire, UK: Palgrave Macmillan.

Samuel, A. L. (1995). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210-229.

Shiller, Z. (2015). Off-Line and On-Line Trajectory Planning. Switzerland: Springer International Publishing.

Solomon, M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations Research, 35(2), 254-265.

Solomon, M., & Desrosiers, J. (1988). Survey paper—Time window constrained routing and scheduling problems. Transportation Science, 22(1), 1-13.

Widmer M., Hertz A. (1989). A new heuristic method for the flow shop sequencing problem. European Journal of Operational Research

Winston, W.L., Goldberg, J.B. (2004). Operations Research: Applications and Algorithms (4th ed.). United States: Cengage learning

http://www.mcs-bv.nl/watertransport-informatie/ retrieved on 05-09-2017

https://www.epochconverter.com/ retrieved on 31-10-2017

APPENDIX A: LISTS OF BARGES AND TERMINALS

This appendix has been removed from the public version.

APPENDIX B: PROBLEM IDENTIFICATION AND PROBLEM CLUSTER

During our research we found several problems in the problem identification phase. Below these problems are mentioned in the form of a full list and a cleaned up list.

List with all existing problems:

- Planners get orders shorter in advance
- The planning is not met
- The planners get an overload of information
- It is not efficient to plan by hand
- The barges could move transport more efficiently
- Cargo information goes to JIT (Just in Time)
- More aspects in cargo & infrastructure can be considered when automated
- Transports change dynamically
- Information overload
- Not an optimal plan for transport
- Inefficient transport
    - Unneeded delays
    - Extra resources needed
    - More transport by truck than needed
    - Utilization at resouces is too low
    - Transport gets more expensive
- Competitive disadvantage
- There is no scheduling algorithm
- The planning process is not automated enough to be future proof
- There has not been time to develop and implement an algorithm

Cleaned up list:

- The planners get an overload on information
- Cargo information goes to JIT (Just in Time)
- More aspects in cargo and infrastructure can be considered when automated
- Not an optimal plan for transport
- Inefficient transport
    - Unnecessary delays
    - Extra resources needed
    - More transport by truck than needed
    - Utilization of resources too low
- Competitive disadvantage
- The planning process is not automated enough to be future proof
- There has not been time to develop and implement an algorithm.

APPENDIX C: LISTS OF KEY CONCEPTS

To be able to get key theoretical concepts on the subject of barge planning, the following search string has been filled in at Scopus:

TITLE-ABS-KEY ((barge OR ship) AND (planning OR scheduling))

We choose by hand an article with high citation impact from these search results. The article of Christiansen, M. (2004) fits this criterion with a Field-Weighted Citation Impact of 43.69.

The following terms are coined in this article:

- *Routing*
- *Scheduling*
- *Ship*
- *Fleet size*

Since these terms do not make up an extensive list, we decide to look further than only barge planning. In her Master Thesis L. Baranowski (2013) states that planning and scheduling is a topic extensively discussed in the healthcare field, where planning and scheduling are very important. Therefore, she did not limit her search to planning on the scope of barges or trucks, but extend it to the healthcare sector as well.

To get a grip on some keywords in barge planning and simulation planning, the Master Thesis *Barge Loading Problems* written for CTT, a company like MCS, by L. Baranowski (2013) is analysed. In her paper, Baranowski uses B. Cardoën et al. (2010). The following key concepts are described:

- *Scheduling*
- *Planning*
- *Solution technique*
- *Mathematical programming*
    - *Mathematical programming*
    - *Linear programming*
    - *Quadratic programming*
    - *Goal programming*
    - *Mixed integer programming*
    - *Dynamic programming*
    - *Column generation*
    - *Branch-and-price*
- *Simulation*
    - *Discrete-event*
    - *Monte-Carlo*
- *Constructive heuristic*
- *Meta-heuristic*
    - *Simulated annealing*
    - *Tabu search*
    - *Genetic algorithm*

In the search through Winston the following addition has been found towards what has already been stated above:

- *Mathematical Model*

Using these concepts, we made a list containing the key theoretical concepts that are applicable to barge scheduling. This list can be found in *section* Key theoretical concepts in barge *2.1 Key theoretical concepts in barge* .

APPENDIX D: SEARCH STRATEGY LITERATURE REVIEW

Number	Criterium	Reason for exclusion
1	“schedul*” OR “plann*” is not mentioned in the abstract	The goal is to build an algorithm for scheduling the containers on barges, therefor planning or scheduling needs to be addressed.
2	Articles written before 1980	There is limited time to conduct a literature research and searching further back would probably not yield better results
3	Field-Weighted Citation Impact is lower than 1.0	According to Löwik, S. (2017) articles are good if they have an impact factor of more than 1.0
4	Written in another language than English	There is no possibility to read for example Chinese articles for me
5	The algorithm or heuristic cannot be applied to barge planning	Algorithms that cannot be applied to barge planning are not relevant for this assignment
Number	**Criterium**	**Reason for inclusion**
1	The abstract has main focus on algorithms, heuristics or mathematical programming	This is the main research subject.
2	Field Weighted Citation Impact is higher than 5.0	If this criterium is not used too many articles showed up to go through within the available time window

Table 12: Inclusion and exclusion criteria

Search string	Scope	Date of Search	Date range	number of entries
TITLE-ABS-KEY (("Schedul*" OR "Plann*") AND mathematical AND programming) AND (LIMIT-TO (LANGUAGE , "English")) AND (LIMIT-TO (DOCTYPE , "ar"))	Title, keywords and abstract	13/05/2017	1980-present	100
TITLE-ABS-KEY (("Schedul*" OR "Plann*") AND mathematical AND model) AND (LIMIT-TO (DOCTYPE , "ar ")) AND (LIMIT-TO (LANGUAGE , "English "))	Title, keywords and abstract	13/05/2017	1980-present	100
TITLE-ABS-KEY (("Schedul*" OR "Plann*") AND simulation) AND (LIMIT-TO (DOCTYPE , "ar ")) AND (LIMIT-TO (LANGUAGE , "English "))	Title, keywords and abstract	13/05/2017	1980-present	100
TITLE-ABS-KEY (("Schedul*" OR "Plann*") AND heuristic) AND (LIMIT-TO (DOCTYPE , "ar ")) AND (LIMIT-TO (LANGUAGE , "English "))	Title, keywords and abstract	13/05/2017	1980-present	100
Total				**400**
Removing Duplicates				-118
Selecting based on exclusion criterium 1 (Scheduling or Planning in abstract)				-62
Removed after reading abstracts and titles with inclusion criterium 1 and exclusion criterium 5 in mind				-211
Total left after criteria and reading abstracts and titles				**9**

Table 13: Bibliographic search

APPENDIX E: SUMMARY OF CONTENT ARTICLES LITERATURE REVIEW

This Appendix contains a summary of the content per article that is found in our literature review.

ALGORITHMS FOR THE VEHICLE ROUTING AND SCHEDULING PROBLEMS WITH TIME WINDOW CONSTRAINTS

In this article, different algorithms for VRSPTW are compared. The authors programmed the algorithms and compared them with different time window relaxations. The main conclusion is that insertion heuristic I1 proved to be very successful. The authors believe that, given its very stable behaviour, the heuristic will perform very well on practical problems.

Insertion Heuristics use two criteria, c1(i, u, j) and c2(i, u, j), after initializing the current route, at every iteration to insert a new customer u into the current partial route, between two adjacent customers i and j on the route.

SCHEDULING WITH BATCHING: A REVIEW

In computational tests with 40, 60 and 80 jobs, and 5 and 10 machines, the best results are obtained with simulated annealing and tabu search, with the former providing slightly better-quality solutions.

We think after reading this article that the content is not applicable to barge planning

TIME WINDOW CONSTRAINED ROUTING AND SCHEDULING PROBLEMS

In many cases k-interchange heuristics work well. Solomon has designed and analysed a variety of route construction heuristics for the VRPTW. Solomon indicates that a *sequential time-space insertion* algorithm proved to be very successful.

SURVEY OF SCHEDULING RESEARCH INVOLVING DUE DATE DETERMINATION DECISIONS

The paper provides a framework for studying scheduling problems involving due date determination decisions. It mainly focuses on ways to determine a due date. Conway studies the effect of various due date assignment methods on the performance of various dispatching rules. He studies: (i) CON, (ii) TWK, (iii) SLK, and (iv) RAN due date methods and finds that the NOP (due dates are determined on basis of the number of operations to be performed on the job; $d_i = r_i + kn_i$) is the most effective method of assigning due dates with respect to the criterion of meeting due dates at high levels of shop utilization. Kanet compares NOP, PPW and TWK due date rules. He finds that TWK (due dates are based on total work content; $d_i = r_i + kp_i$) is superior in terms of mean tardiness performance. Baker also finds that TWK performs best.

NETWORK DESIGN AND TRANSPORTATION PLANNING: MODELS AND ALGORITHMS

The main method used is linear programming. A linear programming with more constraints is faster and easier to solve. This paper mainly focused on planning and routing problems.

COMPUTATIONAL STUDY OF THE JOB-SHOP SCHEDULING PROBLEM

Explaining their way of solving a job shop problem, no comparison with other algorithms.

SIMULATED ANNEALING FOR PERMUTATION FLOW-SHOP SCHEDULING

The simulated annealing performed better on the shift neighbourhood than on the interchange neighbourhood. There is no satisfactory explanation for this result. Fort their 20-job test problems, SA(S,R) produces better solutions than heuristic NEH for 82.5% of the problems

APPLYING TABU SEARCH TO THE JOB-SHOP SCHEDULING PROBLEM

Usually, a tabu list is defined which stores only the opposite of the move applied during the search to transform a solution into a new one (i.e. the move which leads from the new solution to the old one). A solution s' is considered forbidden if the current solution s can be transformed into s' by applying one of the moves in the tabu list.

A NEW HEURISTIC METHOD FOR THE FLOW SHOP SEQUENCING PROBLEM

Not very useful for this thesis. Own developed method, only for faster computations, whereas that is not considered as a bottleneck in this thesis.

APPENDIX F: INTERVIEW FORM

Interview Planners – Scheduling Containers on Barges - 26 oktober 2017

Interviewer
Simon Pruijn
s1598937
06-14512170
sepruijn@gmail.com

Geïnterviewde
Jacob Boorsma

Introductie
Voor mijn afstuderen voer ik een bacheloreindopdracht uit. In deze opdracht kijk ik naar het ondersteunen van het planningsproces voor containers op binnenvaartsschepen met behulp van betere decision support. Hiervoor wil ik graag weten op welk punt er (technische) ondersteuning gewenst is, hoe het planningsproces op dit moment in elkaar steekt en welke factoren belangrijk gevonden worden bij het maken van een planning.

Vragen
1. Kunt u mij in grote lijnen uitleggen hoe het planningsproces in elkaar steekt?
2. Hoe ver van tevoren komt een order gewoonlijk binnen?
3. Kunt u een stappenplan geven van hoe u een planning maakt? Waar wordt als eerste naar gekeken? Waar als tweede?
4. Wat zijn stappen die u vaak repeteert in het planningsproces?
5. Wat zijn stappen die u uitvoert zonder erover na te denken, of om het informeler uit te drukken: op de automatische piloot?
6. Wat zijn belangrijke factoren bij het maken van een planning? Waar wordt op gelet?
7. Op welk punt worden er beslissingen genomen?
8. Wat is een goede beslissing?
9. Wat is een slechte beslissing?
10. Waar is behoefte aan (technische) ondersteuning?
11. In de management wordt vaak gebruik gemaakt van indicatoren die aangeven hoe goed je presteert. Deze Key Perfomance Indicators, ook wel KPI's genoemd kunnen bijvoorbeeld het gemiddeld aantal ziektedagen per werknemer per jaar zijn of hoeveelheid sales in euro's. Welke KPI's let u op bij het maken van een planning?
12. Welke KPI ziet u als het belangrijkst?
13. Zijn er functionaliteiten die u mist of problemen die u ervaart met de huidige decision support tool van NexusZ?
14. Is er nog iets dat u kwijt wil over het planningsproces of wat ik gemist heb?
15. Zijn er nog overige op- of aanmerkingen op dit interview?

APPENDIX G: ANALYSIS WAITING TIMES, HANDLING TIMES AND DIFFERENCE ACTUAL TIME AND PLANNED TIME

For every terminal where 10 or more voyages were available to analyse, we analysed the waiting time, handling time and difference in planned and actual arrival time. This is done by making 3 different box- and whisker plots and by providing a table with mean average, standard deviation, minimum value and maximum value.

Figure 24 shows an example box and whisker plot and how to read it. A box- and whisker plot contains a median (a large line within the box). Exactly half of the measured values lay above the median, and exactly half of the values lay below. The ends of the box represent the lower and upper 25%. This makes that the box represents 50% of all measurements. The lines expanding from the box (in Figure 24 depicted as *a* and *b*) represent the whiskers and are 1.5 times the box size. All points outside this we call outpoints.

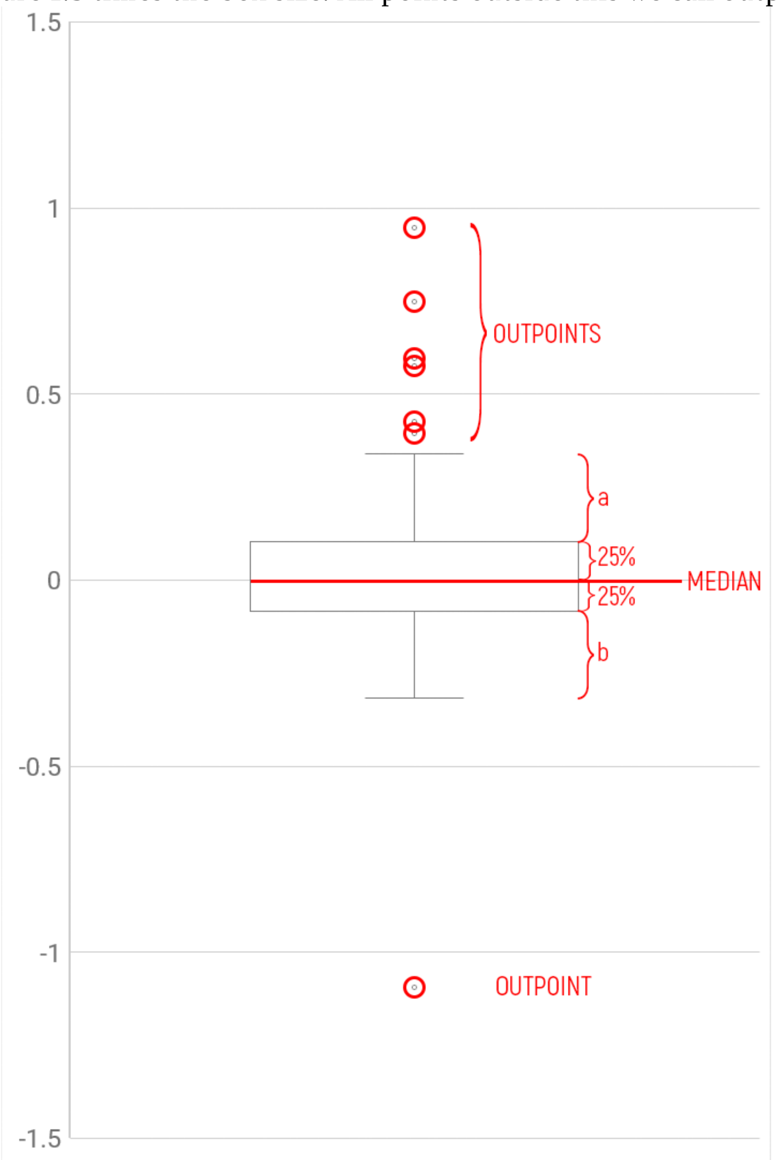


Figure 24 Example image on how to read a box and whiskers plot

The second part of appendix G has been removed from the public version.

APPENDIX H: REGRESSION

To further investigate the dependency of the average handling time on the number of containers we perform linear regression tests with SPSS. In our regression test we removed entries with unrealistic numbers of containers (more than 100) and unrealistic handling times (more than one day). These are probably wrong registered times or when a barge stays overnight. This leads to a smaller number of average containers than found in the earlier analysis. This time we consider a sample set of an average of 16.71 containers per voyage. Hereby an average handling time of 0.03933 days corresponds (almost 57 minutes). The general Descriptive statistics from SPSS are as follows:

Descriptive Statistics

	Mean	Std. Deviation	N
AVG_Handlingtime	.03933394190	.042171926400	678
NO_Containers	16.71	15.901	678

Then we look at correlation between the number of containers and average handling time. We see that there is a correlation of 64% between those two. Meaning that 64% of the handling time is predicted in Table 13. The significance is 0.000, therefore, we can say with a confidence of 0% that there is no correlation between handling time and the number of containers.

Correlations

		AVG_Handlingtime	NO_Containers
Pearson Correlation	AVG_Handlingtime	1.000	.640
	NO_Containers	.640	1.000
Sig. (1-tailed)	AVG_Handlingtime	.	.000
	NO_Containers	.000	.
N	AVG_Handlingtime	678	678
	NO_Containers	678	678

Table 14 SPSS output on correlations

Table 14 below shows that for our linear regression model with SPSS a model can be made that for each container the handling time increases with 0.0016969 days (2 minutes and 26 seconds) and with a handling time of 0.01098 days (15 minutes and 48 seconds) if 0 containers are loaded and unloaded. For the confidence interval of 95%, however, this constant can lay between 0.00743 and 0.0145 days. And the increase per container can lay between 0.00154 days and 0.00185 days.

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.	95.0% Confidence Interval for B	
		B	Std. Error	Beta			Lower Bound	Upper Bound
1	(Constant)	.010983623	.001807467		6.076804085	.000000002	.007434699	.014532548
	NO_Containers	.001696941	7.839490510E-5	.639824907	21.646062730	2.415439320E-79	.001543014	.001850868

a. Dependent Variable: AVG_Handlingtime

Table 15 SPSS regression analysis coefficients

To evaluate further we look at a P-P Plot of the expected against the observed probability. The more towards the line y = x the better. We see a discrepancy between the plot and the line y = x.

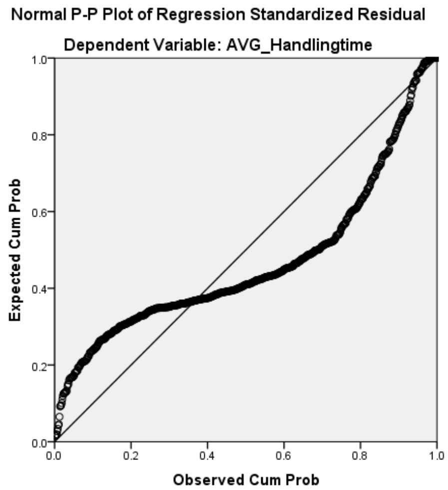


Figure 25 shows the correlation plot between the average handling time (in Days) and the Number of Containers. We draw a trend-line at y = 0.0016969x + .010983623 (the result of our regression analysis). We see that this line represents the trend in the scattering. However, the handling time seems to be dependent on other factors as well. We can also see this when calculating the average handling time per container and standard deviation of it in Excel. The Mean handling time per container is 4 minutes and 29 seconds, while the standard deviation is 5 minutes and 34 seconds.

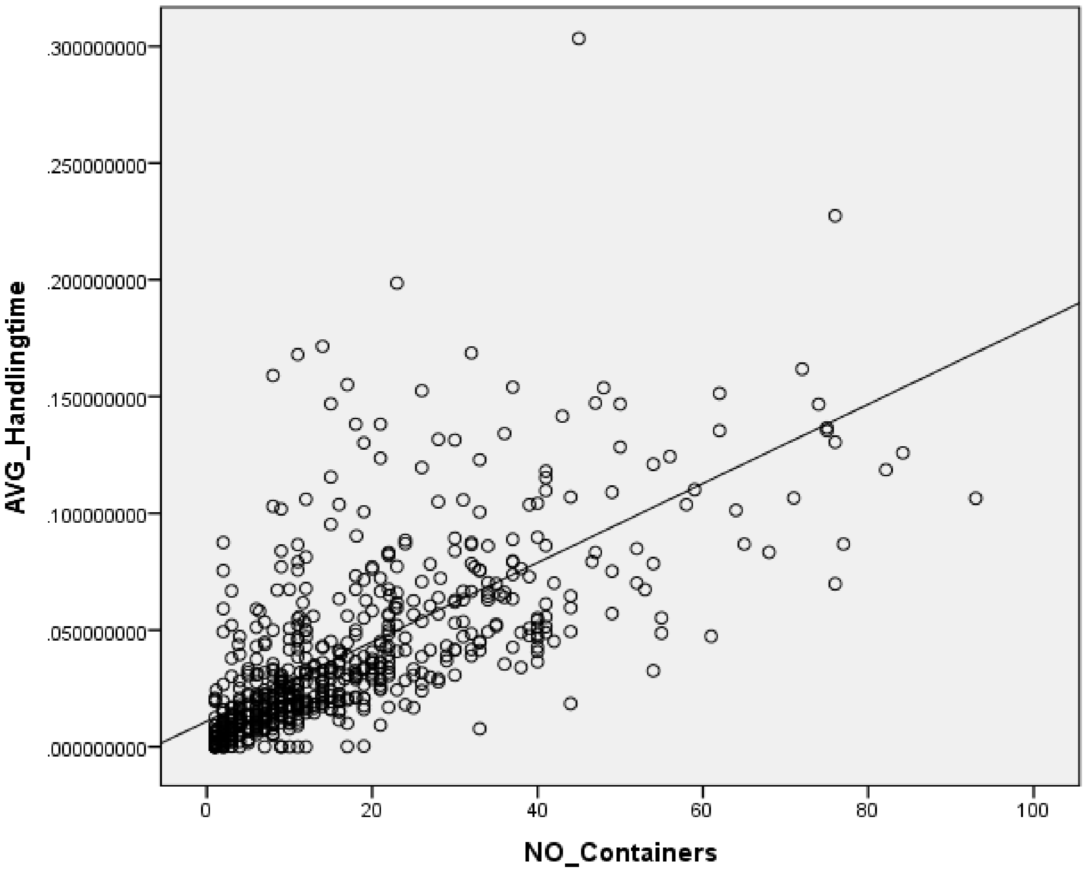


Figure 25 correlation plot for the number of containers with handling time

It seems that the more containers there are, the larger the difference between the handling time and the expected handling time. These are represented respectively by dots and the line in Figure 25. The bar graph in Figure 26 shows the average difference between the handling time and expected handling time in days on the y-axis, for each group of number of containers on the x-axis. We see that for groups containing less than 40 containers, this average difference is lower than 0.025 days, and for 1-10 containers even 0.012 days. For groups with more than 41 containers, this difference seems to be higher. Therfore, the model is more usable to use with less than 40 containers than with higher numbers.

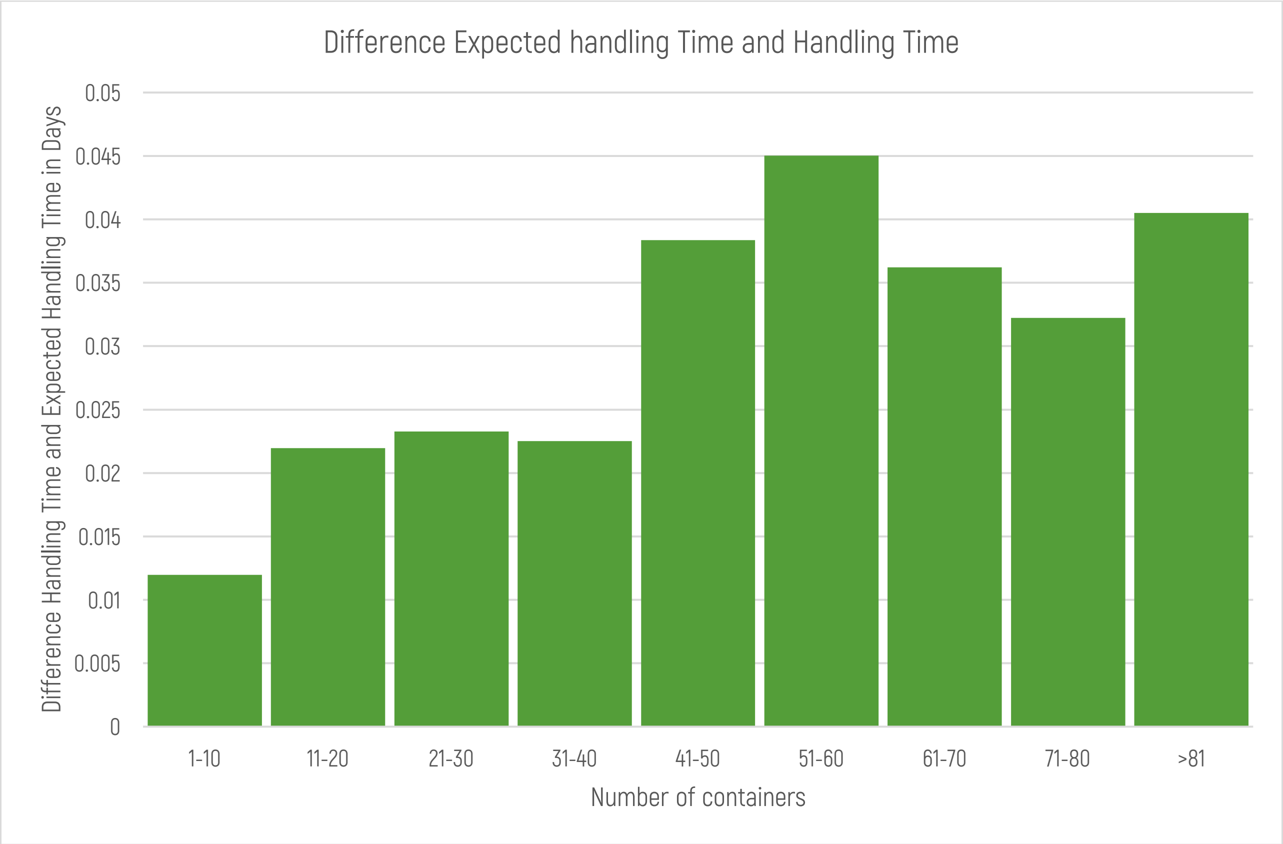


Figure 26 Bar graph difference expected handling time and handling time

APPENDIX I: EPOCH TIMES

NLink uses Epoch milliseconds to store dates and times. The Epoch time, also known as Unix time, is the number of seconds that have elapsed since January 1st, 1920 (Midnight GMT). For example, one day is represented by 86,400,000 milliseconds. A site such as https://www.epochconverter.com/ can be used to convert Epoch time to human readable time.

To convert Epoch to the Excel number system the formula = *((A2 +3600000)/ 86400000) + 25569* can be used and the formatting can be set to Date and/or Time. A2 can be replaced by any cell with an Epoch time in it.

APPENDIX J: INPUT LIST FOR TERMINAL SEQUENCING

The following list is taken from 1 to 25 when moving from the inland terminal to Maasvlakte II and from 25 to 1 when moving backwards from Maasvlakte II towards the inland terminal.

Terminals(1) = "WHT"
Terminals(2) = "BCW"
Terminals(3) = "UP7"
Terminals(4) = "MRS"
Terminals(5) = "PROGECO3"
Terminals(6) = "PROGECO"
Terminals(7) = "UCTEEM"
Terminals(8) = "UCTFRISO"
Terminals(9) = "MBCROT"
Terminals(10) = "KRAREE"
Terminals(11) = "RSTNOORD"
Terminals(12) = "RSTZUID"
Terminals(13) = "INTERFOR"
Terminals(14) = "PCSA"
Terminals(15) = "PCTRO"
Terminals(16) = "CETEM"
Terminals(17) = "WBT"
Terminals(18) = "RCT"
Terminals(19) = "DCS"
Terminals(20) = "DDE"
Terminals(21) = "DDN"
Terminals(22) = "APM"
Terminals(23) = "EUROMAX"
Terminals(24) = "RWG"
Terminals(25) = "APM2"

APPENDIX K: COMPARISON OF THE ALGORITHMS

A comparison between the different algorithms. For explanation, see Chapter 5.

EXPORT 31 OCTOBER 2017																										
	Group 1					Group 2									Group 3			Group 4						Group 5			
	BCW	MRS	PROGECO3	UP7	WHT	INTERFOR	KRAREE	MBCROT	PCSA	PROGECO	RSTNOORD	RSTZUID	UCTEEM	UCTFRISO	CETEM	PCTRO	WBT	APM	DCS	RCT	DDE	DDN	EUROMAX	APM2	RWG	**Total**	#Terminals
Algorithm January	4	0	1	0	0	0	1	20	0	0	0	4	12	0	1	0	0	0	0	12	4	2	13	6	0	**80**	12
Planned	10	0	0	2	0	0	4	5	0	0	0	3	0	0	2	0	0	7	0	4	7	0	17	5	11	**77**	12
Actual	10	0	0	2	0	0	4	5	0	0	0	3	0	0	2	0	0	7	0	4	15	0	17	5	11	**85**	12

Export containers 31 October 2017 allocation comparison

IMPORT 31 OCTOBER 2017																										
	Group 1					Group 2									Group 3			Group 4						Group 5			
	BCW	MRS	PROGECO3	UP7	WHT	INTERFOR	KRAREE	MBCROT	PCSA	PROGECO	RSTNOORD	RSTZUID	UCTEEM	UCTFRISO	CETEM	PCTRO	WBT	APM	DCS	RCT	DDE	DDN	EUROMAX	APM2	RWG	**Total**	#Terminals
Algorithm January	0	3	1	0	24	0	1	1	0	0	0	0	15	1	0	0	2	0	0	9	0	0	0	0	0	**57**	9
Planned	2	3	1	0	10	0	1	11	0	0	2	3	0	1	0	0	2	5	0	19	1	6	1	20	0	**88**	16
Actual	2	3	0	0	10	0	1	11	0	0	2	3	0	1	0	0	2	5	0	19	6	0	1	20	0	**86**	14

Import containers 31 October 2017 allocation comparison

EXPORT 17 NOVEMBER																											
	Group 1					Group 2									Group 3			Group 4						Group 5			
	BCW	MRS	PROGECO3	UP7	WHT	INTERFOR	KRAREE	MBCROT	PCSA	PROGECO	RSTNOORD	RSTZUID	UCTEEM	UCTFRISO	CETEM	PCTRO	WBT	APM	DCS	RCT	DDE	DDN	EUROMAX	APM2	RWG	**Total**	#Terminals
Algorithm January	0	0	13	0	0	0	10	0	0	0	0	0	0	10	0	0	0	15	0	12	0	9	11	0	0	**80**	7
Planned	5	0	0	0	0	0	5	0	0	0	0	7	0	0	0	0	0	6	0	5	30	6	8	0	31	**103**	9
Actual	5	0	0	0	0	0	5	0	0	0	0	7	0	0	0	0	0	0	0	11	30	6	8	0	31	**103**	8

Export containers 17 November 2017 allocation comparison

IMPORT 17 NOVEMBER																											
	Group 1					Group 2									Group 3			Group 4						Group 5			
	BCW	MRS	PROGECO3	UP7	WHT	INTERFOR	KRAREE	MBCROT	PCSA	PROGECO	RSTNOORD	RSTZUID	UCTEEM	UCTFRISO	CETEM	PCTRO	WBT	APM	DCS	RCT	DDE	DDN	EUROMAX	APM2	RWG	**Total**	#Terminals
Algorithm January	6	1	9	0	10	1	2	5	0	16	0	17	7	5	0	0	0	0	0	0	0	0	0	0	0	**79**	11
Planned	0	1	0	0	0	0	0	0	0	0	0	25	0	0	0	0	0	0	0	27	15	4	15	0	1	**88**	7
Actual	0	1	0	0	0	0	0	0	0	0	0	25	0	0	0	0	0	0	0	27	15	4	15	0	1	**88**	7

Import containers 17 November 2017 allocation comparison

EXPORT 6 DECEMBER 2017																										
	Group 1					Group 2									Group 3			Group 4						Group 5			
	BCW	MRS	PROGECO3	UP7	WHT	INTERFOR	KRAREE	MBCROT	PCSA	PROGECO	RSTNOORD	RSTZUID	UCTEEM	UCTFRISO	CETEM	PCTRO	WBT	APM	DCS	RCT	DDE	DDN	EUROMAX	APM2	RWG	**Total**	#Terminals
Algorithm January	0	0	0	4	0	0	0	2	0	0	0	0	0	0	1	0	0	0	0	0	18	0	26	0	29	**80**	6
Planned	3	0	5	0	0	0	0	2	0	0	0	21	0	0	0	0	1	2	0	6	8	5	29	3	13	**98**	12
Actual	3	0	5	0	0	0	0	2	0	0	0	21	0	0	0	0	1	0	0	6	8	5	29	3	13	**96**	11

Export containers 6 December 2017 allocation comparison

IMPORT 6 DECEMBER 2017																										
	Group 1					Group 2									Group 3			Group 4						Group 5			
	BCW	MRS	PROGECO3	UP7	WHT	INTERFOR	KRAREE	MBCROT	PCSA	PROGECO	RSTNOORD	RSTZUID	UCTEEM	UCTFRISO	CETEM	PCTRO	WBT	APM	DCS	RCT	DDE	DDN	EUROMAX	APM2	RWG	**Total**	#Terminals
Algorithm January	0	10	0	7	6	1	0	14	0	3	0	0	29	9	0	0	0	0	0	0	0	0	0	1	0	**80**	9
Planned	0	0	1	7	2	0	2	1	0	0	1	0	15	7	2	0	1	0	0	0	19	31	7	0	0	**96**	13
Actual	0	0	2	7	0	0	2	1	0	0	0	0	15	7	2	0	5	0	0	0	19	31	7	0	0	**98**	11

Import containers 6 December 2017 allocation comparison

	EXPORT 23 DECEMBER 2017																									
	Group 1					Group 2									Group 3			Group 4						Group 5			
	BCW	MRS	PROGECO3	UP7	WHT	INTERFOR	KRAREE	MBCROT	PCSA	PROGECO	RSTNOORD	RSTZUID	UCTEEM	UCTFRISO	CETEM	PCTRO	WBT	APM	DCS	RCT	DDE	DDN	EUROMAX	APM2	RWG	**Total**	#Terminals
Algorithm January	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	33	1	0	0	20	0	**80**	5
Planned	0	0	0	0	0	0	0	4	0	0	0	1	1	0	0	0	0	1	0	19	33	7	0	0	35	**101**	8
Actual	0	0	0	0	0	0	0	4	0	0	0	1	1	0	0	0	0	18	0	54	40	0	0	0	0	**118**	6

Export containers 23 December 2017 allocation comparison

	IMPORT 23 DECEMBER 2017																									
	Group 1					Group 2									Group 3			Group 4						Group 5			
	BCW	MRS	PROGECO3	UP7	WHT	INTERFOR	KRAREE	MBCROT	PCSA	PROGECO	RSTNOORD	RSTZUID	UCTEEM	UCTFRISO	CETEM	PCTRO	WBT	APM	DCS	RCT	DDE	DDN	EUROMAX	APM2	RWG	**Total**	#Terminals
Algorithm January	5	1	2	0	0	0	0	0	0	0	0	0	5	10	0	0	0	0	0	0	0	1	0	0	1	**25**	7
Planned	0	1	7	4	7	1	4	39	0	0	1	2	3	4	0	0	0	13	0	0	0	3	0	2	0	**91**	14
Actual	0	3	7	0	7	0	4	39	0	0	1	2	3	0	0	0	0	13	0	18	3	0	0	2	0	**102**	12

Import containers 23 December 2017 allocation comparison

	EXPORT 9 JANUARY 2018																										
	Group 1					Group 2									Group 3			Group 4						Group 5			
	BCW	MRS	PROGECO3	UP7	WHT	INTERFOR	KRAREE	MBCROT	PCSA	PROGECO	RSTNOORD	RSTZUID	UCTEEM	UCTFRISO	CETEM	PCTRO	WBT	APM	DCS	RCT	DDE	DDN	EUROMAX	APM2	RWG	**Total**	#Terminals
Algorithm January	5	0	0	2	1	0	0	1	0	0	0	0	0	0	0	0	1	4	0	4	13	27	0	0	22	**80**	10
Planned	4	0	0	2	1	0	0	1	0	0	0	0	2	0	0	0	7	9	0	3	0	1	14	1	0	**45**	11
Actual	4	0	0	2	1	0	0	1	0	0	0	0	2	0	0	0	7	9	0	40	0	0	14	1	0	**81**	10

Export containers 9 January 2018 allocation comparison

	IMPORT 9 JANUARY 2018																										
	Group 1					Group 2									Group 3			Group 4						Group 5			
	BCW	MRS	PROGECO3	UP7	WHT	INTERFOR	KRAREE	MBCROT	PCSA	PROGECO	RSTNOORD	RSTZUID	UCTEEM	UCTFRISO	CETEM	PCTRO	WBT	APM	DCS	RCT	DDE	DDN	EUROMAX	APM2	RWG	**Total**	#Terminals
Algorithm January	0	1	0	4	12	0	13	2	0	0	0	0	1	1	1	0	0	0	0	17	0	0	0	27	1	**80**	11
Planned	0	0	1	12	10	0	0	11	0	0	0	5	0	0	0	0	0	0	0	23	2	0	14	8	2	**88**	10
Actual	0	0	1	12	10	0	0	11	0	0	0	5	0	0	0	0	0	0	0	23	0	0	14	8	0	**84**	8

Import containers 9 January 2018 allocation comparison

APPENDIX L: SCHEDULE SEQUENCE COMPARISON

31 OCTOBER

ALGORITHM

TERMINAL	EX-PORT	IMPORT
BCW	4	0
MBCROT	20	1
RSTZUID	4	0
CETEM	1	0
RCT	12	9
DDE	4	0
DDN	2	0
EUROMAX	13	0
APM2	6	0
WBT	0	2
KRAREE	1	1
UCTFRISO	0	1
UCTEEM	12	15
PROGECO3	1	1
MRS	0	3
WHT	0	24

PLANNED

TERMINAL	EX-PORT	IMPORT
UP7	2	0
BCW	10	2
KRAREE	4	1
RSTZUID	3	3
PROGECO3	0	1
UCTFRISO	0	1
MBCROT	5	11
CETEM	2	0
APM	7	5
DDE	7	1
EUROMAX	17	1
DDN	0	6
APM2	5	20
RWG	11	0
RCT	4	19
WBT	0	2
RSTNOORD	0	2
WHT	0	10
MRS	0	3

ACTUAL

TERMINAL	EX-PORT	IMPORT
UP7	2	0
BCW	10	2
KRAREE	4	1
RSTZUID	3	3
UCTFRISO	0	1
MBCROT	5	11
CETEM	2	0
APM	7	5
DDE	15	6
EUROMAX	17	1
APM2	5	20
RWG	11	0
RCT	4	19
WBT	0	2
RSTNOORD	0	2
WHT	0	10
MRS	0	3

17 NOVEMBER

ALGORITHM

TERMINAL	EX-PORT	IMPORT
PROGECO3	13	9
UCTFRISO	10	5
KRAREE	10	2
RCT	12	0
DDN	9	0
APM	15	0
EUROMAX	11	0
INTERFOR	0	1
RSTZUID	0	17
MBCROT	0	5
UCTEEM	0	7
PROGECO	0	16
MRS	0	1
BCW	0	6
WHT	0	10

PLANNED NLINK

TERMINAL	EX-PORT	IMPORT
KRAREE	5	0
RSTZUID	7	0
MRS	0	1
BCW	5	0
DDE	30	15
EUROMAX	8	15
RWG	31	1
DDN	6	4
APM	6	0
RCT	5	27
RSTZUID	0	25

ACTUAL NLINK

TERMINAL	EX-PORT	IMPORT
KRAREE	5	0
RSTZUID	7	0
MRS	0	1
BCW	5	0
DDE	30	15
EUROMAX	8	15
RWG	31	1
DDN	6	4
RCT	11	27
RSTZUID	0	25

6 DECEMBER

ALGORITHM JANUARY 2018

TERMINAL	EXPORT	IMPORT
CETEM	1	0
DDE	18	0
EUROMAX	26	0
RWG	29	0
APM2	0	1
INTERFOR	0	1
MBCROT	2	14
UCTFRISO	0	9
UCTEEM	0	29
PROGECO	0	3
MRS	0	10
UP7	4	7
WHT	0	6

PLANNED NLINK

TERMINAL	EXPORT	IMPORT
BCW	3	0
RSTZUID	21	0
RSTNOORD	0	1
WHT	0	2
MBCROT	2	1
CETEM	0	2
WBT	1	1
RCT	6	0
APM	2	0
RWG	13	0
APM2	3	0
DDN	5	31
EUROMAX	29	7
DDE	8	19
UCTEEM	0	15
KRAREE	0	2
UCTFRISO	0	7
PROGECO3	5	1
UP7	0	7

ACTUAL NLINK

TERMINAL	EX-PORT	IMPORT
BCW	3	0
RSTZUID	21	0
MBCROT	2	1
CETEM	0	2
WBT	1	5
RCT	6	0
RWG	13	0
APM2	3	0
DDN	5	31
EUROMAX	29	7
DDE	8	19
UCTEEM	0	15
KRAREE	0	2
UCTFRISO	0	7
PROGECO3	5	2
UP7	0	7

23 DECEMBER

ALGORITHM JANUARY 2018

TERMINAL	EX-PORT	IMPORT
UP7	1	0
RCT	33	0
DDE	1	0
APM	25	0
APM2	20	0
RWG	0	1
DDN	0	1
UCTFRISO	0	10
UCTEEM	0	5
PROGECO3	0	2
MRS	0	1
BCW	0	5

PLANNED NLINK

TERMINAL	EXPORT	IMPORT
RSTZUID	1	0
RWG	35	0
RCT	19	0
APM2	0	2
DDN	0	3
DDE	33	0
DDN	7	0
APM	1	13
INTERFOR	0	1
PROGECO3	0	7
UCTEEM	1	3
WHT	0	7
RSTZUID	0	2
MRS	0	1
RSTNOORD	0	1
KRAREE	0	4
UP7	0	4
UCTFRISO	0	4
MBCROT	4	39

ACTUAL NLINK

TERMINAL	EX-PORT	IMPORT
RSTZUID	1	0
RCT	54	18
APM2	0	2
DDE	40	3
APM	18	13
PROGECO3	0	7
UCTEEM	1	3
RSTZUID	0	2
MRS	0	3
RSTNOORD	0	1
KRAREE	0	4
UCTFRISO	0	4
MBCROT	4	39
WHT	0	7

9 JANUARY

PLANNED GENERATED

TERMINAL	EXPORT	IMPORT
BCW	4	0
UCTEEM	2	0
WBT	7	0
DDN	1	0
APM	9	0
APM2	1	8
RWG	0	2
EUROMAX	14	14
DDE	0	2
RCT	3	23
RSTZUID	0	5
MBCROT	1	11
PROGECO3	0	1
UP7	2	12
WHT	1	10

PLANNED NLINK

TERMINAL	EXPORT	IMPORT
BCW	4	0
UCTEEM	2	0
WBT	7	0
DDN	1	0
APM2	1	8
APM	9	0
RCT	3	23
DDE	0	2
RWG	0	2
EUROMAX	14	14
MBCROT	1	11
RSTZUID	0	5
PROGECO3	0	1
WHT	1	10
UP7	2	12

ACTUAL NLINK

TERMINAL	EX-PORT	IMPORT
BCW	4	0
UCTEEM	2	0
WBT	7	0
APM2	1	8
APM	9	0
RCT	40	23
EUROMAX	14	14
MBCROT	1	11
RSTZUID	0	5
PROGECO3	0	1
WHT	1	10
UP7	2	12

APPENDIX M: STEP 5 SEQUENCE COMPARISON

31 OCTOBER

PLANNED GENERATED

TERMINAL	EX-PORT	IMPORT
BCW	10	2
UP7	2	0
KRAREE	4	1
CETEM	2	0
DDE	7	1
APM	7	5
EUROMAX	17	1
RWG	11	0
APM2	5	20
DDN	0	6
RCT	4	19
WBT	0	2
RSTZUID	3	3
RSTNOORD	0	2
MBCROT	5	11
UCTFRISO	0	1
PROGECO3	0	1
MRS	0	3
WHT	0	10

PLANNED NLINK

TERMINAL	EX-PORT	IMPORT
UP7	2	0
BCW	10	2
KRAREE	4	1
RSTZUID	3	3
PROGECO3	0	1
UCTFRISO	0	1
MBCROT	5	11
CETEM	2	0
APM	7	5
DDE	7	1
EUROMAX	17	1
DDN	0	6
APM2	5	20
RWG	11	0
RCT	4	19
WBT	0	2
RSTNOORD	0	2
WHT	0	10
MRS	0	3

ACTUAL GENERATED

TERMINAL	EX-PORT	IMPORT
BCW	10	2
UP7	2	0
KRAREE	4	1
CETEM	2	0
DDE	15	6
APM	7	5
EUROMAX	17	1
RWG	11	0
APM2	5	20
RCT	4	19
WBT	0	2
RSTZUID	3	3
RSTNOORD	0	2
MBCROT	5	11
UCTFRISO	0	1
MRS	0	3
WHT	0	10

ACTUAL NLINK

TERMINAL	EX-PORT	IMPORT
UP7	2	0
BCW	10	2
KRAREE	4	1
RSTZUID	3	3
UCTFRISO	0	1
MBCROT	5	11
CETEM	2	0
APM	7	5
DDE	15	6
EUROMAX	17	1
APM2	5	20
RWG	11	0
RCT	4	19
WBT	0	2
RSTNOORD	0	2
WHT	0	10
MRS	0	3

BCW and UP7 are switched in order
UCTFRIOS, MBCROT and RSTZuid are inserted (in reversed order). Probably due to the restriction of time a barge can (un)load at Maasvlakte and the small import numbers.
DDE and APM are switched in order
RWG and APM2 are switched in order
MRS and WHT are switched in order

17 NOVEMBER

PLANNED GENERATED

TERMINAL	EXPORT	IMPORT
BCW	5	0
KRAREE	5	0
DDE	30	15
DDN	6	4
APM	6	0
RWG	31	1
EUROMAX	8	15
RCT	5	27
RSTZUID	7	25
MRS	0	1

PLANNED NLINK

TERMINAL	EX-PORT	IMPORT
KRAREE	5	0
RSTZUID	7	0
MRS	0	1
BCW	5	0
DDE	30	15
EUROMAX	8	15
RWG	31	1
DDN	6	4
APM	6	0
RCT	5	27
RSTZUID	0	25

ACTUAL GENERATED

TERMINAL	EX-PORT	IMPORT
BCW	5	0
KRAREE	5	0
DDE	30	15
DDN	6	4
RWG	31	1
EUROMAX	8	15
RCT	11	27
RSTZUID	7	25
MRS	0	1

ACTUAL NLINK

TERMINAL	EX-PORT	IMPORT
KRAREE	5	0
RSTZUID	7	0
MRS	0	1
BCW	5	0
DDE	30	15
EUROMAX	8	15
RWG	31	1
DDN	6	4
RCT	11	27
RSTZUID	0	25

RSTZuid is split up in NLink for import and export. Large export and import numbers (7 and 25 respectively)
BCW is placed much later in the actual and planned schedules
DDN And RWG And Euromax are in exact opposite order. No logical reason for this.
MRS is positioned after RSTZuid in human schedule/actual since only small number of containers needs to be loaded, and Waalhaven then not has to be visited twice.

6 DECEMBER

PLANNED GENERATED

TERMINAL	EX-PORT	IMPORT
BCW	3	0
PROGECO3	5	1
MBCROT	2	1
RSTZUID	21	0
RCT	6	0
APM	2	0
EUROMAX	29	7
RWG	13	0
APM2	3	0
DDN	5	31
DDE	8	19
WBT	1	1
CETEM	0	2
RSTNOORD	0	1
KRAREE	0	2
UCTFRISO	0	7
UCTEEM	0	15
UP7	0	7
WHT	0	2

PLANNED NLINK

TERMINAL	EXPORT	IMPORT
BCW	3	0
RSTZUID	21	0
RSTNOORD	0	1
WHT	0	2
MBCROT	2	1
CETEM	0	2
WBT	1	1
RCT	6	0
APM	2	0
RWG	13	0
APM2	3	0
DDN	5	31
EUROMAX	29	7
DDE	8	19
UCTEEM	0	15
KRAREE	0	2
UCTFRISO	0	7
PROGECO3	5	1
UP7	0	7

ACTUAL GENERATED

TERMINAL	EX-PORT	IMPORT
BCW	3	0
PROGECO3	5	2
MBCROT	2	1
RSTZUID	21	0
RCT	6	0
EUROMAX	29	7
RWG	13	0
APM2	3	0
DDN	5	31
DDE	8	19
WBT	1	5
CETEM	0	2
KRAREE	0	2
UCTFRISO	0	7
UCTEEM	0	15
UP7	0	7

ACTUAL NLINK

TERMINAL	EX-PORT	IMPORT
BCW	3	0
RSTZUID	21	0
MBCROT	2	1
CETEM	0	2
WBT	1	5
RCT	6	0
RWG	13	0
APM2	3	0
DDN	5	31
EUROMAX	29	7
DDE	8	19
UCTEEM	0	15
KRAREE	0	2
UCTFRISO	0	7
PROGECO3	5	2
UP7	0	7

MBC Rot and RSTZuid switched in order
KRAREE is placed between UCTEEM and UCT-FRISO
Euromax placed later by human planners

23 DECEMBER

PLANNED GENERATED

TERMINAL	EX-PORT	IMPORT
RCT	19	0
DDE	33	0
DDN	7	3
APM	1	13
RWG	35	0
APM2	0	2
INTERFOR	0	1
RSTZUID	1	2
RSTNOORD	0	1
KRAREE	0	4
MBCROT	4	39
UCTFRISO	0	4
UCTEEM	1	3
PROGECO3	0	7
MRS	0	1
UP7	0	4
WHT	0	7

PLANNED NLINK

TERMINAL	EXPORT	IMPORT
RSTZUID	1	0
RWG	35	0
RCT	19	0
APM2	0	2
DDN	0	3
DDE	33	0
DDN	7	0
APM	1	13
INTERFOR	0	1
PROGECO3	0	7
UCTEEM	1	3
WHT	0	7
RSTZUID	0	2
MRS	0	1
RSTNOORD	0	1
KRAREE	0	4
UP7	0	4
UCTFRISO	0	4
MBCROT	4	39

ACTUAL GENERATED

TERMINAL	EX-PORT	IMPORT
RCT	54	18
DDE	40	3
APM	18	13
APM2	0	2
RSTZUID	1	2
RSTNOORD	0	1
KRAREE	0	4
MBCROT	4	39
UCTEEM	1	3
PROGECO3	0	7
MRS	0	3
WHT	0	7

ACTUAL NLINK

TERMINAL	EX-PORT	IMPORT
RSTZUID	1	0
RCT	54	18
APM2	0	2
DDE	40	3
APM	18	13
PROGECO3	0	7
UCTEEM	1	3
RSTZUID	0	2
MRS	0	3
RSTNOORD	0	1
KRAREE	0	4
UCTFRISO	0	4
MBCROT	4	39
WHT	0	7

KRAREE is placed between RSTNoord and UCTFriso. UCT Friso misses (not in the list)
In human planning MRS for no reason placed earlier

9 JANUARY

PLANNED GENERATED

TERMINAL	EXPORT	IMPORT
BCW	4	0
UCTEEM	2	0
WBT	7	0
DDN	1	0
APM	9	0
APM2	1	8
RWG	0	2
EUROMAX	14	14
DDE	0	2
RCT	3	23
RSTZUID	0	5
MBCROT	1	11
PROGECO3	0	1
UP7	2	12
WHT	1	10

PLANNED NLINK

TERMINAL	EXPORT	IMPORT
BCW	4	0
UCTEEM	2	0
WBT	7	0
DDN	1	0
APM2	1	8
APM	9	0
RCT	3	23
DDE	0	2
RWG	0	2
EUROMAX	14	14
MBCROT	1	11
RSTZUID	0	5
PROGECO3	0	1
WHT	1	10
UP7	2	12

ACTUAL GENERATED

TERMINAL	EX-PORT	IMPORT
BCW	4	0
UCTEEM	2	0
WBT	7	0
RCT	40	23
APM	9	0
APM2	1	8
EUROMAX	14	14
RSTZUID	0	5
MBCROT	1	11
PROGECO3	0	1
UP7	2	12
WHT	1	10

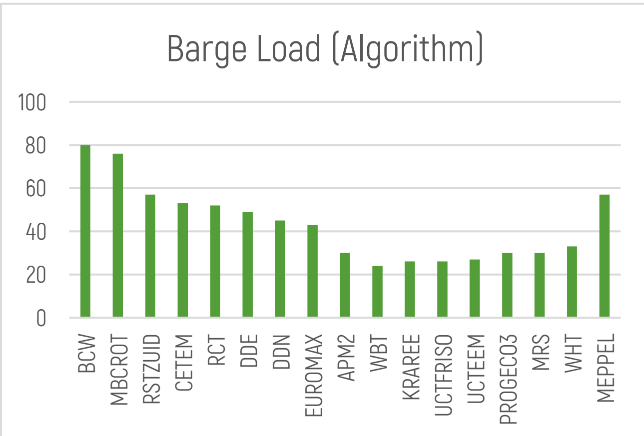
ACTUAL NLINK

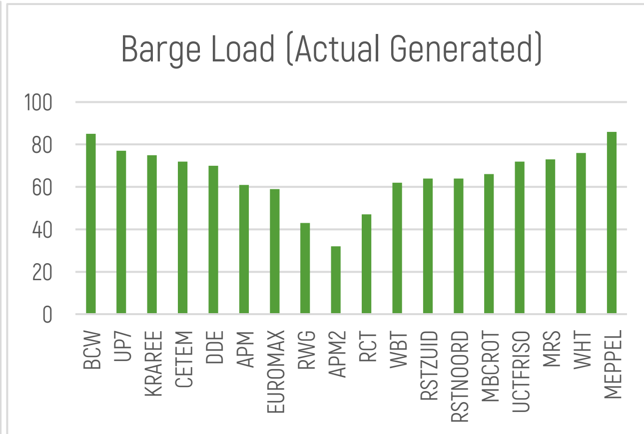
TERMINAL	EX-PORT	IMPORT
BCW	4	0
UCTEEM	2	0
WBT	7	0
APM2	1	8
APM	9	0
RCT	40	23
EUROMAX	14	14
MBCROT	1	11
RSTZUID	0	5
PROGECO3	0	1
WHT	1	10
UP7	2	12

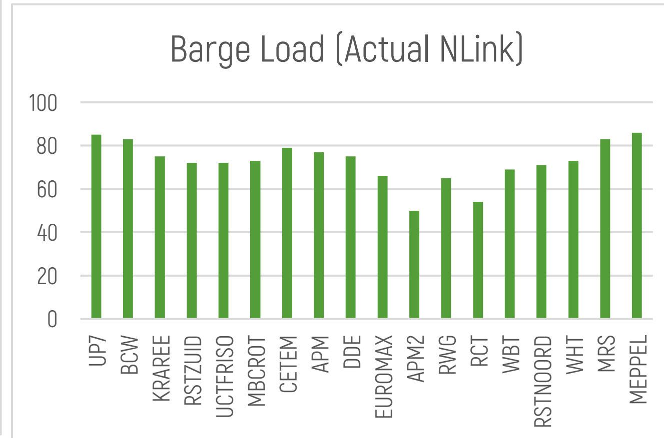
MBCRot and RSTZuid are reversed
WHT and UP7 Are switched in order

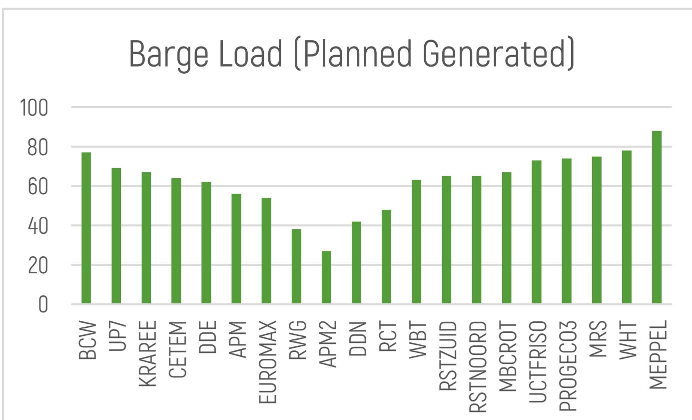
APPENDIX N: BARGE UTILIZATION

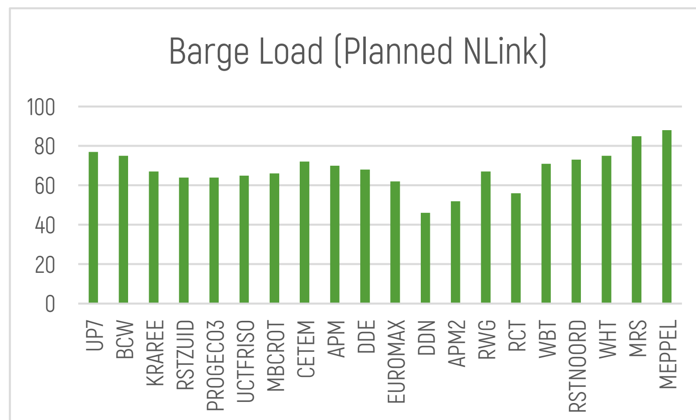
31 OCTOBER



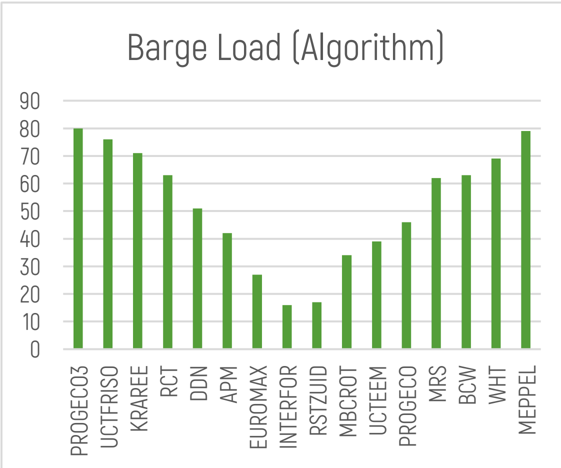


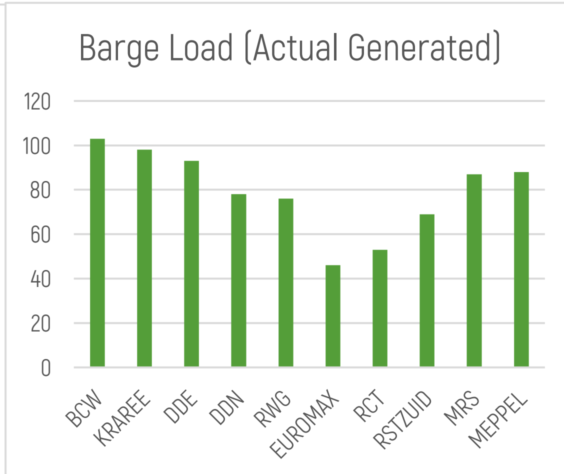


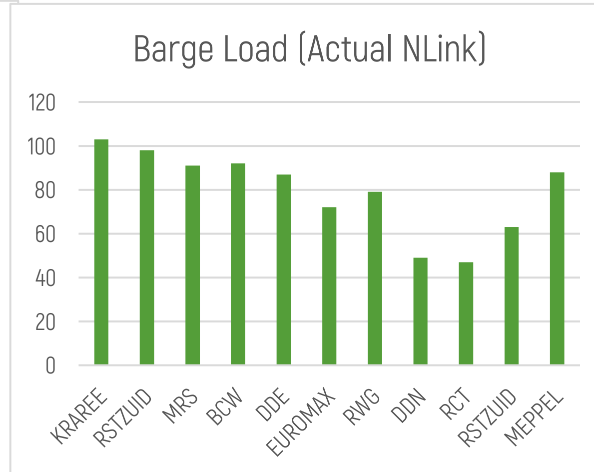


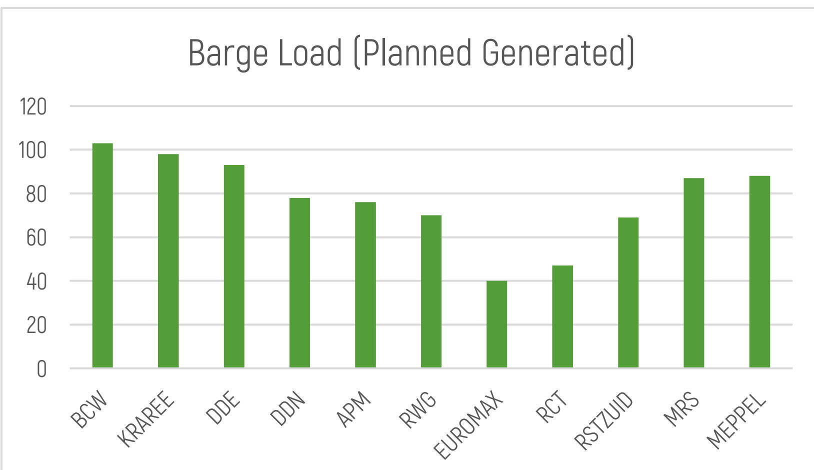


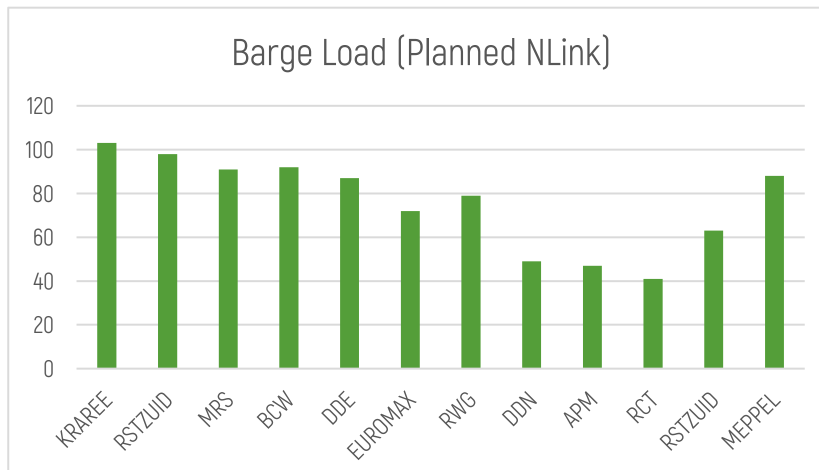
17 NOVEMBER


Barge Load (Algorithm)
90
80
70
60
50
40
30
20
10
0
PROGECO3
UCTFRISO
KRAREE
RCT
DDN
APM
EUROMAX
INTERFOR
RSTZUID
MBCROT
UCTEEM
PROGECO
MRS
BCW
WHT
MEPPEL

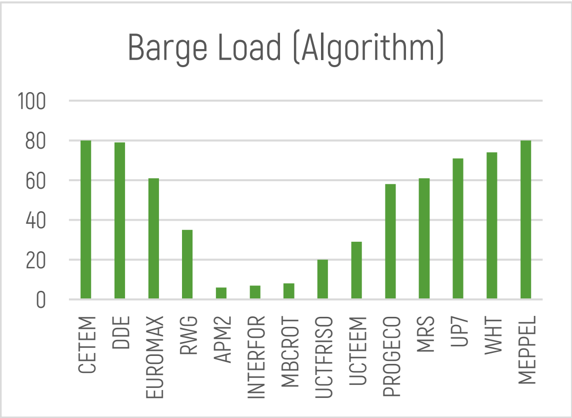

Barge Load (Actual Generated)
120
100
80
60
40
20
0
BCW
KRAREE
DDE
DDN
RWG
EUROMAX
RCT
RSTZUID
MRS
MEPPEL

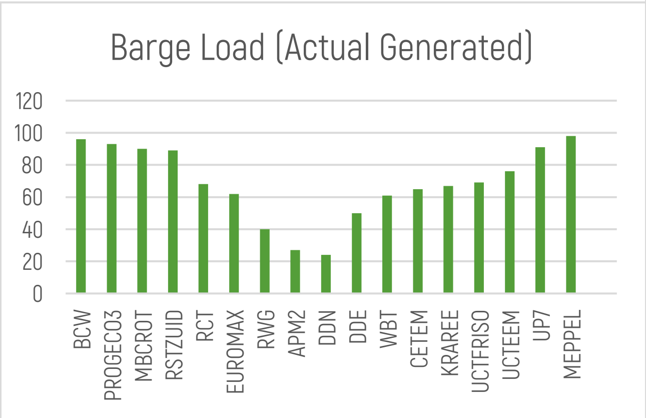

Barge Load (Actual NLink)
120
100
80
60
40
20
0
KRAREE
RSTZUID
MRS
BCW
DDE
EUROMAX
RWG
DDN
RCT
RSTZUID
MEPPEL

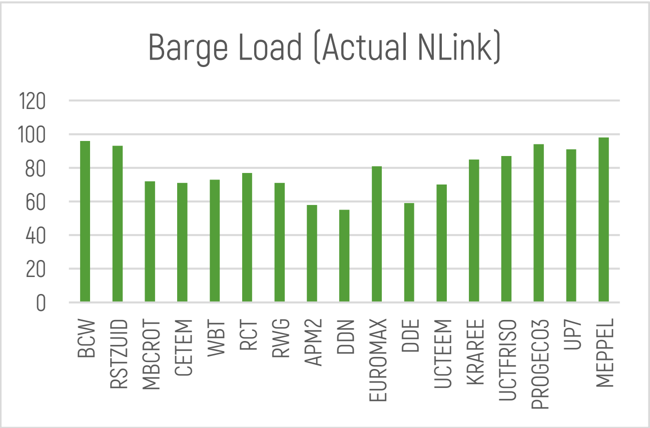

Barge Load (Planned Generated)
120
100
80
60
40
20
0
BCW
KRAREE
DDE
DDN
APM
RWG
EUROMAX
RCT
RSTZUID
MRS
MEPPEL

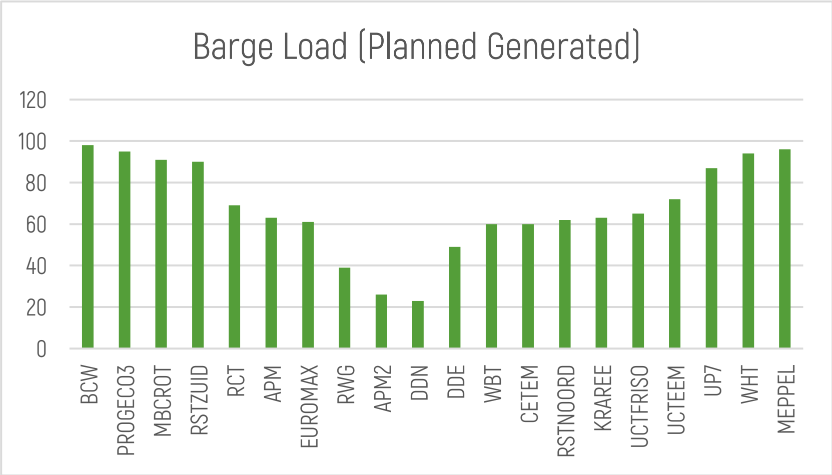

Barge Load (Planned NLink)
120
100
80
60
40
20
0
KRAREE
RSTZUID
MRS
BCW
DDE
EUROMAX
RWG
DDN
APM
RCT
RSTZUID
MEPPEL

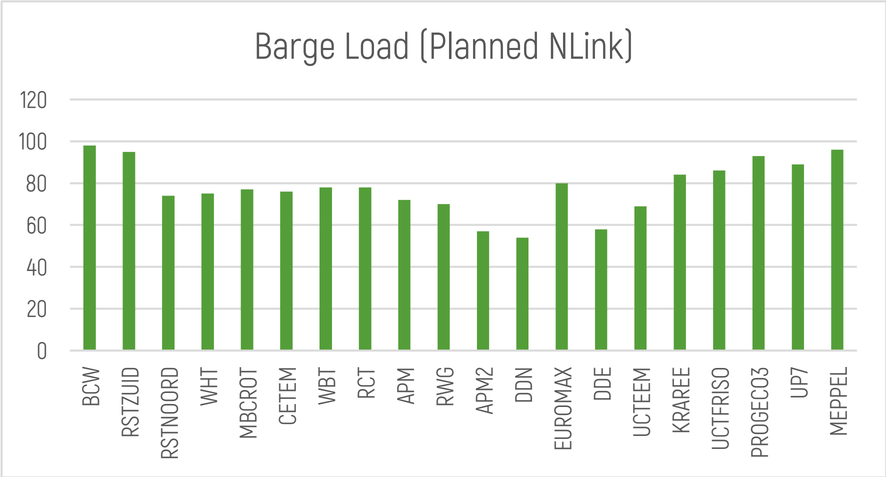
6 DECEMBER


Barge Load (Algorithm)
100
80
60
40
20
0
CETEM
DDE
EUROMAX
RWG
APM2
INTERFOR
MBCROT
UCTFRISO
UCTEEM
PROGECO
MRS
UP7
WHT
MEPPEL

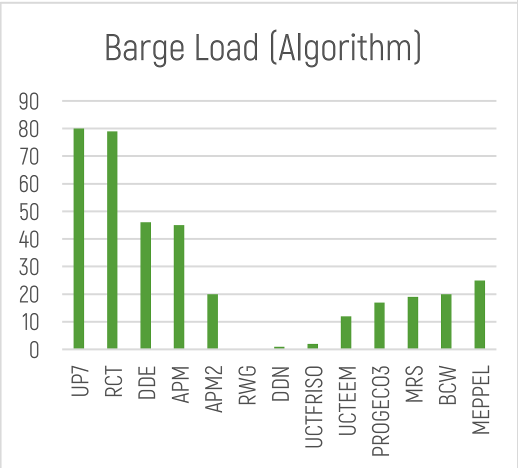

Barge Load (Actual Generated)
120
100
80
60
40
20
0
BCW
PROGECO3
MBCROT
RSTZUID
RCT
EUROMAX
RWG
APM2
DDN
DDE
WBT
CETEM
KRAREE
UCTFRISO
UCTEEM
UP7
MEPPEL

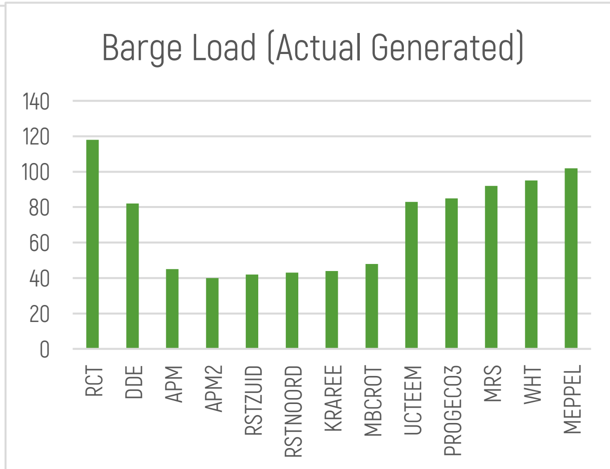

Barge Load (Actual NLink)
120
100
80
60
40
20
0
BCW
RSTZUID
MBCROT
CETEM
WBT
RCT
RWG
APM2
DDN
EUROMAX
DDE
UCTEEM
KRAREE
UCTFRISO
PROGECO3
UP7
MEPPEL

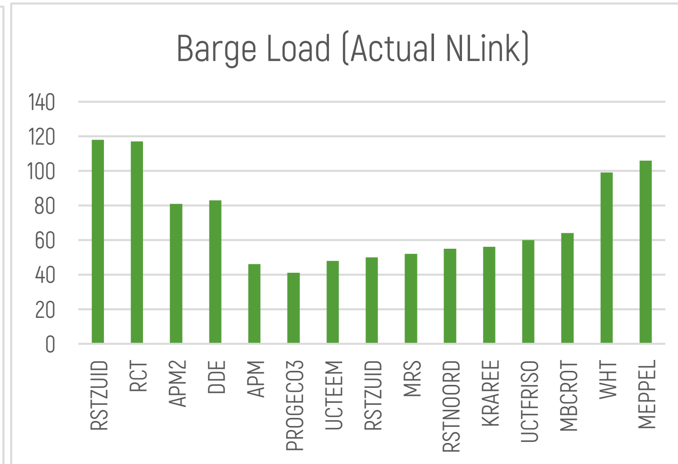

Barge Load (Planned Generated)
120
100
80
60
40
20
0
BCW
PROGECO3
MBCROT
RSTZUID
RCT
APM
EUROMAX
RWG
APM2
DDN
DDE
WBT
CETEM
RSTNOORD
KRAREE
UCTFRISO
UCTEEM
UP7
WHT
MEPPEL

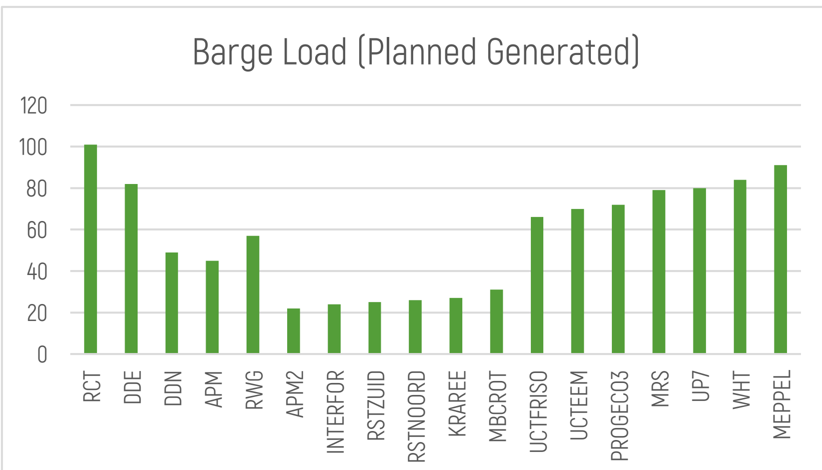

Barge Load (Planned NLink)
120
100
80
60
40
20
0
BCW
RSTZUID
RSTNOORD
WHT
MBCROT
CETEM
WBT
RCT
APM
RWG
APM2
DDN
EUROMAX
DDE
UCTEEM
KRAREE
UCTFRISO
PROGECO3
UP7
MEPPEL

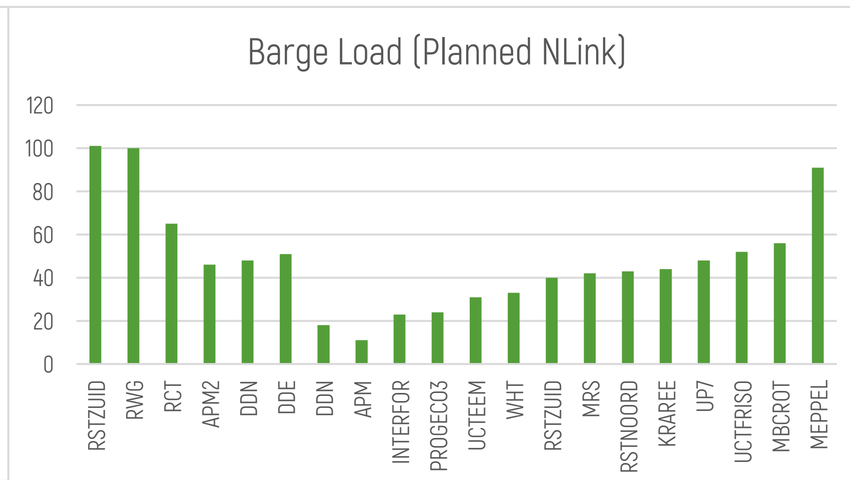
23 DECEMBER


Barge Load (Algorithm)
90
80
70
60
50
40
30
20
10
0
UP7
RCT
DDE
APM
APM2
RWG
DDN
UCTFRISO
UCTEEM
PROGECO3
MRS
BCW
MEPPEL

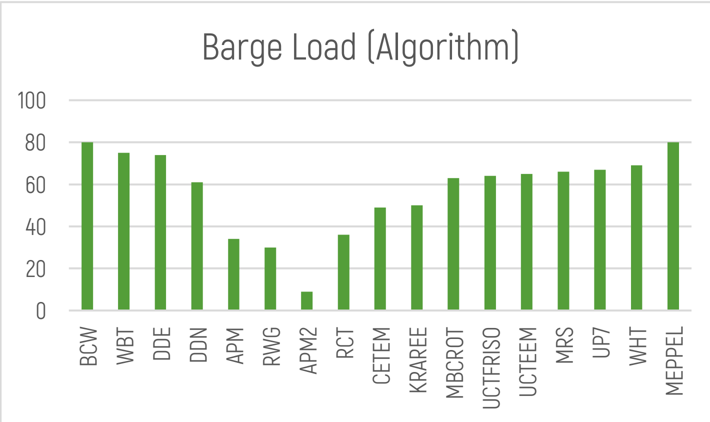

Barge Load (Actual Generated)
140
120
100
80
60
40
20
0
RCT
DDE
APM
APM2
RSTZUID
RSTNOORD
KRAREE
MBCROT
UCTEEM
PROGECO3
MRS
WHT
MEPPEL


Barge Load (Actual NLink)
140
120
100
80
60
40
20
0
RSTZUID
RCT
APM2
DDE
APM
PROGECO3
UCTEEM
RSTZUID
MRS
RSTNOORD
KRAREE
UCTFRISO
MBCROT
WHT
MEPPEL

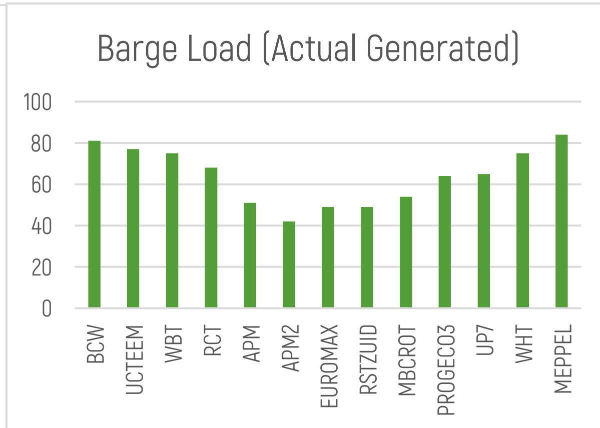

Barge Load (Planned Generated)
120
100
80
60
40
20
0
RCT
DDE
DDN
APM
RWG
APM2
INTERFOR
RSTZUID
RSTNOORD
KRAREE
MBCROT
UCTFRISO
UCTEEM
PROGECO3
MRS
UP7
WHT
MEPPEL


Barge Load (Planned NLink)
120
100
80
60
40
20
0
RSTZUID
RWG
RCT
APM2
DDN
DDE
DDN
APM
INTERFOR
PROGECO3
UCTEEM
WHT
RSTZUID
MRS
RSTNOORD
KRAREE
UP7
UCTFRISO
MBCROT
MEPPEL

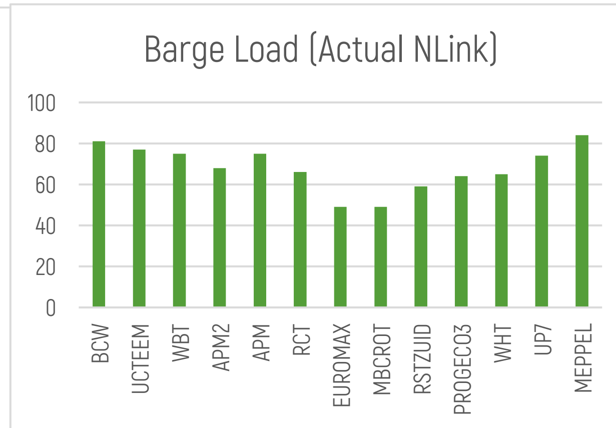
9 JANUARY



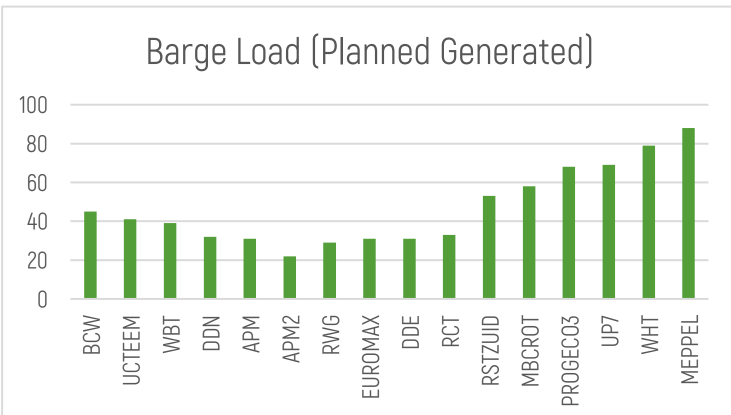
Barge Load (Algorithm)
100
80
60
40
20
0
BCW
WBT
DDE
DDN
APM
RWG
APM2
RCT
CETEM
KRAREE
MBCROT
UCTFRISO
UCTEEM
MRS
UP7
WHT
MEPPEL



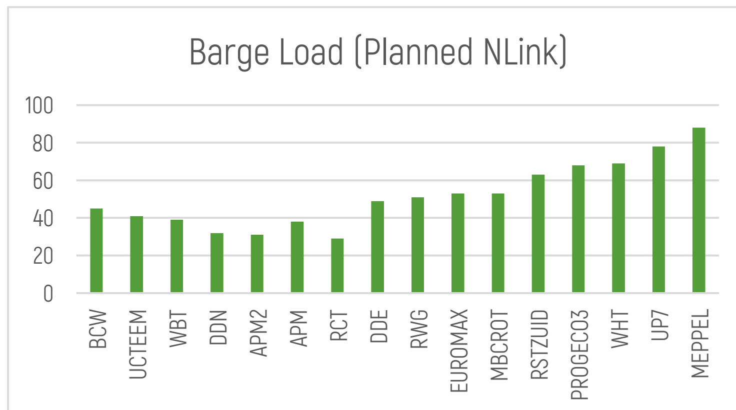
Barge Load (Actual Generated)
100
80
60
40
20
0
BCW
UCTEEM
WBT
RCT
APM
APM2
EUROMAX
RSTZUID
MBCROT
PROGECO3
UP7
WHT
MEPPEL



Barge Load (Actual NLink)
100
80
60
40
20
0
BCW
UCTEEM
WBT
APM2
APM
RCT
EUROMAX
MBCROT
RSTZUID
PROGECO3
WHT
UP7
MEPPEL



Barge Load (Planned Generated)
100
80
60
40
20
0
BCW
UCTEEM
WBT
DDN
APM
APM2
RWG
EUROMAX
DDE
RCT
RSTZUID
MBCROT
PROGECO3
UP7
WHT
MEPPEL



Barge Load (Planned NLink)
100
80
60
40
20
0
BCW
UCTEEM
WBT
DDN
APM2
APM
RCT
DDE
RWG
EUROMAX
MBCROT
RSTZUID
PROGECO3
WHT
UP7
MEPPEL