frontiers in Future Transportation

ORIGINAL RESEARCH
published: 17 August 2021
doi: 10.3389/ffutr.2021.709822

An Evaluation of First-Come, First-Served Scheduling in a Geometrically-Constrained Wet Bulk Terminal

Heleen E. Eisen[1*†], *Joachim E. Van der Lei*[1†], *Joost Zuidema*[2], *Thomas Koch*[3] and *Elenna R. Dugundji*[1,3]

[1]Department of Mathematics, Vrije Universiteit Amsterdam, Amsterdam, Netherlands, [2]Strategy and Innovation, Port of Amsterdam, Amsterdam, Netherlands, [3]Centrum Wiskunde en Informatica (CWI), Amsterdam, Netherlands

In 2022, a new sea lock at IJmuiden is expected to open, permitting an increase in marine traffic of larger ships from the sea to the port. In the interest of facilitating operations, we evaluate the impact of the current first-come, first-served (FCFS) admittance policy in the context of berth allocation for a wet bulk terminal in the port. Four model types are constructed: optimal FCFS; no FCFS with fixed arrival times; 48-h arrival time relaxation; and complete arrival time relaxation. Comparison of the model types is done by means of a rolling time window: of each day within the time frame, a schedule was created for the following 2 weeks, after which the objective value was calculated. When comparing the average of all objective values, it was found that the optimal FCFS model already shows an improvement compared to the historical situation. Between the FCFS and the no FCFS model, there are no considerable differences, because the vessels are constrained to be scheduled on/after their arrival time at the port. When relaxation is allowed, a considerable efficiency gain is possible, especially if larger ships arrive at the port.

Keywords: berth allocation problem, wet bulk terminal, parallel machine scheduling, "first come first served", lock admittance policy

OPEN ACCESS

Edited by:
Edwin Van Hassel,
University of Antwerp, Belgium

Reviewed by:
Edwar Lujan,
National University of Trujillo, Peru
Flabio Gutierrez,
National University of Piura, Peru

***Correspondence:**
Heleen E. Eisen
h.e.eisen@student.vu.nl

†These authors have contributed equally to this work and share first authorship

Specialty section:
This article was submitted to Freight Transport and Logistics, a section of the journal Frontiers in Future Transportation

Received: *14 May 2021*
Accepted: *02 August 2021*
Published: *17 August 2021*

Citation:
Eisen HE, Van der Lei JE, Zuidema J, Koch T and Dugundji ER (2021) An Evaluation of First-Come, First-Served Scheduling in a Geometrically-Constrained Wet Bulk Terminal. Front. Future Transp. 2:709822. doi: 10.3389/ffutr.2021.709822

1 INTRODUCTION

In 2022, a new sea lock at IJmuiden is expected to open (Port of Amsterdam, 2018). Upon completion, this lock will have the title for being the world's largest sea lock, permitting an increase in marine traffic of larger ships from the sea to the port. In the interest of facilitating operations, we evaluate the impact of the current first-come, first-served (FCFS) admittance policy in the context of berth allocation for a wet bulk terminal in the port.

For marine traffic management and channel security reasons, sea vessels are only admitted to the port when a berth at a terminal is available. Berth allocation in research is known as the Berth Allocation Problem (BAP), which involves assigning available berths to incoming ships subject to the given time and berth layout constraints. There are usually two objective functions: to reduce the overall waiting time of ships at the port, and to minimise the probability that an incoming ship's docking request will be refused. According to (Imai et al., 2003), the BAP can have a decisive impact on the efficiency of port operations. However, there are many side constraints involved in BAP, which can hamper the process of allocation.

A special aspect of this case is that the Port of Amsterdam largely serves bulk cargo, rather than containers. Berth allocation for container terminals has been widely studied, but, research on bulk ports is relatively scarce. In this paper we evaluate alternatives to the current berth allocation strategy which done on a first-come, first-served basis. Additionally, since larger vessels need to be able to moor, the current layout of the terminals has to be considered besides the temporal aspect of the scheduling, taking into account that the ships need enough space to maneuver in and out of a berth. Consequently, the BAP for the Port of Amsterdam involves both the spatial and the temporal constraints.

The aim of this research is to investigate if there are other ways for berth allocation that optimise the total berth usage. This is done by first addressing three subproblems: deriving the duration of stay from historical Automatic Identification System (AIS) signals, using this data to predict time windows for how long arriving ships will stay, and determining the berth occupancy in the terminal based on AIS data. Then, four types of models are used to assign berths to arriving ships: an optimal first-come, first-served model, a model where only the arrival times are fixed, a model with arrival time relaxation, and lastly a model with complete arrival time relaxation, meaning a vessel can be scheduled anytime before their estimated arrival time. The outcomes of these models can be used to test if changing the first-come, first-served approach would be beneficial for a wet bulk terminal in the port. The study on this terminal can serve as an example case for other terminals in the port: it can show that further efficiency can be achieved by scheduling in advance and can be used as incentive for other terminals to become more inventive.

This paper first gives background information on the Berth Allocation Problem, then discusses the different data sources that were made available. After this, the methods used to address the BAP are discussed, split up into first deriving the duration of stay, then some information about scheduling and the berth allocation problem, and finally the problem formulation of the BAP of this case. The arrival of larger ships is simulated by considering a series of increasing percentages of the historical ships enlarged to the maximum size permitted by the geometrical constraints of the terminal, and updating their time window for duration of stay accordingly. After presenting the results for the scenarios with current sized ships and larger ships, we discuss how our analytical berth allocation model can be applied for real-time operations. We conclude giving recommendations for further research and application.

2 BACKGROUND

This paper studies the Berth Allocation Problem (BAP) from the perspective of port-scheduling literature. Bierwirth and Meisel (2015) counted 79 new models for berth allocation after 2009, indicating a rising demand for such algorithms. The same authors classified such models in previous research (Bierwirth and Meisel, 2010) based on four features: spatial attribute, temporal attribute, handling time attribute, and the performance measure. The spatial attribute describes the layout of the berths, which can be discrete (BAP-D: one boat can dock at a single berth at a time) (Brown et al., 1994), continuous (BAP-C: boats can moor at random places within the boundaries of the dock as long as they do not overlap with other vessels' positions) (Lim, 1998), or hybrid (one boat is allowed to occupy more than one berth at a time). All these were proven to be NP-hard problems (Garey and Johnson, 1979).

The temporal attribute concerns the arrival of the boats, and can be static (SBAP: vessels arrived already and can moor at any time as long as there is a berth available) (Imai et al., 1997), dynamic (DBAP: allows ships to arrive at individual and deterministic times, while work is in progress) (Imai et al., 2001), cyclic (vessels return to the berth repeatedly after a fixed time), or stochastic (arrival times are defined by stochastic parameters). When it comes to the handling time of the vessels there are five categories: fixed handling times, dependency on the berthing position, dependency on the number of cranes, dependency on the work schedules of the assigned cranes, and combinations of last three (Kovac, 2017). The performance measure concerns the objective functions, which most of the time relate to minimising the time spent by vessels within the port. In terms of the complexity of BAP, heuristic methods are generally used (Clements et al., 1997; Lim et al., 2004; Smith and Pyle, 2004).

Accordingly, it was determined that this case can be classified as discrete, dynamic, with fixed handling times, minimising on one hand the sum of all the departure times summed together, for the client benefit, and on the other the departure time of the last vessel, towards the terminal advantage. Additionally, the terminal on which the analysis is performed is a liquid bulk terminal that handles clean petroleum products. One paper that investigated the BAP within bulk port terminals is (Umang et al., 2013), which also considered dynamic arrivals. However, the spatial aspect was hybrid, which differs from the present research. Moreover, (Umang et al., 2013), addressed the BAP with a different objective, namely minimising the total service completion time, taking into account the cargo types, the conveyors and the pipelines.

3 DATA

This paper uses three main sources of data: historical AIS data, the terminal operational spatial constraints, and the operations data provided by terminal.

3.1 Historical AIS Data

The Automatic Identification System (AIS) is a marine tracking system that is widely used by vessels to avoid collisions and by port authorities to coordinate the traffic within harbours. A vessel informs the harbour about its speed, orientation, and navigational status. Besides the dynamic information, the quay keeps track of other static variables, such as the ID, the name, the type, and the size of each ship.

The AIS data contains every AIS signal received in geographical bounding box around the entire port area from

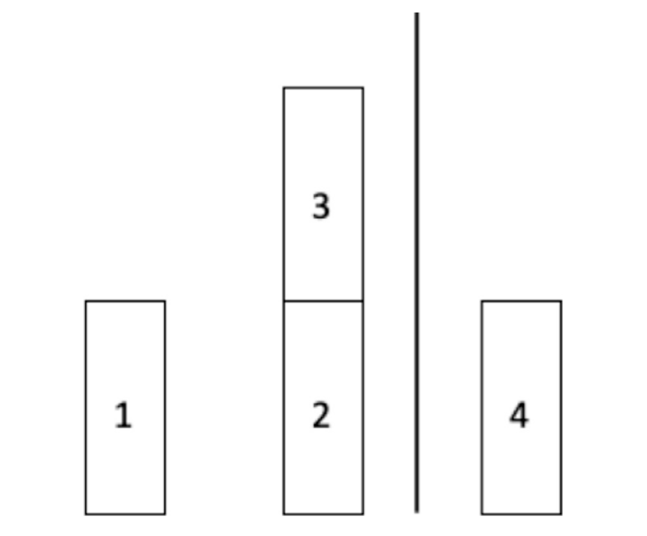


FIGURE 1 | Berth layout of the EVOS Terminal.

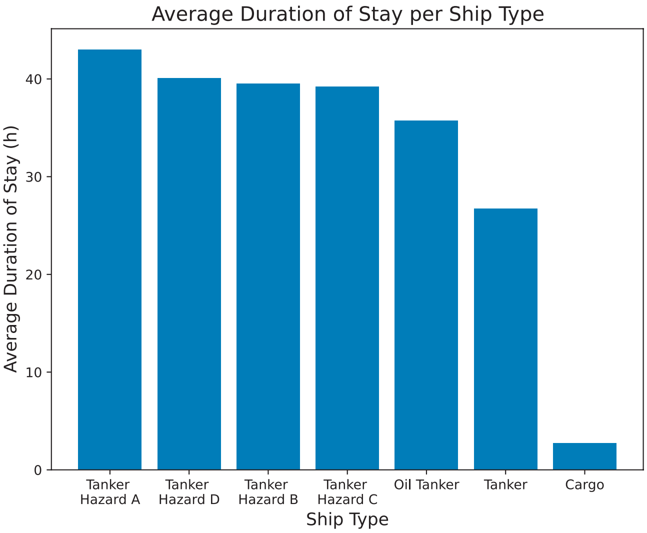


FIGURE 2 | Average duration of stay per ship type.

May 2017 to May 2020, where every row represents one signal instance. This data is split per day and has nineteen features, including the exact timestamp when the signal took place. Nevertheless, several entries are missing or are filled in with a default value when they are not available. For instance, when the draught value is unknown, a 0 is assigned to the respective entry. This affects the quality of the data and requires careful manipulation.

3.2 Terminal Operational Space Constraints

The Terminal Operational Space Constraints (in Dutch: Besluit Aanwijzing Operationele Ruimte) is a matrix that defines the spatial constraints of the terminal. This matrix exists because large vessels can cause manoeuvring problems for other vessels entering or leaving the terminal. For example, as can be seen in **Figure 1**, if a ship placed at berth 3 is too large in width, other ships are not able to leave or reach berths 1 and 2. The TOSC describes which combinations of ship widths of berths 1, 2, and 3 are feasible, based on previously conducted simulations.

The TOSC is essential in the spatial aspect of the problem, where the maximum width of 106 for the ships docked at berths 1, 2, and 3 is used in the mathematical model conducted in **Section 4.3**. A maximum width of 106 is chosen, as the TOSC specifies that as long as the sum of the total width of the constrained berths is smaller than or equal to 106, there are no restrictions.

3.3 Terminal Operations Data

A wet bulk terminal has provided operations data over the period of December 2019–April 2020 that includes more detailed information on the operation, such as the type and quantity of cargo, the pump start and end times, and also the time that the Notice Of Readiness (NOR) was tendered. The latter is given off by a vessel once it has arrived to the waiting area and it is ready to enter the lock. The terminal operations data can be matched up with the AIS data to get a more in-depth view of the entire process.

4 METHODS

In this section, it is first described how the duration of stay is derived from AIS data. After that, some information on scheduling and the berth allocation problem, as well as the problem formulation itself are given. This section also explains the different model types created during the study.

4.1 Deriving Duration of Stay

A few steps have to be taken to get from the raw AIS data to the duration of stay of a particular ship. The first step in the first approach is to reduce the dataset to only contain the ships located in the terminal, and to only contain the ship types relevant for this study: larger, seagoing vessels. The next step is to group all signals sent out by one vessel (when it is at a certain berth). After this, the five most recent AIS signals of each ship are taken, after which it is checked if the centroid of these locations is at a berth in the terminal. This method seems to perform well, as the vast majority of entries can be found within a 1.5 h margin (this means, if a ship arrival or departure is recorded by the terminal at a certain time, the calculated arrival based on AIS is no more than 1.5 h earlier or later). The entries that do not match up, are mostly entries that do not fall in the 1.5 h margin or do not meet the search criteria; for this approach to record an entry, a ship has to send out an AIS signal repeatedly and the centroid of the last five points has to be within a berth. This means that if an AIS signal sent by a ship does not come from a berth repeatedly, no entry is recorded. Unfortunately, after saving the results it became clear that in many cases, several separate entries are recorded for a vessel instead of one that spans across the entire stay. Therefore, if there are several entries recorded for one vessel, yet the difference between the end of the last stay and the beginning of a new one is smaller than or equal to the threshold (in this case set to 2 h), the two entries are merged into one. An important assumption made is that the vessels that are taken into account in this data (mainly tankers and oil tankers) do not return to the terminal within 2 hours of their departure.

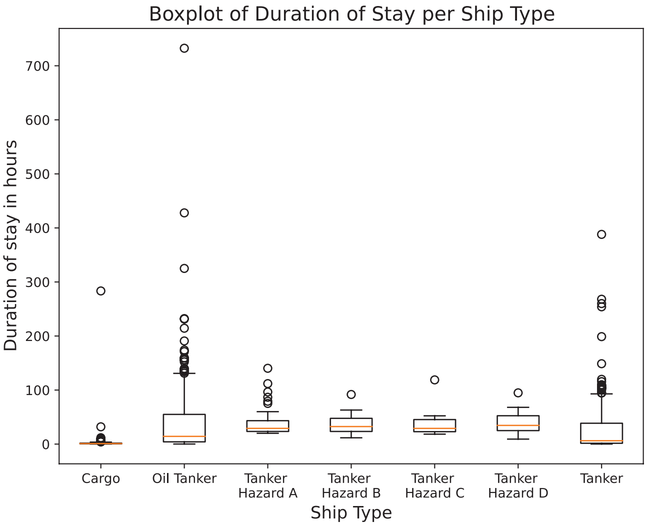


FIGURE 3 | Boxplot of duration of stay per ship type.

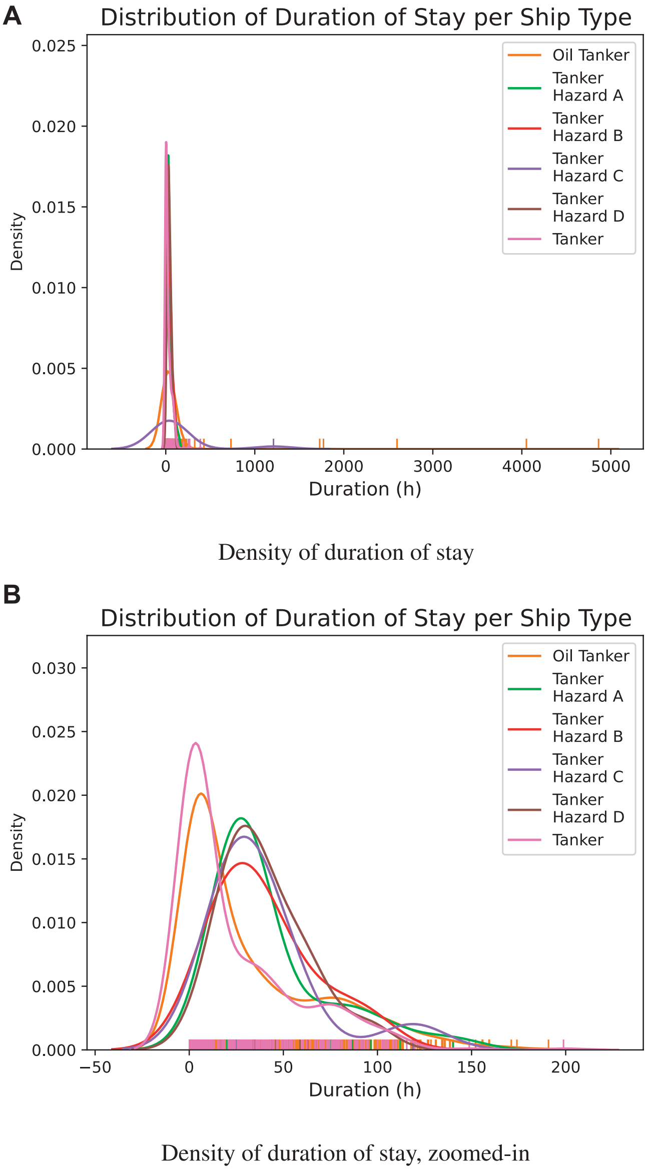


FIGURE 4 | Density of duration of stay. **(A)** Regular, **(B)** Zoomed-in.

Once the duration of stay is calculated, the average duration of stay per ship type can be made, as shown in **Figure 2**. **Figure 3** shows the corresponding box plots per ship type. **Figure 4** shows two distribution plots of the generated duration of stay, where **Figure 4B** is a zoomed in version of **Figure 4A**. It can be noted that ship types "Oil Tanker" and "Tanker" have many entries around 0, which could be due to the "supporting" vessels that are only present at a berth to support other (larger) vessels. The stays of these vessels are also registered in the duration of stay when deriving it from AIS data.

4.2 Predicting Duration of Stay

During the modeling phase, four Machine Learning models were developed to predict the duration of stay: Linear Regression, Random Forest, Gradient Boosting, and Neural Network. The initial judgment behind building four models was to compare them and determine which one generates the most accurate predictions or, in other words, which has the lowest root-mean-square error (RMSE).

Firstly, the Duration of Stay dataset from AIS was randomly split in 80% training and 20% test set, such that the precision of the models could be justly assessed. Unfortunately, during this phase it was discovered that the size of around 1,200 instances is extremely small for a Machine Learning method to learn the underlying patterns from the training data and have enough records left to test the model, considering that it is recommended that the test data has at least 500 records. Accordingly, the goal shifted towards providing a ranking of the four models given multiple samplings, as well as offering insights into the way their prediction power can be improved. In other words, for each sampling of training and test data, the four models were built and their RMSE was compared to see if there is a model that performs generally better.

The first model built was Linear Regression. In order to select the explanatory features, the Backward Elimination technique, also known as the Step-down method, was used. This method works as follows: it first computes the Linear Regression considering all the variables, testing each in a t-test. If the largest p-value of a feature is greater than the significance level (0.05), then the variable is removed. The entire algorithm is performed repeatedly until this is no longer the case. Consequently, this method is executed during the training phase to select the best features for the Linear Regression method.

The second model is the Random Forest. First, the model was performed given 1,000 trees in the forest as the only hyperparameter. Afterward, the following six hyperparameters were tuned using the Random Search method: the number of trees, the number of features considered for the best split, the minimal size of the terminal nodes, the minimum number of

TABLE 1 | RMSE of each prediction model of the test set.

Algorithms data sampling	Random state 1	Random state 2	Random state 3	Random state 4	Random state 5
Linear regression with feature selection	62.68	346.42	311.81	60.15	58.85
Random forest (MMSI)	80.95	329.05	329.04	192.85	80.18
Random forest (MMSI) with hyperparameter tuning	59.88	341.06	316.57	68.25	58.93
Random forest (/MMSI)	72.11	329.39	325.37	199.72	50.52
Random forest (/MMSI) with hyperparameter tuning	59.55	340.55	315.71	67.16	56.01
Gradient boosting	126.05	321.28	346.83	244.31	116.43
Gradient boosting with hyperparameter tuning	82.7	322.68	322.44	149.27	66.83
Neural network	136.56	343.83	321.91	118.46	88.95
Neural network with hyperparameter tuning	61.35	344.89	312.61	68.09	54.29

samples required to split an internal node, and the possibility to bootstrap samples when building a tree. The Random Search method works by randomly selecting one element from the given options for each hyperparameter. Next, it splits the training data into five parts and consecutively trains the Random Forest with the given parameters on four parts of the data, while validating the performance on the fifth part. Finally, the five scorings are averaged and stored as a result of a single iteration. After all the 20 iterations are computed, the Random Search method returns the hyperparameters of the model that performed best.

The next Machine Learning model built was the Gradient Boosting. In a similar fashion as the Random Forest, the model was first tested on three default parameters, the number of boosted trees (1,000), the number of leaves (1,000), and the learning rate (0.05). Afterward, the Random Search tuning method was used again to check the best performing hyperparameters on the validation data. The parameters tuned this time were the learning rate of the model, the number of leaves for base learners, the number of boosted trees, and the maximum tree depth.

The last model is a Neural Network with four layers, including the input and the output ones. The first three layers have 32 neurons each and “relu” activation. The model has “adam” activation and mean-square error loss. Before the Neural Network was fed with the train data, this data was first normalised to change the values to a general scale, keeping the ranges between values. Without this technique applied to the data, the learning would have oscillated as a result of the varying gradients. After the creation of this model with a batch size of 10 and 100 epochs, the Grid Search method was used to tune these two hyperparameters. The tuning method works in a similar fashion with the Random Search, however, the first scheme tries all the combinations of the given finite set of hyperparameters, validating their prediction power using five-fold cross-validation.

Table 1 presents the RMSE of each model over the test dataset given a different sampling when splitting the data in the training and test sets. From this table, it can be seen that there is no clear pattern with regards to which model performs generally better. Moreover, the model with the tuned hyperparameters performed worse than the basic one multiple times. This can be explained by the bad choice of options or their bad sampling. One possible solution to this issue is switching the tuning method to Grid Search, which tries all the combinations, unlike the Random Search, which randomly selects cases. Moreover, several options per hyperparameter can be tried such that the quality of the hyperparameters is ensured.

The most important conclusion to be drawn from this analysis is that the lack of data does not allow a stable training and testing environment for a prediction model. Hence, acquiring more data, either by using the algorithm which extracts the duration of stay from the AIS data for terminals other than EVOS, or by using other sources, is the first step that needs to be taken towards building an accurate prediction model. Moreover, adding weather data has the potential to improve the prediction, since it was often noted by the Port of Amsterdam and EVOS employees that weather affects the duration of stay. When it comes to the technical part of the models, better tuning of hyperparameters is recommended, especially for the Neural Network, which only considers the batch size and the number of epochs. Accordingly, the number of layers, the number of neurons per layer, their activation, and the model optimiser can also be tuned.

This prediction model could be implemented in a realtime application, as there, the ETA of vessels is unknown. Note that in **section 4.3**, this prediction model is *not* used as the ETA of the historically arrived vessels is known. Therefore, predicting the ETA was unnecessary.

4.3 The Berth Allocation Problem

4.3.1 Scheduling

Generally, scheduling problems are described with a machine that can process jobs (j), where an optimal schedule is to be found for a given objective function. These problems can be written in the form $\alpha|\beta|\gamma$, where α, β and γ represent the characteristics of the machine(s), job(s) and objective function respectively. Characteristics of the machines include how many machines there are and if they are identical or not. If machines are identical, the processing of jobs takes the same amount of time on each machine (parallel machines, P), whereas if machines are not similar, the processing time for a job on one machine differs from the processing time on another machine (unrelated machines R). Regarding jobs, these sometimes have restrictions, for example when they have to be completed before their due date (d), or when they can only be processed after their release date (r). There are also different objective functions, such as minimising the sum of the completion time of each job (C_j), or minimising the maximum completion time, where the completion time is the starting time of a job (s_j) + the processing time of a job (p_j).

Using the theory as described above, the berth allocation problem can be formulated as a parallel machine scheduling problem. It is defined as $P|r_j|\sum C_j$. P indicates that there are parallel machines; r_j indicates the release dates of the jobs; $\sum C_j$ is the objective function that aims to minimise the sum of the completion times of the jobs. In this scenario, the machines can be thought of as the berths, with four parallel machines representing the four berths at the wet bulk terminal. The jobs are then the ships, where their duration of stay is their processing time p_j. This model also takes into account the (estimated) arrival times as the berth allocation is dynamically processed: a ship is only ready to dock once it has arrived. Thus the Estimated Time of Arrival (ETA) of a ship can be seen as the release date (r_j) of a ship.

In addition, since the terminal has a more sophisticated spatial restriction −namely the triangular layout of three of the four berths−, the allocation model stays discrete. This means only one vessel is allowed to dock at one berth at a time. Furthermore, whenever the service of a vessel begins (i.e., whenever it is docked), it cannot be disrupted. To therefore efficiently use the dock, the main aim is to optimise the berths usage rate, which is equivalent to minimising the total completion time of all jobs, or minimising the last completion time over all jobs.

4.3.2 Problem Formulation

The whole berth allocation problem can be denoted as a general $P|r_j|\sum C_j$ (or $P|r_j|C_{\max}$) scheduling problem. To solve the problem, it is treated as a linear programming problem and is formulated as follows (Xu et al., 2012; Umang et al., 2013):

$$\text{Minimise} \quad \sum_{j=1}^{n} C_j \tag{1}$$

$$\text{Or Minimise} \quad C_{max} \tag{2}$$

$$\text{s.t.} \quad \sum_{i=1}^{4} x_{ij} = 1 \quad (j = 1, 2, \dots, n) \tag{3}$$

$$s_j \geq a_j \quad (j = 1, 2, \dots, n) \tag{4}$$

$$s_{j'} \geq s_j \tag{5}$$

$$(j, j' = 1, 2, \dots, n \quad \text{s.t. } a_j \leq a_{j'})$$

$$s_j \geq a_j - e_j \quad (j = 1, 2, \dots, n) \tag{6}$$

$$C_{max} \geq s_j + p_j \quad (j = 1, 2, \dots, n) \tag{7}$$

$$s_j \geq (r_i + B) x_{ij} \qquad (j = 1, 2, \dots, n; i = 1, 2, 3, 4) \tag{8}$$

$$s_{j'} \geq s_j + p_j - M(1 - I_{ijj'}) \tag{9}$$

$$(j, j' = 1, 2, \dots, n \quad \text{s.t. } j \neq j'; i = 1, 2, 3, 4)$$

$$I_{ijj'} + I_{ij'j} \leq \frac{1}{2}(x_{ij} + x_{ij'}) \tag{10}$$

$$(j, j' = 1, 2, \dots, n \quad \text{s.t. } j < j'; i = 1, 2, 3, 4)$$

$$I_{ijj'} + I_{ij'j} \geq x_{ij} + x_{ij'} - 1 \tag{11}$$

$$(j, j' = 1, 2, \dots n \quad \text{s.t. } j < j'; i = 1, 2, 3, 4$$

$$d_j x_{1j} + d_{j'} x_{2j'} + d_{j''} x_{3j''} \leq 106 \qquad (j, j', j'' = 1, 2, \dots, n) \tag{12}$$

$$x_{ij} \in \{0, 1\} \qquad (j = 1, 2, \dots, n; i = 1, 2, 3, 4) \tag{13}$$

$$I_{ijj'} \in \{0, 1\} \qquad (j, j' = 1, 2, \dots, n \quad \text{s.t. } j \neq j'; i = 1, 2, 3, 4) \tag{14}$$

With variables:
n vessels (i.e. vessels 1, 2, . . . , n).
4 berths (i.e., berths 1, 2, 3, 4), where berth 4 is not part of the triangular berth layout.
$x_{ij} \in \{0, 1\}$: binary variable equal to 1 if vessel $j \in n$ is scheduled at berth $i \in \{1, 2, 3, 4\}$, 0 otherwise.
$I_{ijj'} \in \{0, 1\}$: binary variable equal to 1 if vessels j, j', $j \neq j'$ are both scheduled at berth $i \in \{1, 2, 3, 4\}$, and vessel j is scheduled before vessel j', 0 otherwise
s_j: the starting processing time of vessel j, $s_j \geq 0$.
p_j: the given processing time of vessel j (duration of stay of vessel j), $p_j > 0$.
a_j: the arrival time of vessel j (ETA of j), $a_j \geq 0.C_j$: the completion time of vessel j, $C_j > 0$.
$C_{\max}$: the last completion time among all vessels, $C_{\max} > 0$.
d_j: the width of vessel j, $d_j > 0$.
e_j: the earlier arrival margin of vessel j in hours, $0 \leq e_j \leq 48$.
r_i: the remaining docking time of the currently docked vessel at berth i, $r_i \geq 0$.
B: the constant buffer time between each two vessels' berthing, $B > 0$ (in this case 2 h).
M: a big constant number (in this case 10000).

The objective functions:

1) Minimise the sum of completion times over all ships
2) Minimise the maximum completion time among all ships

With constraints:

3) Requires each vessel to be assigned to one berth
4) Requires that each vessel can start its processing only after it has arrived at the terminal
5) Requires vessel j' to be scheduled after j given that $a_j \leq a_{j'}$ (only active when the model enforces a FCFS strategy)
6) Applies arrival relaxation, meaning vessels can be scheduled up to e_j (48) hours earlier than their (a_j) (only active when a_j relaxation is allowed)
7) Ensures $C_{\max}$ is the last completion time among all vessels [only active when the objective function is (2)]
8) Ensures vessels are scheduled only after the vessels that are currently at the terminal have left
9) If vessels j, j' are both assigned to berth i and vessel j is processed before vessel j' (i.e. $I_{ijj'} = 1$), then the start time of vessel j' must be no earlier than $s_j + p_j$
10) and 11) Ensure that one of the $I_{ijj'}$ and $I_{ij'j}$ equals 1 if vessels j and j' are both assigned to berth i. They also ensure that $I_{ijj'} = I_{ij'j} = 0$ if one of vessels j and j' is not assigned to berth i
12) Ensures the spatial constraint under the regulation of the terminal's triangular berth layout.
13) $x_{ij} = 1$ if vessel j is assigned to berth i; $x_{ij} = 0$ otherwise
14) $I_{ijj'} = 1$ if vessels j, j' are both assigned to berth i and vessel j is processed before vessel j'; $I_{ijj'} = 0$ otherwise.

In the scheduling, a buffer of 2 h is added (based on recommendations by Port of Amsterdam). The duration of stay in principle already includes the time for maintenance, checking, cleaning, etc., but the buffer is necessary to allow vessels that finish their service to depart the terminal and vessels that will start their service to arrive to the terminal. Furthermore, in order to generalize this model to other terminals in the port, constraint 12) can be omitted or replaced by another terminal-specific constraint.

4.3.3 Different Model Types

The model is created such that four types arise:

1) first come, first served (FCFS) with fixed arrival times, also called optimal FCFS
2) No FCFS with fixed arrival times
3) No FCFS with 48 h relaxation, meaning that ships can be scheduled up to 48 h earlier than their estimated arrival
4) No FCFS with complete relaxation, meaning that ships can now be scheduled anytime before their estimated arrival.

These types are introduced to display the impact of adjusting the current scheduling policy. Furthermore, two different objective functions are used to represent the two different perspectives: the terminal and the client perspective. The objective function $\sum C_j$ minimises the sum of the completion times over all vessels and would hence fit the client perspective. Since the duration of stay is fixed, vessels will be scheduled as early as possible, meaning the waiting times for vessels overall are minimised. On the other hand, the objective function $C_{\max}$ minimises the maximum completion time, which fits the terminal perspective to handle as many vessels as possible in the same amount of time. In both cases of the objective function, the waiting time for individual vessels could in some cases increase significantly, even though the average waiting time decreases.

5 RESULTS

This section first explains how the results are obtained, then presents the results for simulations with regular sized and larger ships respectively.

5.1 Application of Model

The linear model as described in the section Problem Formulation is solved using PuLP[1], a linear solver package in Python. An outline of the code is available upon request. The results are split up into two scenarios, being the same ships as in the past arriving at the terminal, and the potentially larger ships arriving at the terminal after the new sea lock is used. The data used here is the derived duration of stay data, matched up with the terminal operations data over the time period December 2019 until April 2020. This matching was necessary as the duration of stay data also showed ships that supported larger ships, for example with ship-to-ship operations. As the model should only schedule the incoming seagoing vessels, only ships were included that had a recorded stay in both the generated duration of stay and the terminal operations data.

For both objective functions, the model runs for a 14-day period as the time frame input where it could test different scheduling possibilities. In order to best simulate the berth allocation and avoid specificities caused by certain time frames, each day from December 2019 to April 2020 is used as an entry for the model. Both scenarios (regular sized and larger ships) are tested with four types of model (optimal FCFS, no FCFS, 48 h relaxation, and complete relaxation) for two objective functions ($\sum C_j$ and $C_{\max}$).

A crucial step is applied to ensure a fair comparison with historical data and show the results of our models are at least as good as FCFS. Instead of simply summing up the completion times of all ships newly scheduled within the given time frame, only ships actually berthed during this time period according to the historical data are again scheduled by the model within the time frame. For example, a ship with an ETA on the last day of the 14-day period can according to the model be scheduled at this day, yet historically this ship actually berthed a day later, outside of our time frame. Consequently, this resulted in a substantial deviation to the objective value. Therefore, only scheduling the same ships that docked in the past could avoid such a problem.

Another crucial step is that for the simulation of historical schedules, the NOR (Note of Readiness) is used as arrival time instead of the ETA, as the ETA from AIS data is highly unreliable. This step, in addition to the step described previously, is essential for a fair comparison between historical values and the optimal values according to the model.

5.2 Simulation With Regular Sized Ships

For regular sized ships, the average of summed completion times is taken as the judging criteria of the model's performance, which is in a format of percentage with reference to historical AIS data. **Figure 5** shows the results over the entire time period for the two objective functions and model type 1: the optimal FCFS model. In these plots, a sharp decrease of ships berthed can be noted during mid February. This is due to a server error at the collection site, meaning there is no AIS data available for this time period (February 16–27, 2020). In order to maintain the rolling time window, it was decided to leave those entries in.

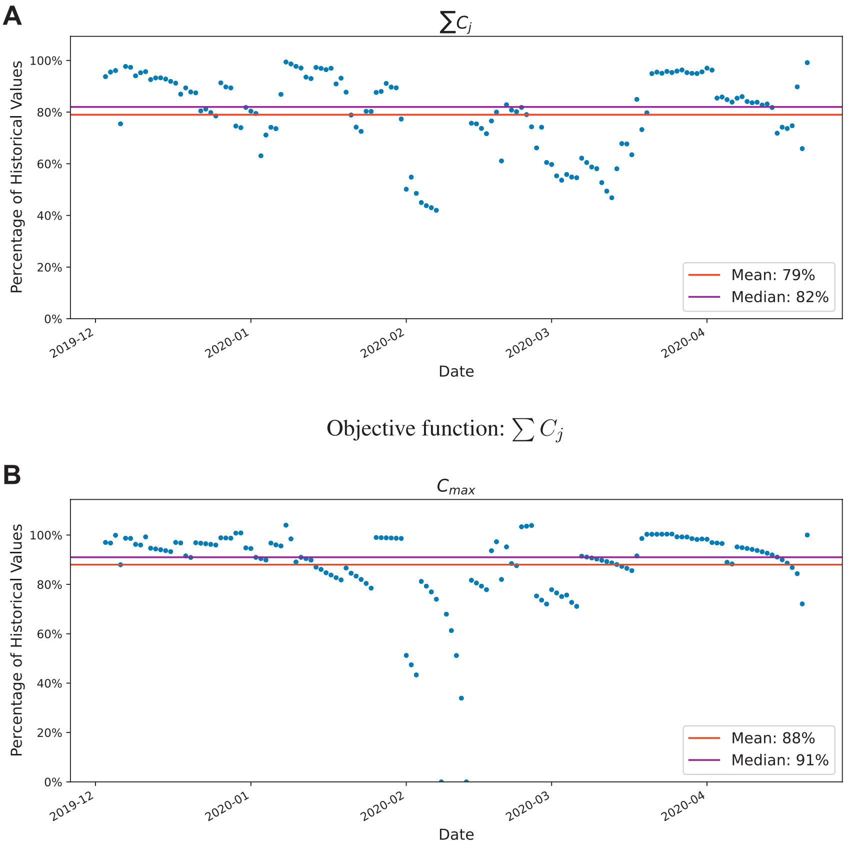
Comparison with historical FCFS berthing already shows an outperformance with the new, optimal FCFS model, with 21 and 12% improvements in the mean objective value of $\sum C_j$ and $C_{\max}$ respectively. No FCFS model has almost the same objective value as optimal FCFS does. This is because both models have fixed arrival times and vessels can only be docked after they have arrived. As a result, not many new schedules could be rearranged. In contrast, no FCFS models with 48 h and complete relaxation could save significant time in both objective functions. This effect can also be seen in **Figure 6**, where an example schedule of each model type is displayed using the objective function $\sum C_j$.

Furthermore, an inspection of changes in individual waiting times is also performed. Some vessels are scheduled earlier than historical allocation and some are rescheduled later consequently. In the corresponding model, the overall individual waiting time is synergistically varied with the objective function.

5.3 Simulation With Larger Ships

To simulate the arrival of larger ships, the same input is taken as for the regular sized ships, only a portion of 25, 50, and 75% of the ships are enlarged to the theoretical maximum sizes that the terminal can handle (meaning dimensions of 285 × 48 × 14.05). The duration of stay for these ships is set according to the rule of thumb as provided by the terminal: 1,250 m^3/h + 1 extra hour for each step to be loaded (in this case 2 h to have an extra margin).

[1] https://pypi.org/project/PuLP/.



Objective function: $\sum C_j$

Objective function: C_{max}

FIGURE 5 | 14-days percentage of historical values of model 1 (optimal FCFS model). **(A)** Objective function: $\sum C_j$, **(B)** Objective function: C_{max}.

Since larger ships will only arrive after the new sea lock has become operational, performances (in terms of the mean objective value) shown in **Table 2** are compared with the optimal FCFS model, which is initialised as 100%. The No FCFS model generates similar results as optimal FCFS because of the same reason as mentioned in section Simulation with Regular Sized Ships. Likewise, smaller objective values are obtained in no FCFS models with relaxations in all scenarios, meaning abandoning the FCFS principle and allowing ships to come earlier could shorten the overall berthing time.

6 DISCUSSION

The simulations performed in this study can also be applied in real time. This section first explains how such a real-time model would work, and then explains the importance of an accurate prediction model.

6.1 A Real-Time Model

As the results have demonstrated there are theoretical improvements that can be made when optimising berth scheduling using our model. However, in order to show that this is also applicable in day to day activities a real-time, more user-centric model had to be designed. In addition, the Port of Amsterdam had expressed that the model should not aim to replace the harbour master, but provide valuable assistance in the decision making process. Therefore, it was essential to firstly understand the needs and responsibilities of our target user: the harbour master.

In order to build up this knowledge, collaboration with port authorities during the entire development process was necessary and provided a clear and concise list of requirements for our model. This means that our model needs these features:

- Real-time overview of incoming ships
- Real-time overview of terminal usage
- Real-time scheduling options (Assistance in berth allocation)

Based on these requirements a web based dashboard as displayed in **Figure 7** was designed and built using Flask[2], a micro web framework written in Python. Flask was chosen with future development in consideration, as it is modular and

[2] https://flask.palletsprojects.com/en/1.1.x/.

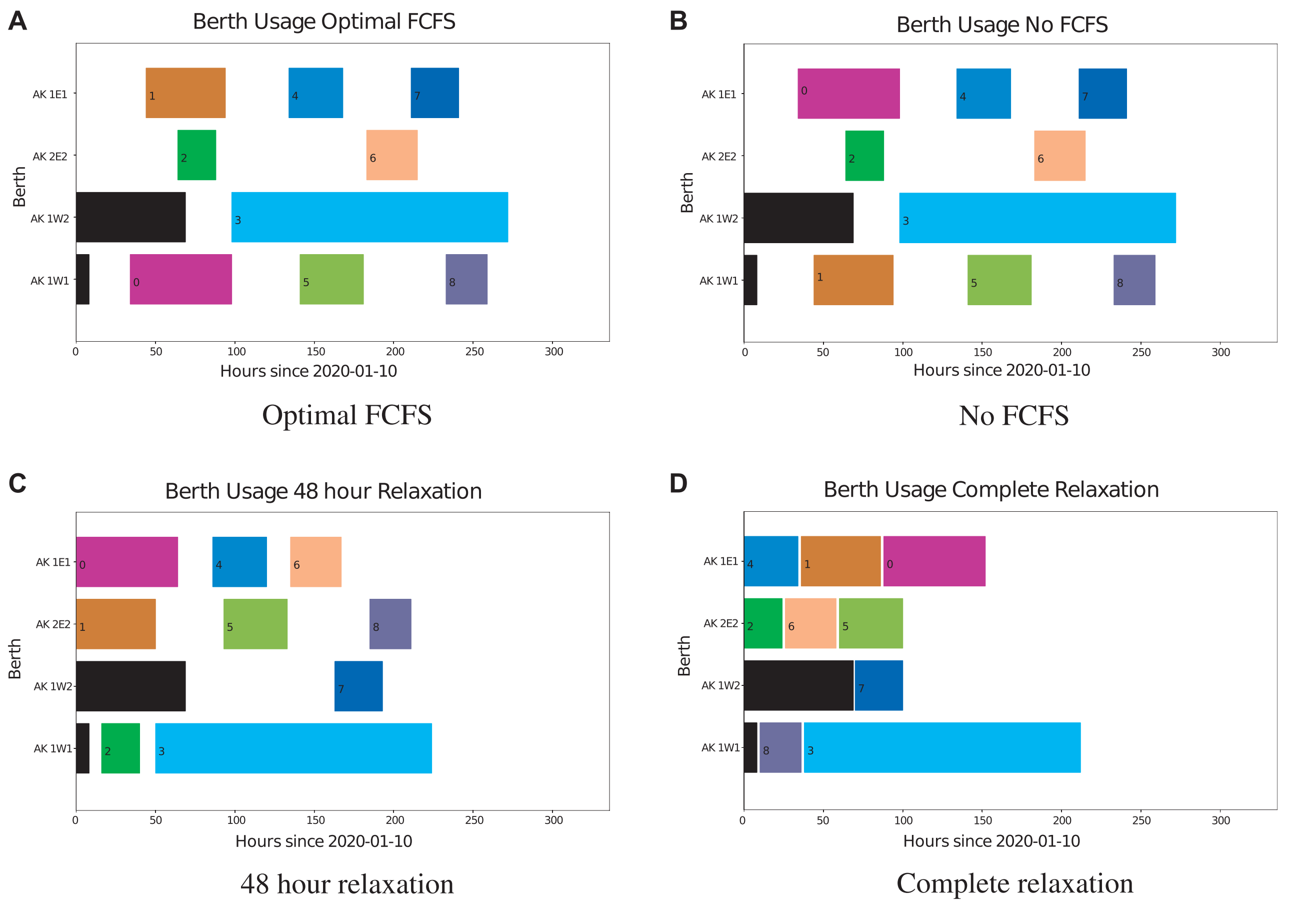


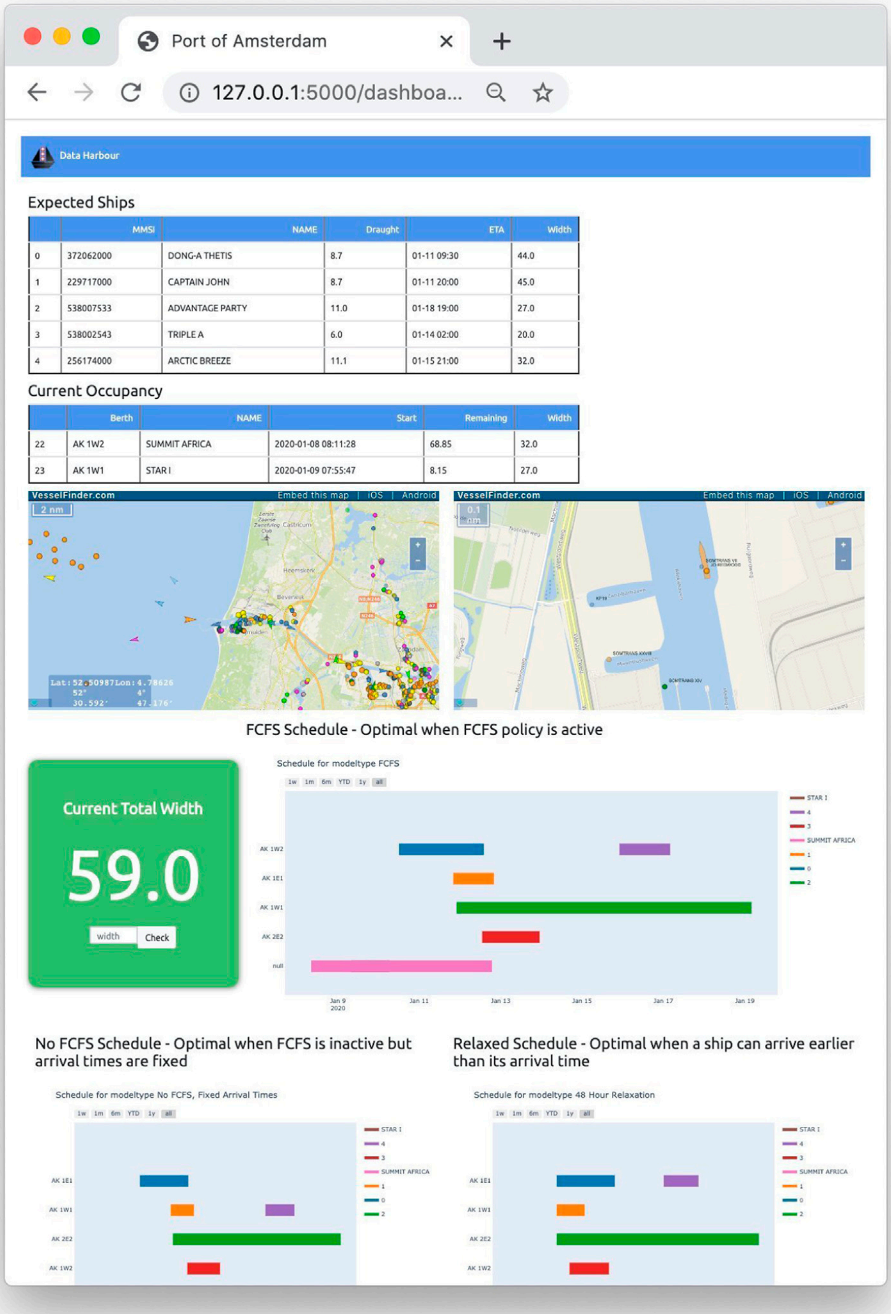
FIGURE 6 | Example schedules of the four different model types for regular sized ships for objective function $\sum C_j$. **(A)** Optimal FCFS, **(B)** No FCFS, **(C)** 48 h relaxation, **(D)** Complete relaxation.

TABLE 2 | Percentual Performance of Model Types (lower is better).

Objective **Function**	**% Of ships enlarged**	**Historical AIS**	**Optimal FCFS**	**No FCFS**	**48 h relaxation**	**Complete relaxation**
$\sum C_j$	0	100%	79%	79%	64%	39%
	25	—	100	99.8	83.7	58.9
	50	—	100	97.0	84.5	65.0
	75	—	100	98.0	87.6	70.3
C_{max}	0	100%	88%	87%	75%	42%
	25	—	100	100	88.4	55.7
	50	—	100	100	88.0	62.5
	75	—	100	100	90.5	66.1

expandable. The dashboard provides information on incoming ships and the current situation using tables. This is further assisted by the use of embedded real-time maps from Vessel Finder[3] that give a geographical perspective and live tracking for incoming ships and already docked vessels. Moreover, the dashboard contains a widget that displays the current width of the geometrically constrained berths in the terminal. This widget also functions as a calculator that will turn red when the width given as input + the current total width of the geometrically constrained berths exceeds the maximum width. This widget automatically resets after 3 s, ready for new input. Lastly, our model is used to provide different schedules based on the different model types (1–3) discussed in section Different Model Types. Consequently, this provides the harbour master with an overview and some

[3] https://www.vesselfinder.com/.



Expected Ships

	MMSI	NAME	Draught	ETA	Width
0	372062000	DONG-A THETIS	8.7	01-11 09:30	44.0
1	229717000	CAPTAIN JOHN	8.7	01-11 20:00	45.0
2	538007533	ADVANTAGE PARTY	11.0	01-18 19:00	27.0
3	538002543	TRIPLE A	6.0	01-14 02:00	20.0
4	256174000	ARCTIC BREEZE	11.1	01-15 21:00	32.0

Current Occupancy

	Berth	NAME	Start	Remaining	Width
22	AK 1W2	SUMMIT AFRICA	2020-01-08 08:11:28	68.85	32.0
23	AK 1W1	STAR I	2020-01-09 07:55:47	8.15	27.0

VesselFinder.com
Embed this map | iOS | Android
2 nm
VesselFinder.com
Embed this map | iOS | Android
0.1 nm

FCFS Schedule - Optimal when FCFS policy is active

Current Total Width

59.0

No FCFS Schedule - Optimal when FCFS is inactive but arrival times are fixed

Relaxed Schedule - Optimal when a ship can arrive earlier than its arrival time

FIGURE 7 | Preview of the dashboard (note that this runs locally).

recommendations that aim to help and streamline the process of managing terminal operations.

The three schedules that are displayed are all optimal schedules, but choosing the right schedule for the right situation depends on the set of rules that hold. If scheduling is only allowed under a FCFS policy, the optimal FCFS schedule is the best. If no FCFS policy is set and vessels cannot arrive earlier than their ETA, the No FCFS model is optimal. Lastly, if vessels can arrive earlier than their ETA, the 48 h relaxation model is optimal. The fourth model (complete relaxation) is excluded from the options, as this is the least realistic model.

It is important to note that the dashboard only imitates a real-time model and can thus be seen as a proof of concept. The dashboard provides all the technical front facing functionalities, however currently depends on historical data as a placeholder and therefore cannot be called real-time. In order to provide a real-time experience a significant amount of data engineering and integration work with existing port systems would still need to be carried out. Furthermore, in real time, the model would rely on the prediction of duration of stay of vessels and the ETA as communicated to the Port by the vessels.

6.2 Prediction Model: Duration of Stay

To create a real-time berth allocation schedule, an estimation of the duration of stay is needed for each vessel. While searching for the most appropriate machine-learning model it became evident that the size of the available data was too small to create a reliable estimation. If more data would be available, the Random Forest algorithm, which allows multi-categorical variables, would be a feasible option.

7 RECOMMENDATIONS

During the research it became evident that the problem was more complex than originally expected. In this section we highlight external factors, suggestions for further research, and ideas for extension of the graphical user interface for use in a decision support system.

7.1 External Factors

There are several external factors that would complicate the allocation model if taken into account. Sometimes, tugboats might not be available, making it impossible for vessels to follow the schedule. Also, weather can have a drastic impact on efficiency. Furthermore, the planning of a single terminal is highly dependent on the planning of the sea lock at IJmuiden, which involves also other terminals. Additionally, volatile oil prices can disrupt the berth allocation. When oil companies have multiple ships in the queue at the sea lock waiting to enter the port until a berth becomes available at a terminal, they can decide to switch the order of the ships around, as well as postpone entering the harbour altogether when oil prices are lower than desired. Some of these factors could in the future be addressed if more data is collected and incorporated in the allocation model. More data would also greatly improve the quality of the prediction model for the duration of stay.

7.2 Alternative Objective Functions

In addition to the two objective functions discussed in this paper, other objective functions were considered and could be investigated in the future. In particular, minimising the sum of the maximum completion times of each berth could be interesting to examine. Instead of only looking at the maximum completion time of a single berth, all berths would constitute to the final objective value. Ultimately, a combination of all three mentioned objective functions might generate the best schedules. However, this would require more computational power as well as add complexity to the model, as objective values would be harder to interpret.

7.3 Further Research

A further implementation of applying weather analysis is of great importance as the weather condition highly affects the speed of ships. Extreme weather like storms will extend the duration of stay of berthed vessels, nevertheless, following wind may speed up a ship's journey and therefore arrive earlier than the estimated time. Additionally, incorporating market analysis of relevant cargo types also has significant impact. It occurs frequently that oil tankers have arrived on time at the waiting area, yet they decide to wait there for longer until the oil prices have a desirable rate, which can disrupt the scheduling. Thus, taking the market factor into consideration could contribute to a higher quality of the simulation model.

Further research on improving the accuracy of the ETA could result in a more robust schedule. In addition, further research on predicting the duration of stay (accompanied with more data) is necessary to produce more accurate predictions. The more accurate the prediction of the ETA and duration of stay are, the more accurate an application of the model will be in real-time for facilitating operations.

7.4 Real-Time Application

The dashboard could be extended by several features, including the addition of a line graph that displays the total width over time in the Gantt charts that display the schedule. Furthermore, instead of a current width displaying widget, the "total allowed width" could be displayed, showing the harbour master how much space is still available.

8 CONCLUSION

This research has shown that AIS data is a very powerful source for spatial analysis. In this case it is first used to derive the berthing time of vessels staying at the port, after which a prediction model was made to predict how long the ships stay at a certain berth. Moreover, AIS data could especially be of use when tracking inland vessels, something that could be of great importance for improving efficiency in terminals.

Furthermore, it has become clear that the current first-come, first-served berth allocation strategy works, but can be improved. Especially when larger ships arrive at the port – something expected to happen when the new lock becomes operational –, a more efficient way of scheduling is necessary to make more efficient use of the terminal. The first-come, first-served model as constructed in this case can be useful when a first come, first serve policy still holds, but this case has also shown alternatives for when there is no FCFS policy, or when the arrival times of ships is relaxed.

When larger ships arrive at the port, the first-come, first-served allocation strategy will not be the most efficient berth allocation strategy anymore. Efficiency can be gained by abandoning the first-come, first-served policy, yet realistically most efficiency can be gained by also allowing ships to be scheduled before their estimated arrival time. Although it might seem unrealistic to let ships arrive before their estimated arrival times, this also highlights the importance of having a reliable ETA: the more reliable the ETA of a vessel is, the more reliable the allocation model works. Similarly, the importance of good prediction of duration of stay is essential to having a reliable real-time allocation model.

The main aim of this research was to investigate if there are other ways for berth allocation that optimise the total berth usage. Furthermore, a main goal was to create a model that supports the harbour master in taking berth allocation decisions. Therefore, the created dashboard shows the different schedules generated by the model in a way where the final decision of allocation is always made by the harbour master.

DATA AVAILABILITY STATEMENT

The datasets presented in this article are not readily available because the data that support the findings of this study are available from Centrum Wiskunde and Informatica (CWI) and Port of Amsterdam. Restrictions apply to the availability of these data, which were used under license for this study. Data are available from the authors upon reasonable request and with permission of CWI and Port of Amsterdam. Requests to access the datasets should be directed to TK, koch@cwi.nl

AUTHOR CONTRIBUTIONS

The authors confirm contribution to the paper as follows: study conception and design: JZ; data collection: TK, ED, and JZ; analysis of results: HE, JL; interpretation of results: HE, JL, JZ, TK, and ED; draft manuscript preparation: HE and JL. All authors reviewed the results and approved the final version of the manuscript.

ACKNOWLEDGMENTS

The authors would like to thank H-J. Oost and P. Valk from Port of Amsterdam for their contribution to the study conception and design, data collection, and the interpretation of results. The authors would also like to thank H. Tao, D. Pop, and U. Rimsa from the Vrije Universiteit Amsterdam for their contribution to the analysis of results, interpretation of results, and draft manuscript preparation.

REFERENCES

Bierwirth, C., and Meisel, F. (2015). A Follow-Up Survey of Berth Allocation and Quay crane Scheduling Problems in Container Terminals. *Eur. J. Oper. Res.* 244, 675–689. doi:10.1016/j.ejor.2014.12.030

Bierwirth, C., and Meisel, F. (2010). A Survey of Berth Allocation and Quay crane Scheduling Problems in Container Terminals. *Eur. J. Oper. Res.* 202, 615–627. doi:10.1016/j.ejor.2009.05.031

Brown, G. G., Lawphongpanich, S., and Thurman, K. P. (1994). Optimizing Ship Berthing. *Naval Res. Logistics* 41, 1–15. doi:10.1002/1520-6750(199402)41:1<1:aid-nav3220410102>3.0.co;2-l

Clements, D., Crawford, J., Joslin, D., Nemhauser, G., Puttlitz, M., and Savelsbergh, M. (1997). "Heuristic Optimization: A Hybrid Ai/or Approach," in Workshop on Industrial Constraint-Directed Scheduling.

Garey, M. R., and Johnson, D. S. (1979). *Computers and Intractability*, 174. Murray Hill, NJ: Freeman San Francisco.

Imai, A., Nagaiwa, K. I., and Tat, C. W. (1997). Efficient Planning of Berth Allocation for Container Terminals in Asia. *J. Adv. Transp.* 31, 75–94. doi:10.1002/atr.5670310107

Imai, A., Nishimura, E., and Papadimitriou, S. (2003). Berth Allocation with Service Priority. *Transport. Res. B Meth.* 37, 437–457. doi:10.1016/s0191-2615(02)00023-1

Imai, A., Nishimura, E., and Papadimitriou, S. (2001). The Dynamic Berth Allocation Problem for a Container Port. *Transport. Res. Part B Meth.* 35, 401–417. doi:10.1016/S0191-2615(99)00057-0

Kovac, N. (2017). Metaheuristic Approaches for the Berth Allocation Problem. *Yugoslav J. Operation* 27, 265–289. doi:10.2298/YJOR160518001K

Lim, A., Rodrigues, B., Xiao, F., and Zhu, Y. (2004). Crane Scheduling with Spatial Constraints. *Naval Res. Logistics* 51, 386–406. doi:10.1002/nav.10123

Lim, A. (1998). The Berth Planning Problem. *Operations Res. Lett.* 22, 105–110. doi:10.1016/S0167-6377(98)00010-8

Port of Amsterdam (2018). New Big Sealock IJmuiden. Available at: https://www.youtube.com/watch?v=CC-vFJlZIwk (Accessed March, 2020).

Smith, T. B., and Pyle, J. M. (2004). An Effective Algorithm for Project Scheduling with Arbitrary Temporal Constraints, in The Nineteenth National Conference on Artificial Intelligence, Topic Planning and Scheduling, San Jose, CA, July 25–29, 2004 4, 544–549.

Umang, N., Bierlaire, M., and Vacca, I. (2013). Exact and Heuristic Methods to Solve the Berth Allocation Problem in Bulk Ports. *Transport. Res. E Logist. Transport. Rev.* 54, 14–31. doi:10.1016/j.tre.2013.03.003

Xu, D., Li, C.-L., and Leung, J. Y.-T. (2012). Berth Allocation with Time-dependent Physical Limitations on Vessels. *Eur. J. Oper. Res.* 216, 47–56. doi:10.1016/j.ejor.2011.07.012

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2021 Eisen, Van der Lei, Zuidema, Koch and Dugundji. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.