Barge Planning and Routing with Uncertainty of Container Availability

Master Thesis

Christel Monrooij (385583)

Supervisor EUR:
T.A.B. Dollevoet
Second Assessor:
C.D. van Oosterom

Supervisors TNO:
S.A. Merriënboer & A. Sangers

August 27, 2018

Abstract

Transportation of containers in hinterland connections is preferably done by barge. This transport depends on a lot of parameters and variables. Barge planners usually try to find a solution in a continuously changing environment. This research tries to model the choices of barges planners in such a way that the model can be used in a Decision Support System (DSS) for planners. The research assumes a complete control of resources, such as barges. The demand is assumed to be stochastic, while the demand to resource allocation is also controlled. The problem is modelled as a Markov Decision Process(MDP), with notation of Rivera and Mes (2016) and solved with an Approximate Dynamic Programming (ADP) solution method as *policy iteration alogrithm with basic functions*. We assume that a state of the MDP can be described by some features, which can well estimate the cost of a certain state. The model is tested with a case study of a network in The Netherlands, with different initialisations and assumptions. Concluding, the model can be used to predict costs of certain actions and can be used to give an advice on certain choices.

ERASMUS UNIVERSITEIT ROTTERDAM
ERASMUS SCHOOL OF ECONOMICS

TNO

Contents

1 Introduction

Containers started a new era in logistics. They made the transportation of goods easier and the standardization of equipment resulted in large cost savings. Containers made the use of economies of scale easier, by using large deep sea vessels, such that the costs for transportation declined. Since an origin or a destination of goods is normally not in a sea port, transportation from and to the port is necessary. In The Netherlands, a large part of the hinterland transport is done by barge. According to the Port Vision of the Port of Rotterdam (2011), 40% is transported by barge, around 12% is transported by train, while most of the containers are transported by truck. In the Port Vision (PoR, 2011), it is stated that the amount of goods transported by barge in 2011 (2.2 million TEU) should be tripled in 2035, which implies that over 7 million TEU need to be transported by barge in 2035. This requires large changes in hinterland transportation.

Most inland terminal operators guarantee a certain service, or number of weekly departures from and to Rotterdam or Antwerp. These companies mostly employ a certain number of container barges, for picking-up and delivering containers. The barge planners make appointments for the barges, and decide which containers are assigned to which barge, whilst making sure containers depart and arrive on time. Most barge planners work with one or a couple of inland terminals, and deal with a lot of different terminals in the sea ports. Barge planners make use of large dashboards with a lot of information, and also experience and own insights are used to compose a barge planning. The planning is continuously adapted because of new available information.

Planning barges in a certain network is a complex problem, and depends on a lot of different varying parameters. Most barge planners encounter problems when dealing with the schedules of terminals. Making an appointment for a barge at a terminal can be hard, since most terminals have a full schedule and limited recourses, like cranes and quays.

Finding a feasible route for a barge, in which all terminals are visited on the agreed moments, for collecting and delivering containers, is a challenge. On an operational level, finding a feasible solution, in which containers can be transported by barge to their destinations on time, is far more important than finding an optimal solution. Barge planners solve these problems continuously, while making use of a planning horizon of approximately a day up to a week. The human factor of experience plays a key role here in finding a solution.

The scope of this research is to find a model, such that human planners can make better informed decisions. When the model has small computation times, it can be used in a Decision Support System (DSS). If the computation times are larger, we can try to come up with an approximate solution. The goal is to make the decision of the barge planner easier and more efficient. The question is how we can model the choices barge planners are faced with, and how can we solve this model such that it can be used as a DSS for planners.

In this research we will first discuss the related literature (Chapter 2). Chapter 3 contains a brief problem description. In Chapter 4 we will discuss the proposed model with notation and solution methods, and in Chapter 5 we will introduce a case study to test and validate the proposed model. Then in Chapter 6, the results of the case study will be presented, as well as some facts about the running times for the different configurations for this case study. Conclusions and suggestions for further research will be discussed in Chapter 7.

2 Literature overview

A lot of research has been done on the logistic problems related to the movement of containers in the port and in the hinterland connections.

Konings (2007) wrote a paper about the container movements and especially the improvements that can be made in the port of Rotterdam. The research of Konings (2007) has the aim to evaluate different scenarios for service operators, on their potential profits. These evaluations are done with a marginal cost/revenue analysis. He emphasizes that long waiting times for barges in the port are encountered more often than not. Reducing these waiting times can result in large costs and time savings. Konings (2007) concludes that the different terminals of the port of Rotterdam can be divided into the terminals at the Maasvlakte, the Botlek area and the Eem- and Waalhaven. This division is done, because traveling between these areas of the port consumes a lot of time. Konings (2007) also concludes that call sizes can differ per port cluster, and next to that, the average waiting time per terminal can differ across terminals.

Douma, Schutten, and Schuur (2009) describes the problem barge planners have to handle everyday. Reasons why activities between barges and terminals fail in cooperation can be classified in three categories.

1. *Autonomy.* Every terminal and barge operator wants to stay autonomous.

2. *No contractual relationships.* There is no contractual relationship between a barge and a terminal. Thus, forcing each other to deliver a certain service or charge each other for poor services is not possible.

3. *Limited information sharing.* Barge operators compete with each other, as do terminals, so sharing information could undermine their competitive position.

Other arguments Douma et al. (2009) makes for this complex problem are the different interests of actors. Douma et al. (2009) proposes a multi-agent based model, to optimize the barge rotation given the different interest of actors and the dynamic environment, where the agents in the model represent all the different terminals and barges. Nevertheless the main focus is on barge performance, since a good barge performance will result in a good terminal performance as well. The model of Douma et al. (2009) does only consider the different players and the quay planning, while for example the individual containers are not considered. The paper looks into the problem of modeling the whole system, even though it might be interesting to look at optimization for one barge (or a certain barge company), while considering the expected freight.

For the specific planning of a barge, we discuss two different papers. First the work of Kooiman, Phillipson, and Sangers (2016), and secondly we will give an overview of the work of Rivera and Mes (2016). These two papers give a good insight in how we can build a good model for the individual barge planning.

2.1 Inland Container Shipping as a Stochastic Assignment Problem

Kooiman et al. (2016) uses a decision model to assign containers to barges. One of the main goals of Kooiman et al. (2016) was to make a DSS, in which some *rules of thumb* are compared with an optimal solution. Several assumptions are made to solve the problem as a Stochastic Assignment Problem (SAP). Here, the following assumptions are made.

1. *Barges have a known fixed capacity.*
2. *Containers can only be unloaded at the port of destination or at the container terminal at the end of the round trip.*
3. *All containers have a known fixed due time and stochastic release times with a known normal distribution. The mean is different for each container, but the standard deviation is equal for all containers.*
4. *All containers are of equal type.*
5. *The travel schedule of the barges is fixed and known.*

The first assumption is quite standard and realistic, since in practice barges have a maximum capacity that does not change over time. However, at certain hinterland connections containers cannot be stacked too high due to bridges and water levels. This influences the capacities of barges. Besides, the capacity is different for each barge due to different types of barges. The second assumption makes sure that only transport from the sea port to the inland terminal and vice versa is allowed in the model. This is realistic in most cases, since barge transport between inland terminals is rarely used. However, in some cases transport between different port terminals is used, before containers are transported to the hinterland. A third assumption is used for the distribution of the release times of containers: a data set is created with some distributions. Since not all arrivals are known in advance, an assumption is made on the distribution of the arrivals of containers. Kooiman et al. (2016) use an hourly based arrival and due time, while in practice deadlines and other dates, except the terminal appointments, can be less precise. The fourth assumption is about the types of containers. In reality a lot of different types of containers are used, varying from 20 ft, 40 ft, high cube, 45 ft, or containers which need extra attention like a reefer container. But for a model, an assumption in which only one container type is used can be useful for limiting the calculation time. The last assumption is the most restrictive one. This assumption results in a fixed route with fixed departure and arrival times, where a barge can and will always be handled at a scheduled time. The schedule used by Kooiman et al. (2016) starts and ends at a container terminal in the sea port, with a fixed order for different inland terminals. However, in practice voyages of barges, scheduled by operators of inland terminals, are planned from an inland terminal with multiple stops at terminals in the port, back to the inland terminal. More flexibility in the routing of the barges might lead to a better solution, in which more containers are transported by barge. More flexibility in the routing might also lead to a model that will resemble more of the planners reality.

The model of Kooiman et al. (2016) is a good start, but the DSS can be tested with other assumptions, mostly regarding assumptions 3 and 5. Also other *rules of thumb* for barge planners can be constructed regarding more uncertainty about terminal availability and routing policies.

De Juncker, Huizing, del Vecchyo, Phillipson, and Sangers (2017) designed a framework for describing different problems in (synchromodal) transport. This notation can be used to show which elements of the model are fixed, stochastic, dynamic or not considered in the model. A division is made in the resources and the demand side. Also the level of known information and optimization together with the number of players in the model is described in the notation. This notation makes it easier to compare different models in (sychromodal) transportation problems.

The model of Kooiman et al. (2016) can be summarized in notation of De Juncker et al. (2017) by $\overline{R}, \xcancel{TH} | \overline{D}, [D2R], \widehat{DRD} | \text{social}(1)$. This means that we consider a global optimum with global information, with one player (social(1)). All resources are fixed ($\overline{R}$), except the terminal handling times, these are not considered ($\xcancel{TH}$). The demand is all fixed ($\overline{D}$), except the demand release date which is stochastic ($\widehat{DRD}$) and the model controls the demand-to-resource assignment ($[D2R]$).

2.2 Service and Transfer Selection for Freights in a Synchromodal Network

The work of Rivera and Mes (2016) is based on the problem of a planner for a synchromodal network. This model makes use of freight from certain origins and a different set of destinations. Between these places, we have a network of different transfer modes. For example everything can be done by truck, but some transfers between terminals are with transportation by barge, train or even plane. The planners need to choose how all freight will be transported. Although the planner might know the complete route for one freight container, he will only implement the first steps for transport. When new information becomes known, like new freights, a different route can be preferred over the initial route. This is modeled as a Markov Decision Process (MDP) where in each state different decisions can be made, while keeping in mind that new freight can occur, following a certain distribution. So at moment t we act on the different scenarios that can occur, with different probability. The planner can choose at t for any freight to (i) transport the freight to its final destination (by truck), (ii) transport the container to an intermediate terminal or (iii) postpone the transport of a freight.

In Rivera and Mes (2016) the network is a directed graph, and the state space can be denoted as $S_t = [F_{i,d,r,k}]_t$. $[F_{i,d,r,k}]$ is a list, where i is the current location, d the destination, r the release day, k the number of days until the due date and t the current time. F stands for the number of freights with the same characteristics. This can limit the options since containers for a destination with the same time window are bundled. At the same time the number of total options of the state space is huge, since this is the combination of all the different container flows and their amounts. The set of actions which can be taken in state S_t is a combination of which freight is transported over which arcs (to an inland terminal, or by truck to a customer), and which freight is kept at a certain terminal.

The model of Rivera and Mes (2016) can be summarized in the notation (De Juncker et al., 2017) as, $\overline{R}|\widehat{D}, \overline{DP}, [D2R]|\text{social}(1)$. This means all resource elements are fixed, all demand elements are stochastic, except the demand penalty ($\overline{DP}$) which is fixed and the global demand to resource ($[D2R]$) allocation is optimized, given global information, with only one player in the model (social(1)).

3 Problem description

As we saw in the first chapters different problems arise when looking into the barge planning. The problem for a barge planner is that for all freight a choice needs to be made at a moment in time, and a planning should be made for all barges. This planning should be in such a way that all freight can be transported to its destination before the due date.

Two things are determined in each decision moment by the barge planner, namely at which terminals new appointments are made and which containers are transported to their destination with a certain barge. Assigning containers to a barge is a demand-to-resource decision that has been studied in literature (Kooiman et al., 2016; Rivera & Mes, 2016).

After making these decisions, new information can become known to the planner: for example new freight can be released for transportation. So in the next decision moment a planner has additional information over all the freight that needs to be transported. The releasing moments of this new freight can be estimated with a certain distribution. This means that the demand is stochastic. The decision of making appointments determines the schedule for the barge. So with the decisions of appointments the origin, destination and the departure times of the barges are determined, which makes the resource origin, destination and departure time controllable.

When an appointment at a terminal is made, the exact number of movements, with the exact container numbers still can change, but after the so called *pre-registration*, this is fixed at a day. When this information is not available at the right time, in the right condition at the terminal, a container cannot be delivered or picked up. When the pre-registration is confirmed, changes cannot be made anymore. In practice terminals can differ in the due time for an appointment and the due time of the pre-registration.

3.1 Time horizon assumptions

In the problem of the barge planning different time windows are used. We have a planning horizon of several days up to a week in most cases. This is the schedule for each barge which contains appointments at different terminals. These appointments are made in a strict terminal schedule with a high precision. An (half) hourly schedule is considered to be a good representation of the reality.

When looking at the time intervals of the arrival and due dates of freight, we see different views in literature. Kooiman et al. (2016) uses an hourly interval, for all dates concerning containers, while Rivera and Mes (2016) makes use of a daily interval for arriving and due dates of containers. In this study we will assume the arrival and due dates are on a daily basis, whereas the decision moments and appointments are hourly based. Rivera and Mes (2016) emphasize that the length of the intervals can be chosen at any length, but for simplicity

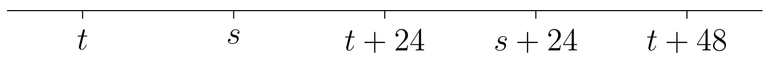


Figure 1: A figure which represents the time horizon. Where t are the moments every 24 hours when new information becomes known, and s, between t and $t + 24$, where the planner makes new decisions for $s + 24$.

and calculation issues they choose a daily basis. We choose for an hourly decision moment, with daily new information about new container flows.

The last time parameter is the horizon on which implications of decisions can be compared. In theory this horizon is infinite, but due calculation issues we use a horizon with the same order of magnitude as the planning horizon, 48 hours in this case. One can argue that the implications of a decision will mainly happen in the next 48 hours.

This all together leads to a rolling time horizon as in Figure 1. The moment t (and $t + 24$,$t + 48$) stands for the moment where new information becomes known. These are the moments between days, such that the daily basis of arrival and due dates for containers can be managed correctly. s ranges between t and $t + 24$. These are the hourly decision moments, where the new appointments are made for the barges. At decision moment s, a new appointment can only be made at moment $s + 24$ due the pre-registration time. The planning horizon at time t goes up to time $t + 48$. When s is equal to $t + 24$ the horizon changes, to a new situation where $t + 24$ is now equal to t, this makes it rolling horizon. The rolling horizon aspect is added to construct a cyclic formulation. This way only the necassary information is kept, to reduce memory and calculation times.

3.2 Possible objectives

If the model should resemble the reality of the barge planner we should use a realistic objective, that is close to the objective of the barge planners. Barge planners want a high customer satisfaction, while still making a feasible planning, for customer, barge and terminal operators.

One of the most used objectives in optimization studies is *minimizing the overall costs*. While the profit of a company drives a lot of decisions, the question is whether a barge planner makes decisions based on the costs of the actions. A barge planner might not know all the implications of his choices regarding the costs.

One of the main priorities of a barge planner is to have a good customer service, which is obtained if all containers are delivered on time at their destination. The objective in Kooiman et al. (2016) is *maximizing the expected number of containers transported by barge*, which will result in a solution where most containers are transported by barge and are delivered on time. This approaches the way barge planners act. Given a certain customer service level, with this objective we optimize the usage of the barge(s). This objective is similar when a penalty for transporting a container by truck is added, which makes choosing barge over truck more attractive. The only modalities considered in this paper are truck and barge.

Since not always all containers can be delivered we would like to minimize the number of containers that cannot be delivered on time. A different approach could be to *maximize the ratio of the containers delivered by barge given the total number of containers*. This way we still maximize the number of containers delivered while we keep in mind the total number of containers in the planning system.

In all objectives listed above, the customer service is taken as a restriction. In some cases, a penalty could also be used for containers that are delivered later than the due date, such that containers are still delivered by barge, but with (some) delay.

The *maximization of the ration of containers delivered by barge given the total number of containers* will be the objective for the remainder of this research. Since most solution methods speak of a minimisation of the costs, in the remainder we will multiply the objective by minus one. This way, we can speak about minimizing the costs of the problem.

3.3 Problem classification

This all together would lead to a problem formulation, $\overline{R}, [RO], [RD], [RDT], \cancel{RP} | \widehat{D}, [D2R] |$ social(1) (De Juncker et al., 2017). This means that in terms of resources we omit the resource price ($\cancel{RP}$), we control the origin ($[RO]$) and destination ($[RD]$) of the resource. The resource departure time ($[RDT]$) is also controllable, while the rest of the resource elements is fixed ($\overline{R}$). All demand related input is stochastic ($\widehat{D}$), except we control the demand to resource allocation ($[D2R]$), in again a global setting with global information, with only one player, one acting barge company, as indicated by social(1).

4 Model description and solution method

In this chapter several parts of the modeling are discussed. First the assumptions for a model are discussed. In Section 4.2 the notation of the model is described and in last section of this chapter a solution method of the proposed model is described.

4.1 Assumptions for a Markov Decision Process

To model the problem of the barge planners we will use a Markov Decision Process (MDP). This enables us to model the different situations a barge planner will encounter, together with the different choices a barge planner can make in each situation. By keeping track of the costs of each chosen action, we can optimize the actions barge planners take.

It is not the first time that MDP is used in logistics problems. For example Mulder and Dekker (2012) used MDP to model the uncertainty of delays that deep sea vessels can encounter during their trips, and which decisions can lead to an optimal solution, taking into account the fuel consumption, and the costs incurred for arriving late at different ports. In the research of Rivera and Mes (2016) MDP was used for the planning of goods over different modes.

In Tijms (2003) MDP is described as a model with a set of possible states I, which is assumed to be finite. For each state i a set of actions is given by $A(i)$, which is also assumed to be finite. The process can be described as a *discrete-time Markov model* when the Markovian property is satisfied. So in state i, when action a is chosen, regardless of the history, a cost of $c_i(a)$ is incurred and the system will be in state j, with probability $p_{ij}(a)$, where the sum of $p_{ij}(a)$ for all $j \in I$ will be equal to 1, for all states $i \in I$ and $a \in A(i)$. In most MDP a *long-run average cost (reward)* per time unit is the used optimality criterion.

We can formulate a stationary policy or a policy R, such that in state i we always perform action a, which leads to (stationary) policy R_i, where we have $P\{X_{n+1} = j | X_n = i\} = p_{ij}(R_i)$. So with policy R_i, the process is a discrete-time Markov Chain with one-step transition probabilities $p_{ij}(R_i)$.

One of the problems that needs to be addressed is that a fixed schedule is not very applicable in practice. In Mulder and Dekker (2012) MDP was combined with mixed integer programming (MIP) to construct the buffer location in the different positions, such that an optimal use could be made of the recovery possibilities. We can combine the MDP with a construction of the (underlying) network for the barges, such that the best routes are found considering the expected amount of freight. However, we would then still construct one static set of routes for barges, while the goal is to construct a dynamic model which selects the routes for the barges based on expected freight.

The question is how can we incorporate the design of the route selection into the actions that can be taken in a state. We can divide a state into three different part. First the set of containers, with all the flows between the nodes (terminals and barges). This set shows the demand for transportation, where the containers are and where they should be transported to. The second part is the scheduling part, this is the schedule for the different barges: the location of the appointments and how much time a barge has between two consecutive appointments. Finally, we have a parameter t to indicate the current time of the day in a certain state. This is also necessary to read the schedule of the barges. This all together leads to the following description of a state S (the exact notation will be given in section 4.2):

$$\begin{aligned} F &= \text{A set containing the different container flows} \\ P &= \text{A set containing the schedule for the different barges} \\ t &= \text{A parameter which states the time of the day, } t \in [0, 23] \\ S &= F \times P \times t \\ K(s) &= \text{Actions that can be performed in state } s \end{aligned} \tag{1}$$

The current demand of freight is represented as F. The routes for the different barges can be constructed from the schedules (P), with the current time of day t and the schedule P, the current location of each of the barges can be shown. A state is defined by these three different objects. $K(s)$ is a combination of containers that can be picked up and the times in the schedules at which this can be done. Hence, an action consists of two different parts. First we determine which containers should be picked up if a barge is at a certain terminal. This can differ for all available containers. It could be that there are containers with a shorter due date. Another option is that there are containers that have to be delivered at a next terminal where we already have an appointment. The second part of an action consists of making new appointments. For each barge, we have the option of not making an appointment, or making an appointment that can be met given the minimum sailing times between terminals.

4.2 Notation

Most of the notation of Rivera and Mes (2016) is used. First a directed graph is used, $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of available nodes and $\mathcal{A}$ is a set of the available arcs. We do not consider the subscript of the time t. This enables us to make use of a cyclic formulation, where we can return to certain states. $\mathcal{N}$ is the set of all terminals in the model, and a dummy terminal for each barge $b \in \mathcal{B}$ in the set of nodes.

$\mathcal{A}$ can be split into two different sets of arcs. First $\mathcal{A}^D$ are all connections between the origins and the destination that can be performed by a truck. A high cost is associated with using a truck and therefore transport by barge is preferred. $\mathcal{A}^B$ are the arcs that can be traversed by barge. A way to reduce the number of considered connections between origins and destinations in the MDP is to neglect all the arcs in $\mathcal{A}^D$, so when a freight is transported by truck, it directly disappears out of the model, at a certain cost. We assume that transportation by truck is always faster, and can be done immediately and directly to its final destination. We also assume that transporting via truck is more expensive than transporting by barge.

$\mathcal{N}^T$ can be split into the nodes in the port $\mathcal{N}^P$ and the inland terminals $\mathcal{N}^I$, where $\mathcal{N}^P$, $\mathcal{N}^I$, make $\mathcal{N}^T$. $\mathcal{N}^T$ and the barges ($\mathcal{B}$, the barges can be seen as a dummy terminal) mutual exclusive make $\mathcal{N}$.

$\mathcal{R} = \{0, \ldots, r^{max}\}$ which represents the days until the container is available at place $i \in \mathcal{N}$. If $r \in \mathcal{R}$ is equal to zero, this means the container is available at node i. $\mathcal{K} = \{0, \ldots, k^{max}\}$ is the number of days until the due date, where we start counting down when $r = 0$. So the final due date at node $d \in \mathcal{N}^T$ is $r + k$, $r \in \mathcal{R}, k \in \mathcal{K}$. As initialisation for the parameters, r^{max} and k^{max}, we advice to take the maximum planning horizon in days, for example 7 days.

A state can be described as follows:

$$S = [F_{i,d,r,k}]_{\forall i \in \mathcal{N}, d \in \mathcal{N}^T \setminus \{i\}, r \in \mathcal{R}, k \in \mathcal{K}} \times [P_{b,t^*}]_{\forall b \in \mathcal{B}, t^* \in T^*} \times [t] \tag{2}$$

$F_{i,d,r,k}$, is the number of freights with the same current terminal i, destination d, the release day r, and number of days until the due date k. This limits the number of options to consider since containers with the same destination are bundled. We only have freight from inland terminals going to port terminals and visa versa. For the barges we accept all the terminals, $\mathcal{N}^{\mathcal{T}}$, as a destination of the freight. When freight is on a barge a container should have been released, $r = 0$.

The schedule of the barges is represented in P_{b,t^*}. P_{b,t^*}, equals i if barge b has an appointment at terminal i at time t^*, where $t^* \in T^* = \{1, 2, \ldots, T^{max}\}$. With $T^{max} = 48$, we plan always 24 hours ahead, since $1 \leq t \leq 24$, where t^* is part of the planning, and t is to track the current time. Because of the properties of an MDP we consider the probabilities and the choices that the next states will bring. This results in a solution where we consider all different options in the future with a certain discount rate for future costs.

Q_i is the capacity of barge i. So $\sum\limits_{\forall d,r,k} F_{i,d,r,k} \leq Q_i$ for each $i \in \mathcal{B}$. The minimum sailing times over all arcs are known and denoted by $L_{i,j}$. It is measured in the same units as t^*. The real scheduled sailing times are calculated from the schedule $[P_{b,t^*}]$.

Each action which can be done in a state consist of two parts. First the assignment of the containers and secondly the scheduling of new appointments for the barges.

$$A(S) \quad = \quad [DF_{i,d,r,k}]_{\forall i \in \mathcal{N}, d \in \mathcal{N}^T \setminus \{i\}, r \in \mathcal{R}, k \in \mathcal{K}} \times [AP_b]_{\forall b \in \mathcal{B}} \tag{3}$$

So each action $A(S)$ is specific to the state S and will lead to a next state at time $t = t + 1$. $[DF_{i,d,r,k}]$ is a vector with the differences in container allocation on all the flows. $[T_{i,d,r,k}]$ shows the number of containers that need to be trucked, and can be seen as a part of $[DF_{i,d,r,k}]$. Several constraints are needed for the movements of the containers formulated in equations (4).

When containers from terminal A to terminal B are placed on barge C, these transitions are given in $[DF_{A,B,r,k}]$ and $[DF_{C,B,r,k}]$. These variables can take both positive and negative values, but the total number of containers must be equal to zero (eq. (4a), (4f) and (4g)). The number of containers that are transported via a barge or truck cannot be larger than the total amount of available freight of that flow (eq. (4b)). Containers can only be transfered to the barge if the barge is at the terminal (eq. (4d)). If the transportation times by barge are too long, it is necessary to transport the freight by truck (eq. (4c)) assuming r and k are daily and $L_{i,d}$ hourly. Finally the capacity of the barge at time t cannot be exceeded (eq. (4e)). Actions in a state should satisfy all equations and be physically feasible as well.

$$\sum_{i,d,r,k} (DF_{i,d,r,k} + T_{i,d,r,k}) = 0 \tag{4a}$$

$$DF_{i,d,r,k} + T_{i,d,r,k} \leq F_{i,d,r,k} \qquad \forall i, d, r, k \tag{4b}$$

$$T_{i,d,r,k} = F_{i,d,r,k} \qquad \forall (r + k) * 24 \leq L_{i,d} \tag{4c}$$

$$DF_{i,d,r,k} = 0 \qquad \forall d \in \mathcal{B} : [P_{d,t*}] \neq i, r = 0 \tag{4d}$$

$$\sum_{d,r,k} (F_{i,d,r,k} + DF_{i,d,r,k}) \leq Q_i \qquad \forall i \in \mathcal{B} \tag{4e}$$

$$T_{i,d,r,k} \in \mathbb{N} \qquad \forall i, d, r, k \tag{4f}$$

$$DF_{i,d,r,k} \in \mathbb{Z} \qquad \forall i, d, r, k \tag{4g}$$

$[AP_b]$ contains an appointment for each of the barges over the *pre-registration* time, which is in this model chosen at 24 hours for all terminals. This appointment can be an appointment at a terminal or can be equal to zero which would mean the barges stays at the last visited terminal.

From the schedule $[P_{b,t^*}]$ we can form the parameter $D^1_{b,t^*} = (i, h)$, which shows the last terminal i that was visited by barge b, at time t^*, h hours ago. A new appointment at terminal j can only be made if $L_{i,j} \leq h$, since the time between the appointments must be long enough to travel between the terminals.

Just as in Rivera and Mes (2016), after making a decision but before entering the next state, new freights become known. Since the release date and the delivery windows are both in days, the new information becomes known only once a day. We choose to let new information become known after $t = 23$, from this t we move again to $t = 0$. This is at the same moment when all indices of the freight flows $F_{i,d,r,k}$ and schedules P_{b,t^*} are updated. The new freights can be formulated as W_t.

$$W_t = [\tilde{F}_{i,d,r,k}]_{\forall i \in \mathcal{N}^T, d \in \mathcal{N}^T \setminus \{i\}, r \in \mathcal{R}, k \in \mathcal{K}} \quad (5)$$

When we use all these notations together we can formulate the process as an MDP.

4.3 Solution method

An MDP can be solved with different methods. This all depends on the amount of different states and combinations of actions. Different ways of solving an MDP are used in literature. The Bellman equations are the most theoretical ones, but in practice these are limited, due to lengthy calculations. Other solution methods that are commonly used are an LP or IP formulation, where the decision variables stand for the percentual time spent in a state, and choosing a certain action. Another way is simulation of the MDP, while testing a decision policy. In Rivera and Mes (2016) a different approach is used. They used an Approximate Dynamic Programming (ADP) solution method because of the large amount of options in the state space.

These four are all methods that can be used to solve an MDP. Preferably an exact solution method should be tried first. However, when computation times increase, we can make use of different simulation methods, or other heuristics such as ADP. Since the number of states will increase enormously, when more container flows are introduced, we cannot iterate over all different states. This makes the problem more difficult and especially for large instances results in large problems.

With this model formulation instances will soon be too large. If we have 3 terminals and 3 barges, this results in 15 different arcs of (i, d). With $|\mathcal{R}| = 7$ and $|\mathcal{K}| = 7$. This results in $15 \times 7 \times 7 = 735$ different container flows. If all those container flows can have different values, this number will increase even more. The number of actions in each state is a permutation of the different parameters of the state, due the possibility of moving each individual container. When including the possibilities for new appointments, the amount of options increases even more and subsequently the amount of different states is even larger. Especially if we consider that our state space also consists of a schedule, with 3 barges by 48 hours, we have a lot of different options.

In the book of Powell (2007) a lot of different approaches for solving an MDP with Approximate Dynamic Programming (ADP) are discussed. ADP is a way to cope with the large instances of an MDP which cannot be solved exactly. Powell (2007) discusses different methods. We will look into a method where iterating over the different states is not necessary, since the number of states can and will explode with larger instances. A suggestion is a *policy iteration algorithm with basic functions*. Each state can be described with some basic features, like the amount of containers, or the capacity that is used. Each state can be described in a set of features, but a set of features does not describe a specific state. The idea is that these features indicate whether a state is expensive or not. The costs are estimated with a function based on the features of a state. We define an approximation function to estimate the real cost of a state:

$$\overline{V}(S|\theta) = \sum_f \theta_f * \phi_f(S) \quad (6)$$

Here, the estimated real cost ($\overline{V}(S|\theta)$) are an estimation of the objective value of for this state. With $\phi_f(S)$, we get the value of feature f in state S, while θ_f is the weight assigned to feature f to estimate the cost of state S. In this case the estimation of eq. (6) is a linear weight sum of the features.

The policy iteration part of the algorithm is that we iterate with different drawings of a random state to update the values of the basic functions, such that estimations become more accurate.

All together this would lead to an algorithm shown in Algorithm 1. First we initialize the algorithm by determining the basic function, to get the features that describe a state. (Line 2) Secondly, we fix the lookahead parameter T at $T^{max} = 48$ and the inner iteration counter is fixed at 10. We initialize the values for θ_f^0 at zero for all f, other options for initialisation of the weights are investigated in the case study. Line 6 contains the initialization of matrix B, this is an identity matrix with on the diagonal some small values. These can be chosen between 10^{-3} or 10^{-6}, this is the step value at which the weights of the estimation are changed.

Then we start with the iteration where we iterate N times, other termination criterion can be used as well. We also count the inner iterations, cause every M-th iteration is different. In every iteration, new estimations are made for the weights, θ_f^n, but we only update the policy function $\overline{V}_p(S|\theta^p)$ every M-th iteration. Therefore, M is called the inner iteration counter. If we would update this estimation every iteration, the new estimation of the weights together with the new observed realized cost, could lead to an over-fit or large fluctuations in the weights (Powell, 2007).

We choose a sample path ω^n such that we can step from one state to another state, on a certain path for the new generated information (W_t). We start in a random state (S_0^n) and generate T ($T = 48$) steps ahead. The state can be drawn randomly, or selected from a previous iteration. We choose an action which is expected to be the cheapest (line 14). After we chose an action, we determine the next state with function $S^M(S_t^n, a_t^n, W_{t+1}(\omega^n))$, where S_t^n is the current state, a_t^n the chosen action, and $W_{t+1}(\omega^m)$ the new random generated information about incoming containers. After we have generated these T steps ahead, we want to calculate what the theoretical real costs, $\hat{v}_0^n$, are of being in state S_0^n. This is done by calculating the sum of the realized costs of all the chosen actions and states after T steps (line 18-20), with a certain discount factor α for future costs.

With $\hat{v}^n$, the realized cost and the estimated costs ($\bar{V}_p$), the estimation error (η^n) can be calculated (line 23) and in the end the new weights θ^n. This is all according to the formulation for a policy iteration algorithm with basic function as can be found in Powell (2007).

4.3.1 List of features

For any model a set of features has to be made. This set of features should be representative for the cost of a state, such that we can formulate a cost function to estimate the cost of a given state. We choose to have features that can be calculated easily when a state is given. For this model several features are used and explained in Table 1. This list of features can be extended depending on the influence of other features on the expected costs. The features which have an influence on the expected cost can differ per case or model and can thus be

Algorithm 1 Policy iteration algorithm with basic function

Initialization
Fix the basic functions $\phi_f(S)$
Set lookahead parameter T, inner iteration counter M and stopping criteria N
Initialize θ_f^0 for all f which leads to $\overline{V}_p(S|\theta^p)$ with $p = 0$
Set $n = 1$ and $m = 1$
$B^0 = \epsilon I^n$

while $n \leq N$ **do**
  Get a random state S_0^n
  Choose a sample path ω^n
  $t = 1$

  while $t \leq T$ **do**
    $a_t^n = \min_{a_t \in A_t^n}\{C(S_t^n, a_t) + \alpha \overline{V}_p(S^M(S_t^n, a_t)|\theta^p)\}$
    $S_{t+1}^n = S^M(S_t^n, a_t^n, W_{t+1}(\omega^n))$
    $t = t + 1$
  Initialize $\hat{v}_{T+1}^n = 0$ and $t = T$
  while $t \geq 0$ **do**
    $\hat{v}_t^n = C(S_t^n, a_t) + \alpha \hat{v}_{t+1}^n$
    $t = t - 1$

  Update of θ_f^n
  $\hat{\eta}^n = \overline{V}_p(S_0^n|\theta^p) - \hat{v}_0^n$
  $\gamma^n = 1 + (\phi^n)^T B^{n-1} \phi^n$ where $\phi^n = \phi_f(S_0^n)$
  $H^n = \frac{1}{\gamma^n} B^{n-1}$
  $\theta^n = \theta^{n-1} - H^n \phi^n \hat{\eta}^n$
  $B^n = B^{n-1} - \frac{1}{\gamma^n}(B^{n-1}\phi^n(\phi^n)^T B^{n-1})$

  Update of policy in M-th iteration
  if $m = M$ **then**
    $p = p + 1$
    $\overline{V}_p(S_t|\theta^p)$ with $\theta^p = \theta^n$
    $m = 0$

  $m = m + 1$
  $n = n + 1$

return θ^N

different in other cases. Removing features if the influence on the expected costs is zero can be an option as well. This way a good model can be built which is representative for reality, and effective for the Decision Support System of the barge planners.

Table 1: A list of the proposed model features

	Name	Explanation
1	*onBarge*	Number of containers on a barge
2	*onBargeWithAppointment*	Percentage of containers on a barge with an appointment before their due date
3	*usedCap*	Percentage of currently used capacity of the barges
4	*ratioInPort*	Percentage of barges with the next appointment in the port
5	*totalContainers*	Total number of containers in the system
6	*toHinterland*	Percentage of containers going to hinterland (import)
7	*toPort*	Percentage of containers going to the port (export)
8	*notAvailable*	Not yet available containers as a percentage of the total number of containers
9	*shortDueDate*	Available containers with a (short) due date as a percentage of the total number of containers *This feature can be added for multiple times until a certain due date*
10	*constant*	A constant of 1 added to the features

4.4 Evaluation

The model can be evaluated in different ways. The algoritme returns the weights, to use these weights they need to be converged. To check whether the weights converged, a plot of the development of the weights (θ^n) throughout the iterations can be examined. Secondly, if the weights have converged, the prediction power of the model can be evaluated. In Algorithm 1, line 23, the difference between the estimated costs and the theoretical or real costs are deteremined. This error is used to adjust the model in each step. We can evaluate the final model, on how large the error is. This can be done in two different ways, first with a scatter plot of the estimated cost ($\overline{V}_p(S_0^n|\theta^p)$) versus the theoretical real cost ($\hat{v}_0^n$). If the plot is on the line $x = y$, the prediction is equal to the real costs. The errors can also be evaluated in an error table with the Mean Squared Error (MSE) and Mean Absolute Error (MAE).

5 Case Studies

We designed a testcase from a dataset provided by a container transport company in the Netherlands. This transport company has several inland terminals, and distributes containers from and to terminals in the Port of Rotterdam (and Antwerp). The most visited (inland and port) terminals are shown in Figure 2. These terminals are also shown in Table 2, where we see the estimated sailing times between the different terminals. These are an estimated guess based on the data. This container transport company normally sails this network with three different barges. In this testcase the capacities are 80, 100 and 120 TUE. Since no difference is made in the size of containers, we assume that each container counts for one TUE.

Figure 2: A network in The Netherlands, with 3 inland terminals (red), and 4 port terminals (black).

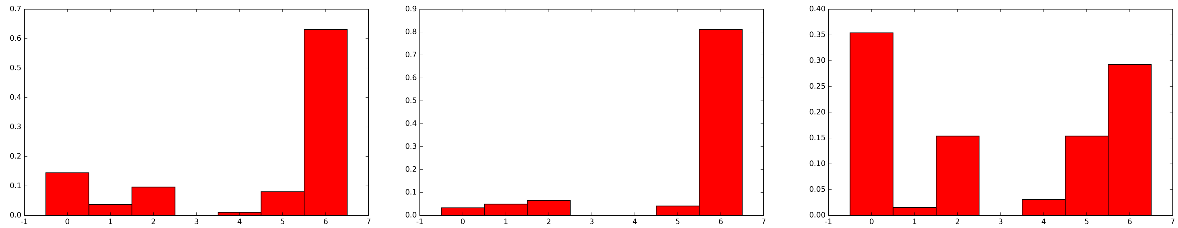
Table 2: Sailing times between different terminals in hours.

	CTN	WIT	ROE	APM2DEL	ECTEMX	ECTDDE	UWTERDAM	INLAND
CTN	-	2	5	16	15	14	13	TRUE
WIT	2	-	3	17	16	15	14	TRUE
ROE	5	3	-	20	19	18	17	TRUE
APM2DEL	16	17	20	-	1	2	3	FALSE
ECTEMX	15	16	19	1	-	2	3	FALSE
ECTDDE	14	15	18	2	2	-	2	FALSE
UWTERDAM	13	14	17	3	3	2	-	FALSE

We now estimate the expectations (or the distributions) of the container flows. A container flow exist out of the origin, destination, release time, due date and the amount of containers. We assume that container flows only exist between terminals from the port to the hinterland or from the hinterland to the port. We determine a certain probability at which a certain number of containers will come into the system the next day.

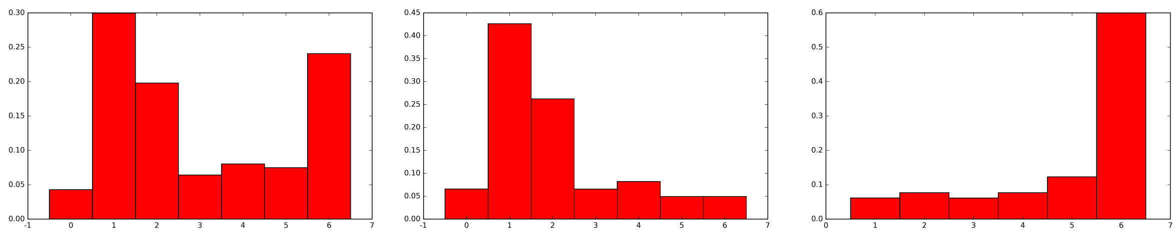
The container data from this company in The Netherlands can give an insight in the container flows. The test set of data exists of a total of 187 container flows. These container flows are split into import (from port to terminal) and export flows, respectively 65 and 122. For more extended analysis these flows could be split on all different origins and destinations, but with a limited data source, this is a good first estimation.

In Figure 3 we see histograms regarding the length the release times of the container flows. If the container flows are larger than 7 days, these are rounded down to 7 days, due to the calculating limitations of the model. This way, we consider the containers which are



(a) All containers (b) Export containers (c) Import containers

Figure 3: Histograms of the realized release times (r). For export, import and in total.



(a) All containers (b) Export containers (c) Import containers

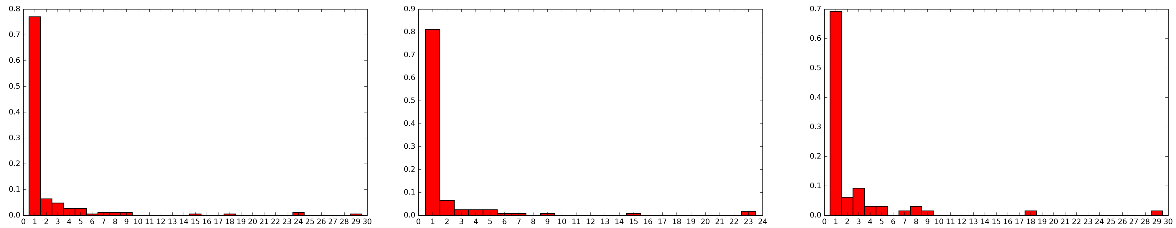
Figure 4: Histograms of the realized delivery windows (k). For export, import and in total.

released within 7 days. We see that most of the containers have a release time of 7 days or higher (Figure 3a), but if we compare the export (Fig. 3b) with the import (Fig. 3c), we see that for import containers the release times are more divided over the different days than for the export containers. This can be explained by the fact that the release times in the port might be more difficult to forecast due to the large container vessels.

In Figure 4, similar data is shown as in Figure 3, but now for the length of the delivery window, the time between the release date and the due date. Again, windows longer than 7 days, are rounded down to 7 days. We see high frequencies for two and three days (Fig. 4a), this is mostly due to the small delivery windows for export containers (Fig. 4b). For the import containers the delivery windows are larger, Figure 4c. Again we see a different pattern between the export and import containers.

For the amount of containers with similar characteristics, we can plot similar histograms (Fig. 5). As can be seen, 77% of the container flows have only a batch amount of one. From the histograms we can conclude as well that some container flows are not used. However, there do exist container flows with multiple containers per flow. Between the export (Fig. 5b) and the import (Fig. 5c) no clear difference seems to be present.

The data of the case study is used to construct a random distribution, such that we can draw a sample path, as is descripted in line 10 of Algorithm 1. Since the case study does not cover all container flows, due the limited range of the data set, the separated aspects of the flows in the data set are used to construct a new distribution. The different aspects of the flows, such as the import versus export, origin versus destination, the release times, the delivery window and the amount of containers, are translated into fractions. These fractions are used as probablities, to construct the random distribution.



(a) All containers (b) Export containers (c) Import containers

Figure 5: Histograms of the realized amount of containers (F). For export, import and in total.

6 Results

This section is divided in different subsections, first the different configurations of the model will be discussed and secondly the different initalisated models will be discussed on the evaluation. All calculations and programs are done in Python using a HP-laptop with an i5 Core processor.

6.1 Numerical results of different model initialisation

First the initalisation of the weights in the model will be dicussed and their corresponding results. Secondly, the effect of adding a constant to the different features is invested. After that the influence of the inneriteration value is discussed, by comparing different settings of this value. Finally, we will look into the effect of some specific features. In Table 3 the different models are shown, with the different specifications per model.

Table 3: Model specifications, with $N = 10000$

Model	M	**Runningtime (sec)**	**Initialisation weights**	**Constant**	**Other**
1	10	8924	1	NO	
2	10	5449	0	NO	
3	10	7124	100	NO	
4	10	8673	1	YES	
5	10	9258	100	YES	
6	10	7425	0	YES	
7	10	4441	random	YES	
8	10	6900	random	NO	
9	5	5604	1	YES	
10	20	5119	1	YES	
11	1	7275	1	YES	
12	2	10056	1	YES	
13	10	4112	1	NO	*only 2 features*
14	10	5099	1	YES	*only 2 features*

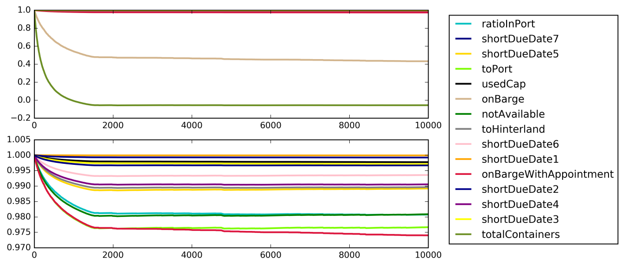


Figure 6: Development of the weights of model 1

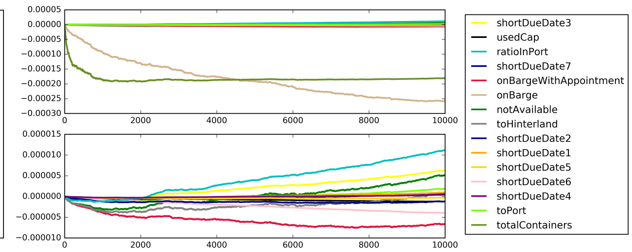


Figure 7: Development of the weights of model 2

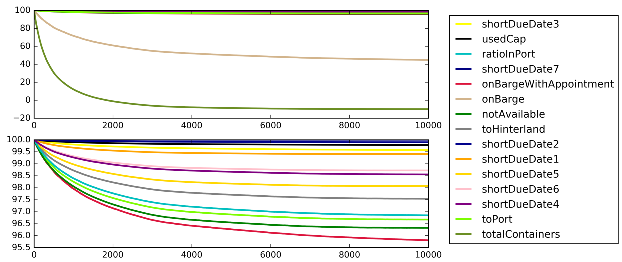


Figure 8: Development of the weights of model 3

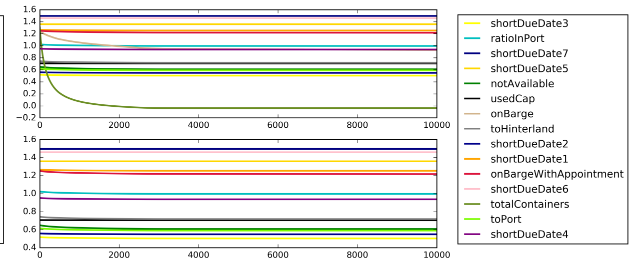


Figure 9: Development of the weights of model 8

6.1.1 Intialisation of the weights

First models 1, 2, 3 and 8, will be discussed. These models only differ on the initalisation of the weights. The inneriteration counter (M) is in all models 10, and no constant is added in these models. The running times are similair, although model 2, with an initalisation at zero, has lowest running times. The difference between the models is as said only the initalisation of the weights of the features in the cost estimation. Model 1 has an initalisation at 1, model 2 at zero. The weights in model 3 are initialised at 100, and finally in model 8 we initialise the weights of the features randomly between 0.5 and 1.5 ($\theta_f^0 \sim U(0.5, 1.5)$). These different initalisations are chosen to see what the implications of the initalisation of the weights can be and if and how these initialisations influence the prediction results.

We evaluate the models based on two parts. First, we look at if and how the weights develop during the iterations. We also look at the performance of the model, how are the estimations? How well do they estimate the expected cost?

In Figure 6 the development of the weights of model 1 is shown. The plot above is the complete plot, while the second plot is zoomed in around the initialisation value of the weights, to show a clear vision of the development of all the weights. Figure 7, 8 and 9, show the development of the weights of respectively models 2, 3 and 8.

When first only examining the weights of model 1 (Fig. 6), two variables show a large deviation of the others, namely variables *onBarge* and *notAvailable*. These variables have an absolute range instead of ranging between zero and one, as the other variables do. In the first 2000 iterations a clear trend is visible, while after 2000 iterations this seems to be somewhat stable or converged.

A similar pattern is visible in the development of the weights with an initalisation at 100 (Fig. 8), but it takes more time and iterations to converge. This might be due to the value of ϵ, which partially determines the stepsize for changing the values of the weights.

For an initalisation at zero (Fig. 7) a different pattern is shown. There is no clear convergence visible for most of the variables. Again with this initalisation variables *onBarge* and *notAvailable* show a trend downwards, different from the other variables. The different pattern for an initalisation at zero, in comparison with the other two models, might be due to ϵ. Since ϵ is relatively large compared to 0, especially compared with the initialisation of the weight with 100.

Then finally the model with the initalisations at random, model 8 (Fig. 9). We should state that this is only one initialisation, with only one random draw from this random distribution. If we look at the results carefully, again two variables (*onBarge* and *notAvailable*) show a pattern towards another value. For the variable *notAvailable* the value is almost going to zero. While for all other variables the initalised value does not change that much.

We want to compare how well a model is estimating the costs for a certain state. Since the final goal is to estimate the expected costs as accurately as possible for a certain state. To get a indication for the estimation accuracy, the estimated costs are compared with the costs calculated in the model afterwards. This can be done in a plot like Figure 10. Here the calculated real costs are plotted on the x-axis and the estimated costs on the y-axis. For model 2, 3 and 8, respectively Fig. 11, 12 and 13, can be compared.

First we notice that in all cases we always estimate the costs higher than the real calculated cost. For the second model (Fig. 11) an interesting pattern is visible. The estimated costs form kind of clusters, for different values of real calculated costs. This might be due to some states being similar for the estimation function, but in reality more different from each other. So this might suggest that these state have a variable that is equal or similar for all states, but results in different real cost.

Secondly, we can observe that a lot of observations are visible around an estimation of zero costs. Since in a scatterplot this is hardly seen, Table 4 gives information on the different models. In absolute values, since all values of the costs are between zero and one (as the cost is modelled as the percentage of containers transported by truck given the total number of containers in the system), the errors are small, which results in an even smaller MSE. So besides the MSE, the MAE is also calculated. To quantify the percentage of costs estimated at zero, the percentage smaller than 0.001 is shown in the last column. As we can see for some models like model 1, this is relatively high, as well as for model 8. On the other hand, for models 2 and 3 this is lower, resulting in a higher MAE.

This all together shows that the initialisation of the weights has an effect on the estimations, and together with the value of ϵ, on how the weights change over time in the iterations. The initialisation at one (model 1) as well as the random initialisation of the weights (model 8), performs rather good and stable.

Table 4: Error information for the different models, with and without a constant.

	without constant				with constant		
Model	MAE	MSE	%<0.001	Model	MAE	MSE	%<0.001
1	0.0041	0.0002	0.851	4	0.0105	0.0004	0.419
2	0.0118	0.0004	0.334	6	0.0085	0.0002	0.38
3	0.0099	0.0006	0.538	5	0.078	0.0003	0.51
8	0.0018	0.0001	0.864	7	0.0101	0.0007	0.652

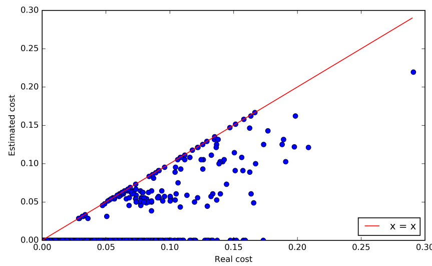


Figure 10: Estimation results of model 1

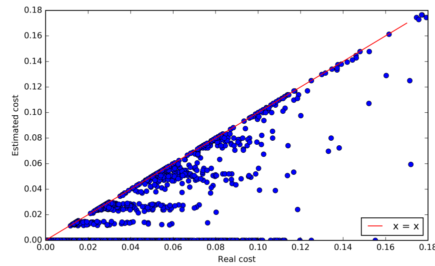


Figure 11: Estimation results of model 2

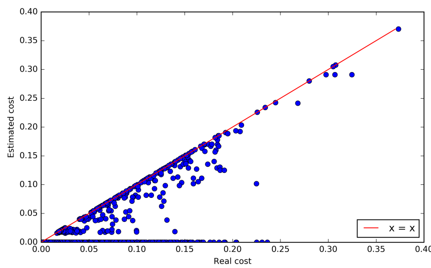


Figure 12: Estimation results of model 3

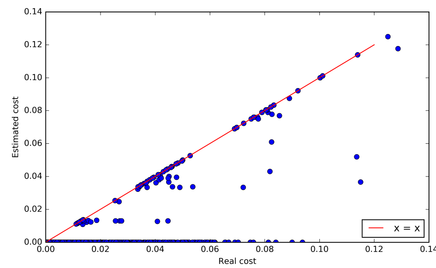


Figure 13: Estimation results of model 8

6.1.2 Adding a constant feature

An other way of changing the input of the models is as described in the list of features (Powell, 2007). Adding a constant feature in the model could help to eliminate the fixed costs in each situation and could contribute to a better estimation of the costs. Model 4, 5, 6 and 7 (Table 3) are models with an added constant. Model 4 is similair to model 1 only the constant is added. So the results should be compared with the corresponding model without the constant.

As we see in Figures 14-17 a similar pattern as before for all the different weights is shown. For example, in Figure 14 and 15, we see a clear red line indicating the value of the weight for the constant in these models. This line seems to converge. For the model with an initialisation at zero the line for the constant is clearly visible as well, but shows a lot of fluctuations, compared to a model with an initialisation at zero (model 2).

So for how the weights change over time, no different observations can be made with or without a constant. Overall no large changes in the other values of the weights can be observed. In some specific cases, we see some differences in the final versions of the weights.

For comparing the results of the estimations we can inspect Table 4 and Figures 18-21. The table shows on the left hand side the values of the models without an added constant, and on the right hand side, the same results for models with a constant. Model 1 and model 4 both have an initalisation at one, implying that the results can be compared.

For the percentage of estimated cost very close to zero, we see an equal or lower percentage in the table with the added constant in the model. This can be explained by the fact that the constant is added, so in all states a basic cost is always assumed. The higher the weight of the constant, a larger basic cost is estimated in each state. Only situations with a large positive effect on the state, will result in an estimated cost close to zero.

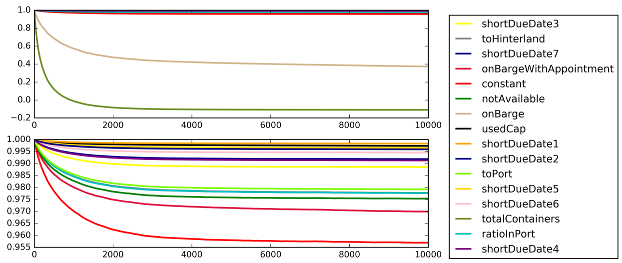


Figure 14: Development of the weights of model 4

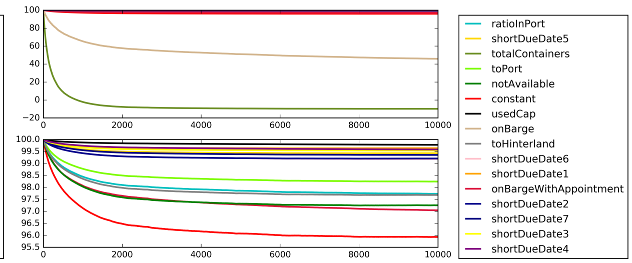


Figure 15: Development of the weights of model 5

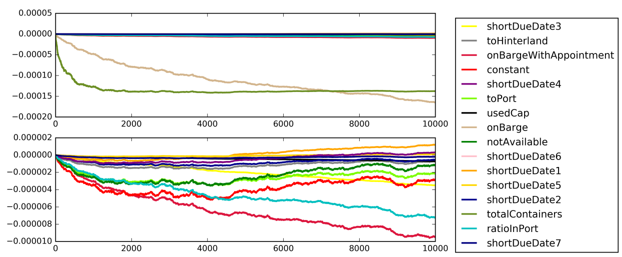


Figure 16: Development of the weights of model 6

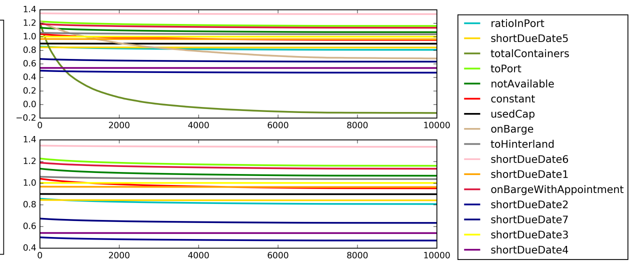


Figure 17: Development of the weights of model 7

When comparing the MAE with each other, we see that for model 6 an 5, a lower MAE is reached than in a model without a constant. This suggests that the model with an added constant is better in predicting the true calculated costs. For the models initialised at 1 (models 1 and 4) and randomly initialized (model 8 and 7), a higher MAE is reached in the case with an extra constant. This does not give a clear answer to the question whether a constant should be added or not. But since the weights of the initialisation at zero (model 2 and 6) have not completely converged, we could argue that these results are not representive for the models initialized at zero. In most cases adding a constant would lead to a better prediction. An other way to argue that a constant should be added is by saying the fixed costs in each state will now be represented in the weight of the constant feature.

Overall adding a constant will lead to more benefits than the possible harm it will cause to the model and its prediction power.

6.1.3 Inneriteration value (M)

In Algorithm 1 the inneriteration value M is used for the update of the coefficients of the estimation function. This M determines in which iteration the values are updated. This to ensure a certain iteration with possible outliers does not effect the values too much (Powell, 2007).

To see if and how the inneriteration value M has an effect on the coefficients and the predictions of these coefficients, models can be made with different inneriteration values. For the first models discussed in this thesis an inneriteration value of 10 was used. For comparison, model 4, can be compared with models 9 - 12. With respectively an inneriteration value of 5, 20, 1 and 2, as can be seen in Table 3.

Model 4 is compared with the other models, as we have seen before the coefficients change during the iterations as seen in Figure 14. This pattern is really similair with the patterns seen, and quite charasteric for this case.

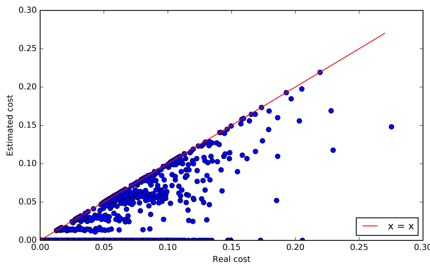


Figure 18: Estimation results of model 4

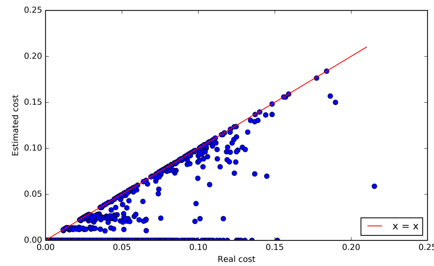


Figure 19: Estimation results of model 5

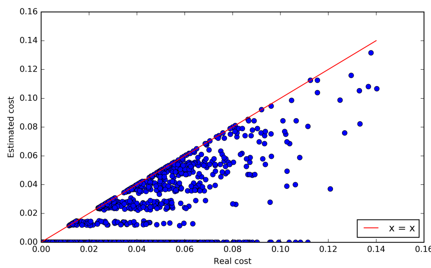


Figure 20: Estimation results of model 6

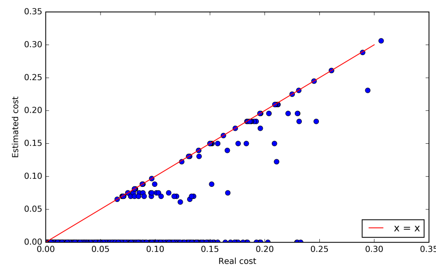


Figure 21: Estimation results of model 7

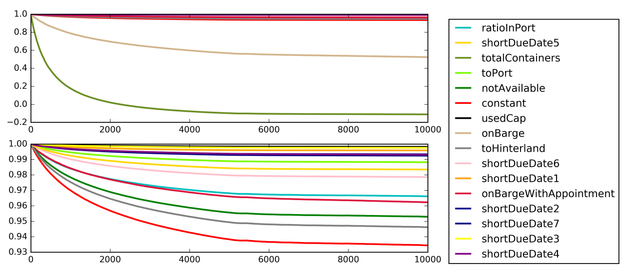


Figure 22: Development of the weights of model 9

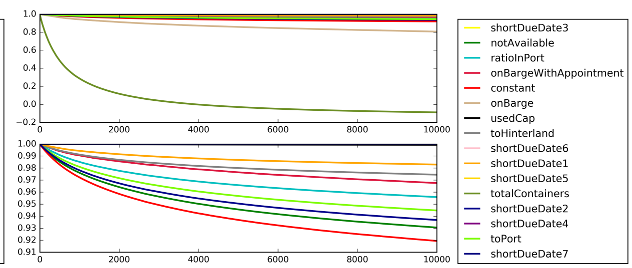


Figure 23: Development of the weights of model 10

For models 9 - 12, Figures 22 - 25, show the changes of the weights during the iterations. When comparing these results with the model of 4, we see a slower convergeces of the weights. And some features get assigned a different weight in model 9. This could be due to the randomness of the runs. Since all model weights are based on some randomly created instances.

Model 10 (Fig. 23) shows an even slower convergence, in the weights. As for model 4, after 10000 iterations, the weights have converged, but for model 10, this is rather hard to say. The weights change rather slowly, with very small steps. This could be due to the fact that the inneriteration value is twice as high as in model 4.

In model 11 (Fig. 24) the inneriteration value $M = 1$. The change of the weights seems to be similair to Model 4, but could have the possible disadvangtage of the large effects of the randomness. Similair conclusions can be drawn for model 12.

When we look at the prediction power of the different models, Table 5 gives an overview of the different models and their different errors in the predictions. Model 10, seems to have the lowest errors, but has also a high percentage of very low estimations of almost zero

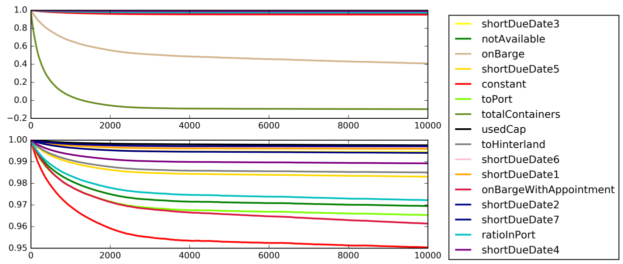


Figure 24: Development of the weights of model 11

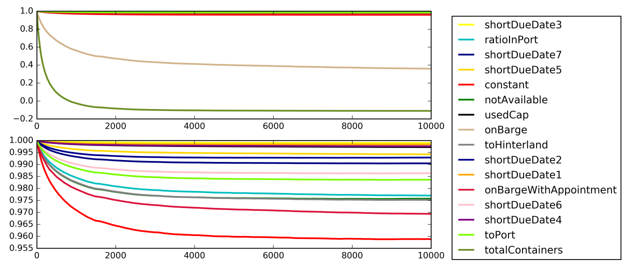


Figure 25: Development of the weights of model 12

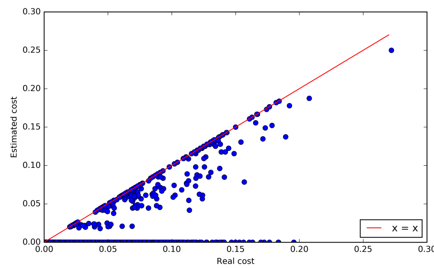


Figure 26: Estimation results of model 9

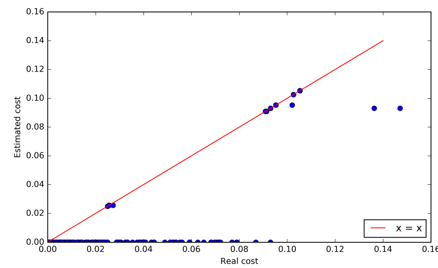


Figure 27: Estimation results of model 10

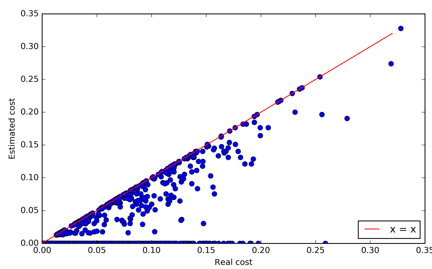


Figure 28: Estimation results of model 11

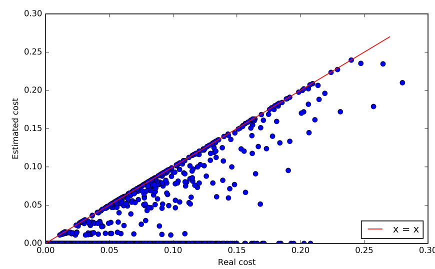


Figure 29: Estimation results of model 12

estimated cost. When looking at the plot of the estimated cost, versus the realistic cost in Figure 27, almost all costs are estimated at (almost) zero, except two clusters, which are rather accurately estimated. The fact that more estimations are close to zero can be due to the fact that the weights have not converged as much as the other models show more signs of convergence.

The models 9, 11 and 12 show a similair pattern as model 4 in Figure 18. These figures give no new insights in the used algorithm. The prediction errors of the different models is except for model 10, similair, if we look at Table 5.

The question which inneriteration parameter M should be used, can be best answered with the convergence of the weights. An inneriteration parameter should be not too large (like model 10), or too small like models 11 and 12, due the influences of randomness. Using a very small inneriteration parameter will lead to higher running times (Table 3), while using a very large M will increase running times as well, because it requires more time to converge.

So Model 4 or 9 are preferred over others.

Table 5: Error information for the different inneriteration values M.

Model	MAE	MSE	%<0.001
4	0.0105	0.0004	0.419
9	0.0076	0.0004	0.676
10	0.002	0.0001	0.887
11	0.0098	0.0005	0.535
12	0.0118	0.0007	0.456

Table 6: Error information for the different sets of features.

Model	MAE	MSE	%<0.001
1	0.0041	0.0002	0.851
13	0.0027	0.0001	0.797
4	0.0105	0.0004	0.419
14	0.0016	0.0001	0.893

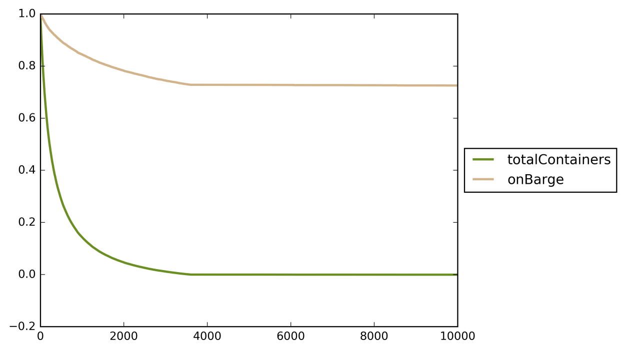


Figure 30: Results of the weights of model 13

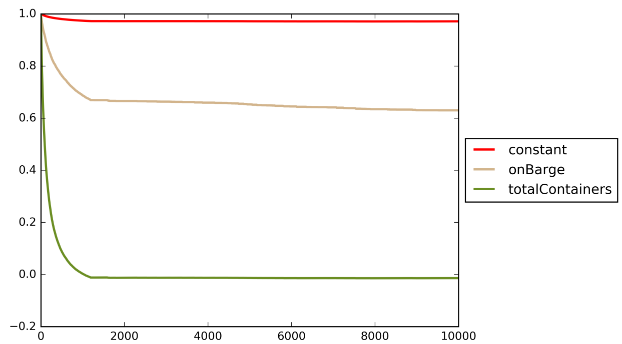


Figure 31: Results of the weights of model 14

6.1.4 List of features

Models 13 and 14 only contain a small number of features, namely the *totalContainers* and *onBarge*. These two features show in the previous models a large deviation in the weights, and possibly thus a high importance in modelling the costs of a state. To see if the other features contribute we can compare these two models with respectivily models 1 and 4.

The convergence of the weights of both models are similar and shown in Figure 30 and Figure 31. The final weights are similair to the weights in model 1 and 4. When looking at the plots with the estimated costs, plotted against the realised costs, we see again high percentages of very low cost estimations, as is also visible in Table 6. Figure 32, for Model 13, shows the clusters again for the different costs. These clusters can also be explained by the low number of features, in such a way that there are less features to determine in which state the algorithm currently is. This could partially explain the clusters.

In Table 6, the error rates seem to be lower in the models with less features. One could argue that these models predict better. So looking into the contribution of certain features could be profitable. But on the other hand, the two chosen features are the only two which are not a ratio between $[0, 1]$ but an absolute number. This could also explain a part of the high impact of these two variables. One could argue that this high impact is grounded since this two have a high impact on whether the barges can deliver the containers in time.

6.2 States and kind of results

In what kind of states are these desicision taken and are they representable for the case study? To conduct a good research into some models, only models 1, 4, 9 and 14 are taken

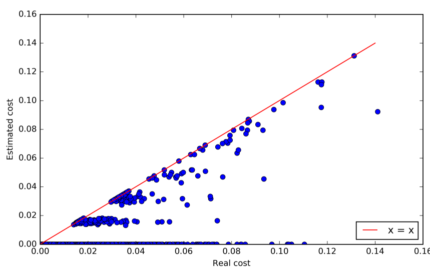


Figure 32: Estimation results of model 13

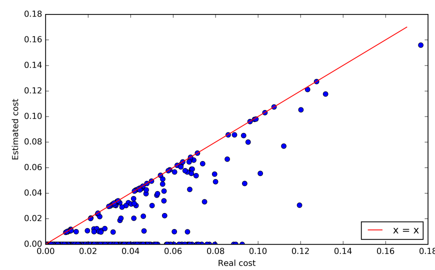


Figure 33: Estimation results of model 14

Table 7: Avergares and variances of the different feature values throughout the runs of four different models, namely model 1, 4, 9 and 14.

Model	1				4				9				14			
Feature	Mean	Var	Min	Max	Mean	Var	Min	Max	Mean	Var	Min	Max	Mean	Var	Min	Max
onBarge	3.116	51.102	0.000	44.000	9.591	49.999	0.000	42.000	4.483	22.199	0.000	23.000	1.927	23.361	0.000	33.000
onBargeWithAppointment	0.133	0.094	0.000	1.000	0.575	0.152	0.000	1.000	0.352	0.187	0.000	1.000	-	-	-	-
usedCap	0.011	0.001	0.000	0.137	0.033	0.001	0.000	0.140	0.015	0.000	0.000	0.083	-	-	-	-
ratioInPort	0.529	0.073	0.000	1.000	0.529	0.074	0.000	1.000	0.530	0.073	0.000	1.000	-	-	-	-
totalContainers	65.000	12.885	52.000	104.000	69.344	25.804	51.000	99.000	44.670	11.447	35.000	60.000	96.067	20.533	70.000	111.000
toHinterland	0.312	0.000	0.192	0.421	0.523	0.001	0.392	0.644	0.820	0.000	0.667	0.867	-	-	-	-
toPort	0.688	0.000	0.579	0.808	0.477	0.001	0.356	0.608	0.180	0.000	0.133	0.333	-	-	-	-
notAvailable	0.617	0.001	0.385	0.769	0.580	0.002	0.404	0.784	0.720	0.003	0.533	0.914	-	-	-	-
shortDueDate7	0.086	0.000	0.000	0.222	0.086	0.001	0.000	0.222	0.103	0.002	0.000	0.400	-	-	-	-
shortDueDate6	0.241	0.001	0.000	0.429	0.132	0.001	0.000	0.308	0.329	0.008	0.000	0.750	-	-	-	-
shortDueDate5	0.400	0.002	0.037	0.714	0.076	0.001	0.000	0.207	0.269	0.008	0.000	0.444	-	-	-	-
shortDueDate4	0.255	0.001	0.000	0.545	0.195	0.001	0.037	0.412	0.080	0.007	0.000	0.364	-	-	-	-
shortDueDate3	0.014	0.002	0.000	0.364	0.280	0.003	0.000	0.533	0.043	0.003	0.000	0.375	-	-	-	-
shortDueDate2	0.004	0.000	0.000	0.444	0.198	0.004	0.048	0.435	0.116	0.002	0.000	0.500	-	-	-	-
shortDueDate1	0.000	0.000	0.000	0.182	0.033	0.002	0.000	0.313	0.060	0.008	0.000	0.429	-	-	-	-
constant	-	-	-	-	1.000	0.000	1.000	1.000	1.000	0.000	1.000	1.000	1.000	0.000	1.000	1.000

into consideration. For the four different models the different features are valued on their mean, variance, minimum and maximum. These results are all visible in Table 7.

Models 4 and 9 contain all the features including the constant, model 1 contains all the features without the constant. Finally, model 14 contains only the constant, and the features *onBarge* and *totalContainers*.

Starting with the features related to the barges and the containers that are on the barges, namely features *onBarge*, *onBargeWithAppointment*, *usedCap*, *ratioInPort*. The feature *onBarge* has in all models a rather low mean, and a very large variance. This could be explained by the fact that when inspecting the states individualy in a lot of cases no containers are on board, and in some (more rare) cases a lot of containers are on the barges. This can explain the large variances with the low means. This could also point out the fact that the model options are not yet perfectly modeled, and need some improvements. The maximum number of containers is in model 9 relatively lower in comparison with the other models.

The feature *onBargeWithAppointment* is in the range $[0, 1]$ and this is also visible in the minimum and maximum values of the feature. If the number of containers on the barge is zero, the *onBargeWithAppointment* is zero as well. This feature has a higher mean in model 4 compared to model 9, and the lowest mean in model 1. The variances are relativily similair for all the models.

The capacity on the barges in this case study is 200 TUE in total. This fact and the number of containers on the barges can be used to explain the low *usedCap* rates in the models. The averages are very close to zero and show little variance. Since it is a ratio it can be at most 1, but in these cases the capacity ratios never rise above 0.2. A possible explaination could be in the random distributions for the new arriving containers. If the avaible containers are low in comparison to the capacity of the barge, it can result in a low ratio of the *usedCap*.

The last feature related to barge is *ratioInPort*. Barges which have their next appointment at a terminal in the port are counted as in the port area. The three different barges can be in the port or not. This feature will have the values $(0, \frac{1}{3}, \frac{2}{3}, 1)$. In all models the mean of this feature is a bit over 0.5, which shows that barges spent more time in the port. However, this could also be an effect of the 4 port terminals chosen and only 3 inland terminals.

The other features are more related to the container supply and demand. Like *totalContainers*, *toPort* and *toHinterland*, *notAvailable* and the *shortDueDate*.

As said before the feature *totalContainers* is one of the features which is not scaled, but is an absolute value of the number of containers which are in total in the system known. These are both the containers already on the barges or present at the terminals, as well as the containers which are currently known in the system, but have not arrived at a terminal yet. Model 9 has less containers on average than the other three models, this is well visible in the minimum and maximums in the different models. The difference in the total number of containers in the model can partially be explained by the (random) initialisation of the problem, while the chances of new incoming arriving containers are similair for all models that were tested.

The features *toPort* and *toHinterland* depend on each other. This is the percentage containers that have the destination in the hinterland (import) or in the port for export purposes. As we can see, model 1 has a very different ratio between the import and the export on average, than models 4 and 9. Recall from the data of the case study that we had

65 import and 122 export containers. Hence, we would expect more export containers than import containers. The means of model 1 are more similar to the data of the case study. These large differences between the models could also be due to the initialisation. Another argument for this claim is the low variances, with a small range between the minima and maxima, which suggest no large changes in the container supply.

The feature *notAvailable* is related to *totalContainers* which represents the part of the containers, that are known but not present yet to be picked up. As we can see this percentage is rather stable, around half of the containers for all the models. In all models we see low variances.

The other seven features for the *shortDueDate* are similair as *notAvailable*, related to *totalContainers*. These represent the fraction of containers which is due in 1,2,..,7 days. The averages over the due dates differ across the models, but these differences might be due to the initialisations. In an ideal situation these ratios would match with histograms in Chapter 5. However, in all models we see again these low variances in values of the demand and supply side of containers. This suggests that the ratio almost does not change. This would suggest further research into the way new entries of containers are introduced to the system.

At last the averages of the *constant*, this is always one, due definition, as the results in Table 7 show.

7 Conclusion

The main question of this research was how to model the choices of barges planners in such a way that the model can be used in a DSS for planners. When comparing with the different research already done in barge planning, most research assumes a fixed routing of barges (Kooiman et al., 2016) or a fixed demand of containers. The model of Rivera and Mes (2016) assumes a network where the freights are stochastic and a lot of different transfer selections are possible with this model.

When comparing the problem described with the problem of Rivera and Mes (2016), it makes a good start for modelling. When adapting this model, a solution need to be found for the routing of the barges. Therefore, the problem can be classified as $\overline{R}, [RO], [RD], [RDT], \cancel{RP} | \widehat{D}, [D2R] |$social(1) in the classification method of De Juncker et al. (2017), where the objective is to maximize the ratio of the containers delivered by barge given the total number of containers.

To model the problem of the barge planners, a Markov Decision Process (MDP) is used. Extending the notation of Rivera and Mes (2016), a state can be described as the number of freights with an assigned current location, destination, release date and due date, as well as the current sailing schedule of the different barges. In between the different states new information can become known. This can also be modelled in the MDP as the chance of entering a certain next state, also depending on the choice made by the planner.

Since a large extended MDP cannot be solved mathematically, a different solution method, such as ADP, is requested. When, like in this problem, iteration over the states is not an option, a policy iteration alogrithm with basic functions could be a good option. Other options for applying ADP on this model have not been explored, but could be worth exploring, since an approximation of a model might be too general for some cases. However, this approximation is chosen since it does not iterate over all the states, which is needed due the very large number of states.

The algorithm of the policy iteration alogrithm with basic functions is a good start and has lots of options to optimize the algorithm. First of all, in this model only a linear basic function with a certain list of features is used. But other options for functions or features might result in better results, different per case. Research into the effects of features on the costs of a state and the possible relation between them, might result in a better basic function to approximate the costs of a certain state.

The algorithm needs an initialisation of the weights in the basic function. As can be concluded for our research, the initialisation of the weights does have an effect on the convergence of the algorithm as well as on the results. Different initalisation of the weights have been tested in this research. The algorithm also needs an initialisation of the parameter ϵ which could effect the change of the weights, if the weights are relatively small in comparison with ϵ. These effects have not been researched yet. If this suspected effect is real, it would explain the different results that an initialisation with weights at zero shows in comparison with other initalisations.

The objective that is used to optimized this model, *maximize the ratio of the containers delivered by barge given the total number of containers*, is in the interval $[0, 1]$. When large amount of containers are already known, but not yet present on the terminal, the objective will be small and is in a lot of states close to zero. This can be seen in the figures like Figure 10, where a lot of costs are really close to zero. Differentation between zero and a number close to zero is difficult. Another objective could be worth researching, or with less changes a logaritmic scale for the ratio could be a simpler solution for differentation between small values.

For larger instances this solution method can lead to problems due the possibly long running times. These running times increase if the number of options in each state increases. The relation between extra terminals and/or barges is not linear but increases drastically when adding an extra terminal due the increase in options for transitions to next states. Due to the long running times with large instances, this model can only be used in limited clear distinct instances, where the influences of the instances are minimal or can be modelled in the transition probability for changing to a certain state.

As we saw throughout the different models the random elements of the models can have an effect on the prediction power of the model. As the models are initialised with a certain division over the current available containers and no containers are in the barges yet, the first runs are highly effected by the initialisation. Using a large initialisation period is adviced, to minimize the effects of the initialisation on the final prediction of the weights in the model.

Besides the effect of the random initialisation, the new incoming containers are based on a random element as well. The distributions for these containers were estimated based on the data of the case study as we described in Chapther 5. But as the averages of for example feature *usedCap* show, the capacity of the barges is barly used. This could suggest that the distributions of the case study do not match the current capacity of the barges. Research with other data might show if this is indeed the case, or that another explanation could be the cause of these figures.

To conclude this research, this model has the potential to be used in a DSS for barge planners. The basic cost function can be estimated quickly for a barge planner, but determining the weights of the basic function is a time consuming proces. For larger instances this might get problematic, preferably the model should be applied on clear distinct instances. We would advice to test different initalisations, since these can have large effects on the prediction results. More research is needed especially into the effects of the random elements of the initalisation and the model. Testing this model with different case studies could give a better insight in the performances of the model.

7.1 Acknowledgements

At last I would like to thank my colleages at TNO for showing me that a passion for optimisation is not weird. My supervisor at the Erasmus University Rotterdam for the patience he had with my thesis. And ofcours my friends and family who kept me motivated to finish off this project.

References

De Juncker, M., Huizing, D., del Vecchyo, M. O., Phillipson, F., & Sangers, A. (2017). Framework of synchromodal transportation problems. In *International conference on computational logistics* (pp. 383–403).

Douma, A., Schutten, M., & Schuur, P. (2009). Waiting profiles: An efficient protocol for enabling distributed planning of container barge rotations along terminals in the port of rotterdam. *Transportation Research Part C: Emerging Technologies*, *17*(2), 133–148.

Konings, R. (2007). Opportunities to improve container barge handling in the port of rotterdam from a transport network perspective. *Journal of Transport Geography*, *15*(6), 443–454.

Kooiman, K., Phillipson, F., & Sangers, A. (2016). Planning inland container shipping: a stochastic assignment problem. In *International conference on analytical and stochastic modeling techniques and applications* (pp. 179–192).

Mulder, J., & Dekker, R. (2012). Designing liner shipping networks. *Report/Econometric Institute, Erasmus University Rotterdam*, 1–40.

PoR. (2011). *Port vision 2030.* Retrieved from `https://www.portofrotterdam.com/en/the-port/port-vision-2030` (Accessed: 12-10-17)

Powell, W. B. (2007). *Approximate dynamic programming: Solving the curses of dimensionality* (Vol. 703). John Wiley & Sons.

Rivera, A. P., & Mes, M. (2016). Service and transfer selection for freights in a synchromodal network. In *International conference on computational logistics* (pp. 227–242).

Tijms, H. C. (2003). *A first course in stochastic models.* John Wiley and sons.