

Barge rotation planning and quay scheduling in the port of Rotterdam

Albert Douma*, Peter Schuur, Marco Schutten

University of Twente, The Netherlands

Beta working paper WP-217

May 15, 2007

Abstract

We consider the problem how to align barge rotations with quay schedules of terminals in the port of Rotterdam. Every time a barge visits the port, it has to make a rotation along, on average, eight terminals to load and unload containers. A central solution, e.g., a trusted party that coordinates the activities of all barges and terminals, is not feasible for several reasons. One of the reasons is that barges and terminals want to stay autonomous and in control of their own operations. We propose a multi-agent based approach of the problem, since a multi-agent system can mirror to a large extent the way the business network is currently organized and can provide a solution that is acceptable by each of the parties involved. We examine the value of exchanging different levels of information and evaluate the performance by means of simulation. We compare the results with an off-line scheduling algorithm. The results indicate that a multi agent based control is promising for this problem.

Keywords Barge planning, quay scheduling, multi agent based control, simulation, dynamic assignment

1 Introduction

The barge rotation planning and quay scheduling problem we consider is based on a real problem in the port of Rotterdam. The port of Rotterdam is the largest container port of Europe and handled

*A.M. Douma, University of Twente, School of Management and Governance, P.O. Box 217, 7500 AE Enschede, The Netherlands, a.m.douma@utwente.nl.

about 5.8 million containers in 2006. Most of these containers are shipped to the hinterland by means of train, truck, and barge. In this study, we focus on the barge container sector, which has become an increasingly important transportation mode over the past decades. We will specifically look at the problem how barge rotations (sequence of terminal visits) and terminal quay capacity can be aligned in such a way that terminals achieve a high utilization and barges can leave the port according to their sailing schedule.

The barge rotation planning and quay scheduling problem is rather complex. In the port, there are about 30 terminals, whereas 75 to 100 barges visit the port daily, each visiting eight terminals on average. Some of the terminals also handle sea going vessels and some are closed during the night, which limits the possibilities to handle barges. Currently, the planning of a barge rotation is done by the barge operator that has contracted the barge. The planning of the quays is done by the terminal operator of the corresponding terminal. On arrival of a barge in the port, the concerning barge operator contacts all terminals that need to be visited and tries to make appointments such that the barge can leave the port in time. Communication with the terminals is done by means of phone, E-mail, and fax. However, this takes too much time and due to changes and disturbances, it is often not possible to execute a barge rotation or quay schedule as planned. This is due to, among others, the high interdependency of terminals and barges. A delay at one terminal has a dominoes effect on the rest of the terminals. Moreover, it turns out that information is not reliable since barge operators try to obtain convenient time windows by, e.g., lying about the real number of containers that need to be loaded and unloaded, to get a preferred or to guarantee enough planned handling time at the terminal. Terminal operators, on the other hand, know this and take that into account when they schedule the barge. We refer to this phenomenon as ‘strategic behavior’.

Currently, a poor alignment of activities, uncertainties during operations, and strategic behavior of both terminals and barge operators lead to inefficient use of quays and long sojourn times of barges in the port. For freight forwarders this affects the reputation of the port and the attractiveness of using barges for transportation. This is a serious issue today.

In the past, several attempts have been made to establish a central trusted party that coordinates the activities of both terminals and barges. However, these attempts were not successful due to the following reasons:

- *Autonomy.* Every terminal and barge operator wants to stay autonomous, i.e., in control of its own operations.
- *No contractual relationships.* There exist no contractual relationships among barges and terminals, which means that barge operators and terminal operators cannot contractually force each other to deliver a certain service or charge each other for poor services.
- *Limited information sharing.* Since barge operators compete with each other (as terminal operators do), they are very reluctant to share information that possibly deteriorates their competitive position.

Additionally, from a mathematical point of view the problem is even more complex since we deal with

- Different players each having different interests.
- A highly dynamic environment, i.e., there are a lot of events and disturbances.
- A low structured and loosely coupled network, i.e., there exists no clear hierarchy in the market and every day different players can be involved (different barges or even new terminals).

A solution for the barge rotation planning and quay scheduling problem should mirror the current market structure. Important is that the solution is acceptable by all parties involved. Acceptability is possibly even more important than optimization. Optimization is a hard concept in this context, since all parties have different interests and we deal with a dynamic setting in which barges arrive over time.

The aim of this study is to develop and evaluate a multi agent based control to align barge rotations and terminal quay capacity and to gain insight in the way the proposed system functions. We define key performance indicators and compare the results of the multi agent based controls (obtained by simulation) with an off-line benchmark, which is a central optimization algorithm that assumes that all information over the whole planning horizon to be known in advance. We only consider deterministic - dynamic settings. Deterministic means that all sailing and handling times are precisely known. Dynamic means that barges arrive over time and plans are made and updated in real time. In this study, we do not consider disturbances and the effect on the operations of

barges and terminals. We assume that all decisions (made by terminals and barges) are made in real time and that never two barge operators plan a rotation concurrently, but one after another.

The outline of the paper is as follows. First we mention some related problems and earlier studies to the barge rotation planning and quay scheduling problem. In Section 3 we discuss the decentralized approach and the way the communication among barge and terminal operators is modelled. We describe different levels of information exchange and discuss the barge operator and terminal operator intelligence. In Section 4 we describe the off-line benchmark model and the algorithm used to solve the model. Section 5 describes the experimental settings we considered. Section 6 presents the results obtained by means of simulation (for the decentralized controls) compared with the results obtained with the off-line benchmark. Finally, we draw some conclusions and show directions for further research.

2 Related problems

The problem we consider is about routing and scheduling decisions. We mention two related problems in the literature, namely the berth allocation problem and the attended home delivery problem. Besides we mention an earlier study to the barge handling and quay scheduling problem in the port of Rotterdam.

The berth allocation problem (BAP) concerns the assignment of berths to ships (for an overview, see, e.g., [18][4]). In the BAP, it is usually assumed that the arrival times of ships as well as the processing times are known at the time the berth plan is made (see e.g. [4][10]). In our problem, arrival times of barges are uncertain and terminals have to plan quays based on partial information.

The problem we consider is also related to the attended home delivery problem (AHDP) [2], where a carrier offers attended deliveries of packages at the homes of customers. An attended delivery means that a customer has to be present when the package is delivered. To optimize the route the carrier travels, [2] consider incentive schemes to influence the preferred time windows of a customer. We consider a similar problem, however, having multiple carriers. Each carrier visits a subset of the total set of customers. Customers can deal with only one carrier at a time and customers want to plan carriers close to each other in order to decrease the time they need to be at home. In our problem, customers can use incentive schemes to let the carriers apply for

convenient time slots. Carriers in turn aim to minimize the time they have to travel for delivering all packages.

Some years ago a first study was performed to the barge handling and quay scheduling problem in the port of Rotterdam to investigate the possibilities for a decentralized control structure [3][13][15]. This study focussed on an off-line planning system, where barge rotations were planned one day in advance and not updated during execution. The aim of the study was to create feasible plans. In this study we model a real time planning system and, besides feasible plans, we aim to optimize the operations of both barges and terminals. To reveal the interests of barges and terminals, in-depth interviews were performed with both barge and terminal operators [7]. This gave us insight in how responsibilities are divided over the chain and how different parties react upon each other. These elements are important for an agent-based control system which is implemented in an inter-organizational setting.

The contribution of this study to the existing literature is the following. First, we present a business problem where decentralized control, in contrary to centralized controls, could provide an acceptable solution for the parties involved. Second, we develop a decentralized control based on waiting times. Third, we evaluate the added value of exchanging different levels of information by means of simulation and compare the results with an off-line algorithm which solves the problem centrally. Fourthly, we provide insight in the way a decentralized system functions containing multiple competitive parties which are mutually dependent on each other.

3 Decentralized approach

In this section, we present a multi agent based approach to the barge rotation planning and quay scheduling problem. All players in the network are *opportunistic*, meaning that opportunities are exploited for their own benefit with no regard for the consequences for other players. We focus on a real-time planning system in which barge rotation and terminal quay plans are made and updated over time and decisions are made in real-time. We first present the conceptual agent model, describe alternative levels of information exchange and present algorithms for both the barges and the terminals.

3.1 Conceptual multi agent model

To define the agents, we applied a *physical decomposition* of the problem, since this comes closest to the current situation. A physical decomposition means that agents represent entities in the physical world [17], i.e., every barge operator and every terminal operator is equipped with an agent acting in their best interest. To simplify our design, we model only two different types of agents: barge operator agents and terminal operator agents. We assume that all barge operator agents are equal and the same holds for all terminal operator agents. So, every barge operator agent would make the same decision if it would be in the same situation as an arbitrary other barge operator agent. We focus on the question how to create intelligence for these agents such that both individual and mutual benefits can be achieved.

The communication mechanism is important for the way agent intelligence is developed. Agents can communicate directly (negotiation), or indirectly by means of a contract net protocol [19]. The latter requires however a kind of (virtual) currency that is exchanged between parties, which is strange in the current situation, since there are no contractual relationships between terminals and barges. We propose a direct communication mechanism (negotiation) that mirrors daily practice and is probably easier accepted by barge and terminal operators. Direct communication means that every barge operator contacts all the concerning terminal operators directly to make appointments. Based on the responses of all terminal operators it decides when it is going to visit every terminal and in which sequence. This is actually the way the manual system works at the moment.

To illustrate the proposed agent model we depicted one barge operator contacting several terminal operators about convenient handling times – see Figure 1. In the figure, we see one barge operator on the left, who asks its agent to find an ‘optimal’ rotation. The barge agent starts to communicate with the terminal agents about convenient time slots. Once it has determined an optimal rotation (based on the information provided by the terminal agents), it confirms its arrival moment to all the terminal agents and communicates the resulting rotation to the barge operator. The terminal agents update their schedules based on the decision of the barge operator and communicate the new schedules with the corresponding terminal operators.

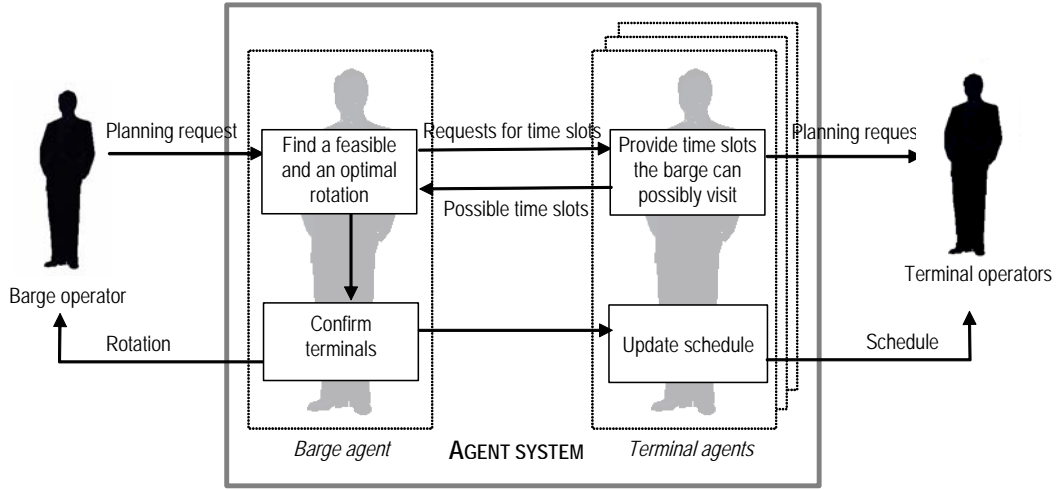


Figure 1: *Illustration of the communication in the multi agent system*

3.2 Performance indicators

Typical for multi agent systems is the existence of players with different (conflicting) interests. To reveal the main interests of both barge and terminal operators in-depth interviews were conducted [7]. The interviews revealed that the main interest of barge operators is to stick to the sailing schedule as much as possible. The sailing schedule is made by the barge operator about once a year and determines at what time a barge should be present at certain locations. Ocean carriers use this sailing schedule to contract barge operators for transportation. Reliable schedules are therefore important. The main interest of the terminal operator is to use its quays as efficiently as possible, i.e., to maximize the productivity of personnel, cranes and quays. Both barge and terminal operators have more specific secondary objectives, but for simplicity we omit these.

To evaluate the system performance we defined the following key performance indicators. These indicators abstract from the individual interests and measure the ‘group’ performance, i.e., the average performance of barge operators and terminal operators. In our model terminals have fixed capacity, which means that the utilization degree is determined by the number of barges that arrive in the port. We therefore focus on barge performance, which reflects the terminal performance as well. The more efficient terminals operate their quays, the smaller the waiting times at terminals and the shorter barges sojourn in the port. The key indicators are:

1. *Fraction of barges leaving the port too late.* Leaving the port too late means that a barge leaves the port later than the time set in its sailing schedule. This indicator measures the

fraction of barges that did not manage to leave the port *in time*.

2. *Average lateness*. To get an impression how ‘early’ or ‘late’ the barges leave the port, we measure the average lateness of all barges. Lateness is the difference between the actual time the barge leaves the port and the time set in the sailing schedule. Lateness can be negative.
3. *Average tardiness*. To evaluate the average ‘misery’, we measure the average tardiness of all barges. The tardiness of a barge is equal to the maximum of zero and the lateness of the barge.

For the off-line benchmark, i.e., the situation in which we know all information in advance, we took the indicator 3 as primary objective and indicator 1 as secondary objective. Aim of the off-line benchmark is to get an impression of the solution quality of the multi agent approach. Since it is hard to define one single objective for the whole system (given the different conflicting interests) we feel that these objectives are reasonable, since they optimize the performance of all barges and terminals, regardless of the specific individual performance.

3.3 Levels of information exchange

Recall that players in our problem are very reluctant to share information, to prevent deterioration of their competitive position. However, at least some information needs to be shared to establish a form of coordination in the network. To evaluate the value of sharing (more) information, we consider three levels of information exchange. The results can help the players in the network to make a decision about the information to be revealed, compared to the extent they think their competitive position is threatened. We assume that the communication between a barge and a terminal is done in real-time, i.e., that decisions by barges and terminals are made in a split second.

We consider three levels of information exchange:

1. *No information exchange at all*. A barge simply sails the shortest path through the network and only notifies the concerning terminals *that* it will arrive, but not *when*. As extension barges could include historical information about waiting and opening times while planning their rotation.
2. *Yes/No*. A barge can ask a terminal whether a certain arrival time is convenient and the terminal only replies yes or no. The barge continues until it has a yes from all terminals

it has to visit. To find a feasible rotation, the barge either delays terminal visits or swaps terminal visits. This is more or less the way barges currently plan a rotation. They have very limited insight in the occupation of the terminals during the day.

3. *Waiting profiles.* Terminals give barges information about the maximum amount of time a barge has to wait until its processing is started after it has arrived. This information is provided for every possible arrival moment during a certain time horizon. After the barge operator has determined the rotation with the smallest sojourn time, the arrival times are communicated to the terminals. Section 3.5.3 explains how waiting times are calculated. In fact, the waiting time expresses the time till the next moment the processing of a barge can be started.

The option of issuing time windows is not considered separately, since it is included in option 3. We first explain the concept of waiting profile and next show the advantages of communicating waiting profiles. We define a *waiting profile* of terminal i as a t -parameter family of pairs (t, w_{it}) , where w_{it} is the maximum waiting time when the barge arrives at time t at terminal i , for all t during a certain time period $[0, T]$.

A waiting profile is generated by a terminal after the request of a barge and is barge specific. To calculate this profile, the terminal operator needs to know the number of containers that has to be loaded and unloaded. Based on this information, the terminal considers its current quay schedule and determines all possible time intervals in which the handling of the barge could be started without violating appointments made with other barges. The available time intervals during a certain time period are translated to a waiting profile (in Section 3.5.3 we explain this in more detail). The maximum waiting times in the waiting profile are *guaranteed* maximum waiting times. This is a service to the barges to accurately estimate the arrival time at the next terminal. The barge in return needs to be at the terminal at the announced time, otherwise it has to make a new appointment and builds up a bad reputation which can be used by a terminal as input for the generation of the waiting profiles. This means that barges are more or less forced to arrive timely at the terminal. In this way we decouple the operations of terminals. We assume that the terminal operator includes all possible starting times of the barge handling in its waiting profile. In a later study we also aim to include a kind of look-ahead and the reputation of a barge in the generation of the waiting profiles.

Issuing waiting profiles is attractive for both terminals and barges. First, terminals can indicate

preferred handling times by varying the waiting times during the day, by increasing the maximum waiting time during periods that are expected to be busy. This makes waiting profiles different from time windows, since the latter only indicate whether the processing of a barge can be started or not. Moreover, a terminal can increase the maximum waiting time a bit to create flexibility in its quay schedule. In this way, a terminal may be able to increase its utilization, since it has more possibilities to schedule the barges with which it has already agreed a maximum waiting time. Second, barges have information about maximum waiting times (which are guaranteed) and can determine which sequence of terminal visits minimizes the sojourn time in the port.

3.4 Barge operator agent

In our model, the barge operator agent has to decide in which sequence the barge is going to visit the terminals. The quality of the decision depends, of course, on the information provided by the terminal. We describe the mathematical model and propose a way the barge operator agent makes decisions based on the available information. With respect to the barges, we do not consider barge capacity constraints, stowage plans (the way the containers are stacked on the barge), and closing times of containers (specified times certain containers need to be at the terminal). According to people in practice, the capacity and stowage constraints are not really restrictive for the planning of the rotation. Most restrictive is the first terminal the barge visits, because it first needs to unload containers before it can pick-up new ones. We do not take this into consideration. Closing times of containers introduce a specific constraint on the rotation of the barge. For simplicity we omit these.

3.4.1 Model and notation

The barge operator agent has to make two decisions, namely i) in which sequence the barge visits the terminals and ii) the specific time each terminal is visited. We assume that every barge operator agent makes these decisions at the moment the concerning barge enters the port. Since we consider barges that arrive over time, the different barge operator agents make their decisions successively at distinct times. We assume that the information of the terminals is reliable and does not change during the time the barge operator agent needs to make its decisions. Important here is that decisions of barges can be made in real time.

Let us consider one specific barge entering the port. The barge operator agent of this barge is assumed to have the following information. First, it knows the set \mathcal{N} of terminals that have to be

visited by the barge. This set is a subset of all the terminals in the port. For every $i \in \mathcal{N}$ the agent knows how much handling time h_i is needed at the terminal. The handling time consist of the time needed for loading and unloading of containers, as well as the mooring time on arrival and departure. Besides, the agent knows the sailing times s_{ij} between every pair of terminals (i, j) , where $i, j \in \mathcal{N}$. We assume both the travel and the handling time to be deterministic. Every barge enters and leaves the port via the *port entrance and exit point*, which is a specific point barges have to pass to access the port.

The aim of the barge is to finish all the activities within a certain time window and to leave the port before the end of this time window. In this study we have defined the objective of the barge to finish all activities in the port as soon as possible, although in practice there is usually a trade-off between fuel costs and waiting time. The extent to which a barge operator agent can realize this objective depends both on the actual state of the port and the information it has about this. Assume that the barge operator agent knows the maximum waiting times w_{it} , for every terminal $i \in \mathcal{N}$ and every arrival moment t . The barge operator assumes that has wait the maximum waiting time at a terminal. We can now model the barge operator agent by a *time dependent traveling salesman problem* (TDTSP). We define the time dependent travel time $\tau_{ij}(d_i)$, as the sum of the sailing, handling and maximum waiting time going from terminal i to j , where the barge departs at time d_i at terminal i . Let the arrival time a_j at terminal j be defined as $a_j(d_i) = d_i + s_{ij}$ then the time dependent travel time $\tau_{ij}(d_i)$ is given by:

$$\tau_{ij}(d_i) = s_{ij} + w_{ja_j(d_i)} + h_j$$

So, $\tau_{ij}(d_i)$ gives the maximum amount of time between leaving i and leaving j . Aim of the barge is to find a rotation along all terminals it has to visit, such that the sum of the time dependent travel times is minimized. We assume that a barge visits a terminal only once and that the handling process is never interrupted after the handling has started.

3.4.2 Algorithms applied to solve the model

Several studies have been devoted to the TDTSP and TDVRP in the last decades. We mention especially the contributions of [12][6][9]. For a more extensive literature review, we refer to these papers as well. The three mentioned papers each follow a different approach to solve the TDTSP or TDVRP. In [12] Malandraki et al. present a heuristic based on Dynamic Programming. This

heuristic solves relatively small instances near optimally, under the assumption that the *non-passing property* holds with respect to travel times. Non-passing means that earlier departure from the origin node guarantees earlier arrival at the destination node [1]. [6] presents an extension of the Clark-and-Wright algorithm using a modified savings formula (based on the work of [14]). They repeat their construction algorithms for different combinations of parameters in the savings formula and use a 2-opt procedure after each construction to improve the route. Finally, [9] presents (among others) a parallel tabu search algorithm and show the performance of this algorithm on several test instances.

An important constraint in our model is that the TDTSP needs to be solved in real time. We defined this (on a Pentium IV 3.4 GHz dual core computer), as less than 200ms. For that reason we did not consider the algorithm of [9], but first implemented a savings algorithm with the following savings formula:

$$\phi_{ij} = \tau_{i0}(d_i) + \tau_{0j}(d_0) - g \cdot \tau_{ij}(d_i) + f \cdot |\tau_{i0}(d_i) - \tau_{0j}(d_0)|$$

and we constructed several routes for $g \in [0, 3]$ and $f \in [0, 1]$ (see [14]). Every construction is followed by a 2-opt procedure with 1000 iterations. To compare the performance of this heuristic we also implemented a depth-first branch-and-bound algorithm (B&B). However, from preliminary tests we did two observations. First, it turned out that we could solve the TDTSP optimally using B&B within 100ms for instances with up to nine to ten terminals. Second, the savings based algorithm resulted in solutions with an optimality gap varying from 10-40%.

In the port of Rotterdam, barges visit at most twenty terminals, which means that we need an algorithm that is able to solve instances with up to about twenty terminals. We therefore implemented the DP heuristic of Malandraki. In the DP heuristic of Malandraki at most H best partial tours are retained in every stage. We compared the algorithm with an exact branch-and-bound algorithm for different experimental settings. In the experiments we varied the network size (up to fifteen terminals), where terminals are clustered in one to three regions. Waiting times are randomly generated for different average waiting times. In the DP heuristic we retained the best $H=1000$ partial tours. From the results we conclude the following:

- The DP-heuristic of Malandraki yields a solution within 1% of the optimum within 200ms for all experimental settings we considered.

- For instances up to 10 terminals, depth-first branch-and-bound is faster than the DP heuristic of Malandraki.

We therefore use depth-first branch-and-bound for instances with at most nine terminals and the DP-heuristic of Malandraki for instances with more than nine terminals.

3.4.3 How decisions are made by the barge operator

In our experiments, we compare several levels of information exchange by terminals and we implement this as follows:

1. *No information exchange at all.* We assume that the barge operator has no information about the state of the network and plans a rotation only based on the sailing time between terminals. We still apply the same algorithms as described before (Section 3.4.2), but now the waiting time w_{it} is equal to zero at every terminal $i \in N$ and departure time t .
2. *Yes/No.* In this option the barge operator has no information about the state of the network as in option 1. Every time a barge arrives in the port it solves a TSP and remembers the ten best solutions. Then it starts to settle the first rotation (sequence of terminals) by asking the terminals successively whether a certain arrival time is convenient. It continues until all terminals have replied ‘yes’ in response to the barge. If the rotation can be finished within the time window set by the sailing schedule, the barge sticks to this rotation. If the rotation leads to a delayed departure from the port, it cancels all appointments made for the previous rotation and tries to settle the next rotation by asking terminals successively for convenient times. These steps are repeated until either a rotation can be finished in the time window set by the sailing schedule or it has considered the last rotation (the tenth). In that case, the barge sticks to the rotation, whether it leads to a delayed departure from the port or not. This approach is similar to the model used in a previous study [3].
3. *Waiting profiles.* We assume that the barge operator has information about guaranteed maximum waiting times at all terminals $i \in N$ for an infinite horizon. The barge operator uses this information to solve the TDTSP, assuming that the barge has to wait the (maximum waiting) time indicated by the terminal.

In the experiments we evaluate the three different levels of information exchange in detail.

3.5 Terminal operator agent

In our model, the terminal operator has to decide about convenient times barges can be processed. The quality of the decision depends on the information the terminal operator has about future barge arrivals. We assume that terminal operators lack this information for barges that did not enter the port. We describe the mathematical model and present the way terminal operators make their decisions. All terminals in our model are opened 24h a day.

3.5.1 Model and notation

Every terminal in the port has a terminal operator agent which has to negotiate with barges about convenient handling times. Every terminal has a set of berth-crane-team combinations Q . Every $q \in Q$ is a combination of resources which are all necessary to handle a barge. A barge is assigned to one $q \in Q$ and every $q \in Q$ processes at most one barge. We assume that the handling time of each container is deterministic.

The terminal operator agent has two tasks. First, it has to make appointments and second, it has to keep its appointments. Making appointments concerns the generation of the waiting profile, indicating the preferred handling times of the barge. Keeping appointments is about scheduling the barges with which already appointments are made, such that no appointment is violated. We discuss these tasks successively.

3.5.2 Keeping appointments

The keeping appointments task is the scheduling of *expected barges* on the quays such no appointment is violated. Expected barges are barges that already made an appointment with the terminal and still need to be processed. From time to time, it might be beneficial for a terminal to reschedule barges on the quays, e.g., when a barge arrives earlier than its expected arrival time. This is possible when a barge does not need to wait the maximum waiting time at previous terminal(s).

Information about expected barges is collected by making the appointments with these barges. When an appointment is made, the barge is added to the list of expected barges. The terminal registers the EAT, the agreed maximum waiting time (MWT) and the processing time (PT) for this barge. The PT is equal to the handling time (h_i) of the barge. To schedule the expected barges, the terminal has to decide how the barges are assigned to the available quays, such that all appointments are met, i.e., the handling of no barge is started later than its latest starting

time $LST = EAT + MWT$. Starting the handling of a barge later than its LST is not permitted in our model.

To assign the expected barges to the available quay-crane-team combinations of the terminal we applied list-scheduling in combination with depth-first branch-and-bound (see [16]). We define the *terminal lateness* of a barge as the planned starting time of this barge minus its latest starting time. The primary scheduling objective is to minimize the maximum terminal lateness (which needs to be less than or equal to zero) and the secondary objective is minimizing the average terminal lateness of all expected barges. A good initial solution for the branch-and-bound algorithm is usually to sort barges based on latest starting time (LST), which results in a strong bound and makes the algorithm very fast.

3.5.3 Making appointments: how to construct the terminal waiting profile

When a barge asks for waiting times, the terminal operator makes a waiting profile expressing the *maximum* waiting time until the handling of the barge is started, for every arrival moment during a certain planning horizon. We assume that the terminal only deals with other barges (no sea vessels et cetera). The determination of the waiting profile consists of several steps. We first make a waiting profile for each quay and then combine the quay waiting profiles to a terminal waiting profile.

Waiting profile for one quay Let us first consider the generation of a waiting profile for a single quay. Assume that we have a schedule and that we have to make a waiting profile for barge b with expected processing time PT. To explain our approach, we use the example data in Table 1. Assume in our example that barge b has an PT equal to 15. Remark that the PST (planned starting time) and the EDT (expected departure time) are not communicated with the barge, unless the handling of the barge is started.

Barge	<i>Agreements</i>		Planned starting time (PST)	<i>Schedule</i>	
	Expected arrival time (EAT)	Latest starting time (LST)		Processing time (PT)	Expected departure time (EDT)
B1	10	20	10	10	20
B2	30	40	30	10	40

Table 1: Example of a quay schedule and the corresponding agreements

To determine the quay waiting profile, we first determine all start intervals in the current quay schedule. A start interval is defined as a time interval in which the handling of the concerning

barge could be started immediately, without violating any appointments made with other barges. The start intervals are determined by considering all possible insertion points i in the schedule. Insertion point i means insertion after the i^{th} barge in the schedule ($i = 0$ means insertion before all scheduled barges). Every insertion point is evaluated as follows:

1. Consider all barges that are already planned at the quay. Replan barges before the insertion point as early as possible and barges after the insertion point as late as possible. Remark that we do not re-sequence the planned barges. The resulting schedule must be such that no appointment is violated, i.e., no barge is planned to be started later than its latest starting time.
2. The starting time of the resulting start interval is equal to the EDT of the last planned barge before insertion point i . If $i = 0$, the starting time is equal to the actual time.
3. The end time of the start interval is equal to the PST of the first planned barge after insertion point i , minus the processing time of barge b (recall that a gap represents possible starting times of the barge). If i is equal to the number of planned barges, the end time of the start interval is set to infinity.
4. Each feasible start interval is added to the list of start intervals. For every interval we also remember the corresponding insertion point in the schedule, to be able to quickly insert the barge after it has announced its arrival time.

Repeating the steps for all possible insertion points, results in a list with start intervals ordered on start time. Table 2 presents the start intervals for the example schedule of Table 1.

Start interval	Start time	End time	Insertion point
1	0	5	0
2	20	25	1
3	40	∞	2

Table 2: *The start intervals in the example*

Once we have determined the start intervals, we need to evaluate whether the start intervals are distinct. If the end time of start interval i is larger than the start time of start interval $i + 1$, we set the end time of start interval i equal to the starting time of start interval $i + 1$. We cannot make one start interval out of these two start intervals, since they correspond with different insertion points in the schedule.

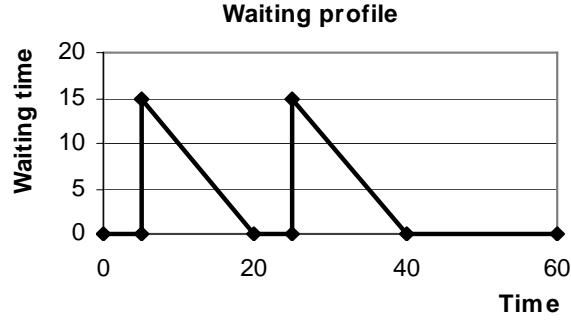


Figure 2: Example waiting profile for a single quay

To determine the waiting time profile we evaluate, starting from the current time t_0 , for all possible arrival times t in the time interval $[t_0, \infty)$ the corresponding waiting time. We do this as follows. If arrival time t is one of the start intervals, the corresponding waiting time $w_t = 0$. If arrival time t is not element of in one of the start intervals, the corresponding waiting time w_t is equal to the time until the start time of the first start interval after t . The latter means that waiting times are linear functions in time, which means that we only need to evaluate the times $t \in [t_0, \infty)$ at which the waiting times change. Changes correspond directly with the start times of all start intervals, including the actual time t_0 . Considering our example we denote the resulting waiting profile in Table 3, where $t_0 = 0$.

Time	Waiting time	Insertion point
0	0	0
5	15	1
25	15	2

Table 3: Data representation of a waiting profile

The waiting times are given by linear functions with slope -1 and a minimum of zero. See Figure 2 for a visual representation.

Waiting profile for the terminal To determine the waiting profile for the terminal we take for every time $t \in [t_0, \infty)$ the minimum of the waiting times at each of the quays. To do so we only need to evaluate the times t at which the waiting time changes at one of the quays and to calculate the corresponding waiting time at the other quays. Consider the following example with two quays and two different waiting profiles (see Table 4).

<i>Quay 1</i>			<i>Quay 2</i>		
Time	Waiting time	Insertion point	Time	Waiting time	Insertion point
0	0	0	0	0	0
5	15	1	10	15	1
25	15	2	25	5	2

Table 4: *Example of a waiting profile of two quays*

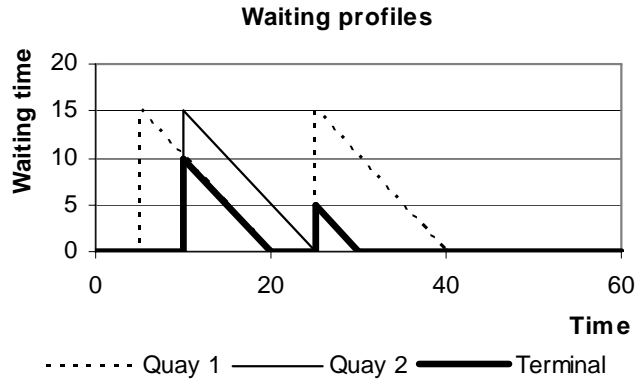


Figure 3

In the example it is clear that the waiting time changes at time $t = 0, 5, 10, 25$. Since a waiting profile can be expressed by linear functions, the resulting terminal profile can be easily obtained (see Table 5 and Figure 3).

Time	Waiting time	Quay	Insertion point
0	0	1	0
5	0	2	0
10	10	1	1
25	5	2	2

Table 5: *Data representation of the terminal waiting profile*

The minimum waiting time in the waiting profile is now equal to zero. This means that a barge can get a maximum waiting time of zero after its arrival time. The result is that the terminal operator has no flexibility in its schedule anymore, since this barge must be started at the agreed time. To increase flexibility we add some slack s to the waiting time, which means that we augment the waiting time at every time t with an amount s . In the experiments we evaluate different values of s to see the impact on the performance of both the terminals and the barges (see Section 5.1).

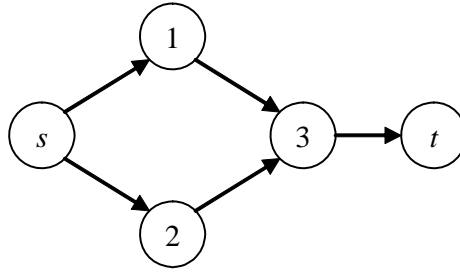


Figure 4: *Example of an activity-on-node network*

4 Centralized approach

To get an impression of the solution quality of the multi agent approach, we compare it with the situation in which we know all information in advance. We call this situation the off-line problem. We model this off-line problem as a Resource Constrained Project Scheduling Problem (RCPSP; see, for example, [5]) as a base model. In the classical RCPSP, a single project, consisting of a number of activities, has to be scheduled. To process the activities, a number of resources are available. Each resource consists of a number of parallel processors. Each activity requires during its processing a number of units of each resource (for example, 2 units from Resource 1 and 5 units of Resource 2). Between activities, precedence relations exist. An activity cannot start before all its predecessors are finished.

For solving the RCPSP, a graph representation is widely used in the literature. This graph representation is known as an ‘activity-on-node network’. See Figure 4 for the activity-on-node network for a simple project, consisting of three activities, which all have a node in the graph. There are precedence relations between Activity 1 and Activity 3, and between Activity 2 and Activity 3. This means that Activity 3 can only start when both Activity 1 and Activity 2 are finished. Besides the nodes representing an activity, there are two dummy nodes s and t . Dummy node s represents the project start and dummy node t represents the project end. If we give each node a weight that is equal to the required processing time, then the length of a longest path from s to a node i is equal to the earliest possible starting time of Activity i . The length of a longest path from s to t is equal to the earliest possible completion time for the project.

We represent the terminals by resources in the RCPSP. The number of processors for each resource is equal to the number of quays of the associated terminal. In this way, we model that a terminal can handle two barges at the same time if it has two quays. Each terminal visit of a barge is

represented as an activity. The processing time of an activity is equal to the load and unload time of the associated terminal visit.

We need to change the properties of the activity-on-node network in a number of ways to deal with characteristics that are not modelled in the basic RCPSP. For example, we give an arc from s to a node a weight that is equal to the arrival time of the associated barge in the port. In this way, the earliest possible starting time of this activity is at least equal to this arrival time. Moreover, we give an arc between two nodes (representing terminal visits of the same barge) a weight that is equal to the sailing time from the first terminal to the second.

In the basic RCPSP, the objective is to minimize the makespan, i.e., the time to complete all activities in the project. For our problem, the primary objective is minimizing the total *tardiness*. The secondary objective is to minimize the number of late barges. The tardiness of a barge is equal to its *lateness* if the barge leaves the port too late and is zero if it leaves the port in time. The lateness of a barge is the difference between the actual departure time and the departure time set in the sailing schedule. The total tardiness is the sum of the tardiness over all barges considered in the model.

To solve the resulting problems, we used an algorithm that is based on the adaptive search algorithm by [11]. In this algorithm, a (large) number of schedules is generated based on randomized priority rule scheduling. For the basic RCPSP, this algorithm finds schedules that are close to optimal very fast. We adapted the algorithm to deal with the new objective function.

5 Experimental settings

To evaluate the multi agent system for different levels of information exchange, we consider different scenarios and compare the results with the off-line scheduling algorithm. In this section, we describe the scenarios, the variables, and the parameters. Recall that all handling and sailing times are deterministic. As time unit we use minutes in our experiments.

5.1 Scenarios

We created different scenarios varying along the dimensions presented in Table 6. Remark that the utilization degree is input for the model, i.e., we generate the number of barges and terminal visits such that we obtain the desired utilization degree.

Variable	Value
Network layout	three variants (see Section 5.2)
Number of terminals per region	4 and 9
Number of quays per terminal	1 and 2
Utilization degree	50, 75, 90%
Time window barge	fixed, variable (see Section 5.4)

Table 6: *Dimensions varied in the experiments*

Every scenario is evaluated using the off-line benchmark and a simulation of the multi agent model. For the multi agent simulations, we consider the three levels of information exchange as mentioned in Section 3.3. For the waiting time implementation we also vary the value of s (the additional flexibility in the waiting profile) for $s \in \{0, 30, 60\}$, with s in minutes. Note that $s = 0$ resembles the issuing of time windows. See for an overview Table 7.

Model	Variants
Off-line benchmark	-
No information exchange	-
Yes/no communication	-
Waiting times	slack in waiting profile $s = \{0, 30, 60\}$

Table 7: *Models considered*

Remark that the result of the off-line benchmark and the yes/no information exchange, both depend on the time window the barge is expected to be in the port. For the other simulations this time window does not affect the results of the simulation, since it is not directly taken into account by the decisions of the barge and the terminal operator.

5.2 Network layouts

We consider three different network layouts (see Figure 5), which are inspired by the layout of large ports around the world (Rotterdam (layout II), Antwerp (layout III), Hamburg (layout III), Singapore (layout II), and Shanghai (layout II)). We do not claim that our networks layout fit these ports exactly, but they are reasonable variants.

We vary the number of terminals per region (either four or nine terminals) as well as the number of quays per terminal (one or two). The sailing time between two terminals only depends on the regions each of the terminal belongs to, not on the Euclidian distance. In the port it is not possible to sail straight from one terminal to another, since there are only a few connecting water ways. We therefore assume that sailing time within a region is always equal to 20 minutes. Sailing through a region (without visiting a terminal) takes 40 minutes). Sailing times between regions

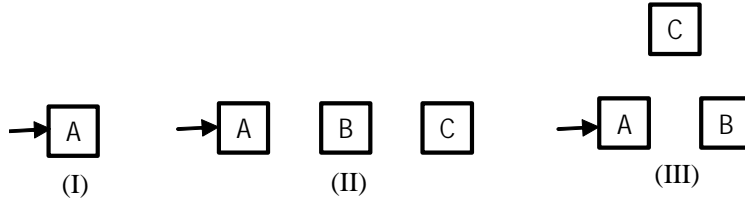


Figure 5: Three network layouts; one region, three regions in line and in a triangle. The arrows represent the port entrance and exit point.

are denoted in Table 8. So, to travel from a terminal in region A to a terminal in region B takes $20 + 80 + 20 = 120$ minutes

From/to region	Line			Triangular		
	A	B	C	A	B	C
Port entrance and exit point	0	120	240	0	120	120
Region A	0	80	200	0	80	80
Region B		0	80		0	80
Region C			0			0

Table 8: Sailing times (in minutes) between regions for different network layouts

Note that for the line network the triangle inequality does not hold at first sight. However, sailing distances are denoted among terminals. To give an example; sailing from a terminal in region A to a terminal in region C takes 240 minutes. Sailing from a terminal in region A to a terminal in region B and successively to a terminal in region C, takes two times 120 minutes.

5.3 Parameter settings and distributions

The number of barges that visit the port within the planning horizon is derived from the number of terminals in the network, the number of quays per terminal, the average utilization degree, and the average number of terminal visits in a rotation. The inter arrival time between barges is exponentially distributed with a mean depending on the number of barges that arrive at the port entrance point every hour.

Parameter	Value
Time to load or unload a container	3 min.
Mooring time	10 min.
Maximum number of terminal visits per rotation	15

Table 9: Parameter settings

The call size (sum of the containers to load and unload) at a terminal follows a normal distribution

with mean 30 containers and a standard deviation of 10 containers. The number of calls in a rotation is triangularly distributed with a minimum a , maximum b and mode c . The mode denotes the most frequent value in the distribution. The minimum a is equal to one. The maximum b is equal to the maximum number of calls in a rotation or the number of terminals in the network. Mode c is equal to $(a + b) / 2$. Other parameters are denoted in Table 9.

5.4 Time window of the barge

Most barges, sailing the river Rhine to Rotterdam, sail according to sailing schedules that are determined once a year. This means that the time a barge is supposed to be in the port is fixed, irrespective of the number of calls in the port. On the other hand, one might argue that the sojourn times of barges in the port depend also on the number of terminal visits in the rotation. We therefore choose to consider both fixed and variable time windows.

Fixed time windows are determined considering an *average barge* as follows. We assume an average number of calls and an average call size per call. We assume that this barge visits all regions in the network. Based on that, we can calculate the expected handling time (including mooring time) and sailing time. This is an estimation of the minimum time an average barge needs in the port to finish all its activities. To add slack for waiting at terminals, we multiply the sum of the handling and sailing time with some factor (1.75). This factor is chosen such that a reasonable number of barges can leave the port within their time window. The exact value of the factor is not very important in our experiments, more important is the fact that the time window of all barges has the same size (in one single experiment). The size of the time window does not depend on the utilization degree of the network.

The variable time window is calculated as follows. For every barge, we calculate the sum of the handling time (including mooring time) and the expected sailing time. The result is increased with some fixed percentage of slack and a variable percentage depending on the number of terminals in the rotation. The slack per terminal is 3% and the fixed percentage depends on the utilization of the network and is for a utilization degree of 50, 75, and 90% equal to 10, 50, and 100%, respectively. So, for a barge which has to visit eight terminals in a network with a 50% utilization degree, this means that the time window is equal to total handling and sailing time in the rotation times $1 + 10\% + 8 \cdot 3\% = 1.34$.

6 Results

In this section, we present the results and insights obtained after simulating all the experimental settings explained in Section 5. We first present some general results and insights and then focus more specifically on the different levels of information exchange and the comparison with the off-line benchmark. Finally, we take a closer look at the rotations of the specific barges and draw some conclusions.

Regarding the simulations. Every scenario is simulated for every level of information exchange, using a warm-up period of ten days (which proves to be sufficiently long) and a cool-down period of three days. The simulated time of every scenario varies from 100 to 200 days, depending on the utilization degree of the network. To calculate the results of the off-line benchmark, we took only nineteen days, with the same warm-up and cool-down period as in the simulations. The reason we did not consider the same 100-200 days as in the simulation, is that we suffer from the curse of dimensionality which leads to very large computation times.

6.1 Comparing the different levels of information exchange and the off-line benchmark

There are major differences between the different levels of information exchange with respect to the performance indicators defined in Section 3.2. For all considered scenarios we compared the performance of every level of information exchange with *no information exchange*. The differences are calculated for the percentage of barges that leave the port too late and for the average lateness of barges. Table 10 and 11 present the 95% confidence intervals of the differences specified for the different levels of information exchange, the number of quays per terminal and the two different barge time windows. Table 10 shows, e.g., that the fraction of barges leaving the port too late decreased on average with 6.5% and 9% (respectively for fixed and variable time windows) using yes/no information exchange compared to no information exchange, averaged over all considered scenarios where terminals have only one quay.

From the results it becomes clear that issuing waiting profiles performs significantly better for all scenarios than only yes/no information, which in turn performs significantly better than no information exchange in terms of percentage of barges leaving the port too late, but worse in terms of average lateness.

Information	One quay		Two quays	
	Fixed TW	Variable TW	Fixed TW	Variable TW
No information	[0; 0]	[0; 0]	[0; 0]	[0; 0]
Yes/No	[-9; -4]	[-16; -6]	[-11; -3]	[-14; -7]
Waiting profiles, $s = 0$	[-33; -12]	[-40; -23]	[-28; -14]	[-40; -21]
Waiting profiles, $s = 30$	[-41; -21]	[-56; -45]	[-48; -21]	[-56; -29]
Waiting profiles, $s = 60$	[-38; -21]	[-59; -47]	[-45; -19]	[-58; -30]

Table 10: Average difference in the percentage of barges leaving the port too late (in percent points) on a 95% confidence level for different levels of information issuing compared with no information issuing

Information	One quay		Two quays	
	Fixed TW	Variable TW	Fixed TW	Variable TW
No information	[0; 0]	[0; 0]	[0; 0]	[0; 0]
Yes/No	[-57; 108]	[-47; 86]	[-31; 242]	[-57; 154]
Waiting profiles, $s = 0$	[-1286; -639]	[-1284; -638]	[-1206; -317]	[-1204; -317]
Waiting profiles, $s = 30$	[-2013; -795]	[-2011; -794]	[-1491; -385]	[-1489; -384]
Waiting profiles, $s = 60$	[-2093; -770]	[-2091; -769]	[-1478; -351]	[-1476; -351]

Table 11: Average difference in the average lateness (in minutes) on a 95% confidence level for different levels of information issuing compared with no information issuing

To get an impression of the quality of the multi agent approaches, we make a comparison with the off-line benchmark based on a subset of all scenarios. We considered all scenarios with nine terminals per region, since these are the most complex and realistic scenarios. We analyzed the simulation results for the same set of days as was considered by the off-line benchmark. Figure 6 depicts the average tardiness and the average lateness of every simulated level of information exchange, compared to the off-line benchmark. We need to be careful to base the comparing on the fraction of barges leaving the port too late, since in a few scenarios the fraction found by the off-line benchmark was zero. It is unclear, however, whether this can be achieved easily or not. Regarding the performance of all levels of information exchange for the fraction of barges leaving the port too late, we see the same pattern as in Figure 6, which show that exchanging waiting time profiles clearly outperforms the other levels of information exchange. Taking into account that the off-line benchmark has all information in advance and weighs the interests of different barges against each other, we think that the decentralized control based on exchanging waiting profiles is very promising in the port. Not only because of the acceptability, but also considering the increase in performance that could be achieved.

If we take a closer look at the waiting profiles we observe that the benefit of a certain amount of slack added to the waiting profile clearly depends on the utilization degree of and the capacity in the network. In general we observe the following pattern: the higher the utilization degree the

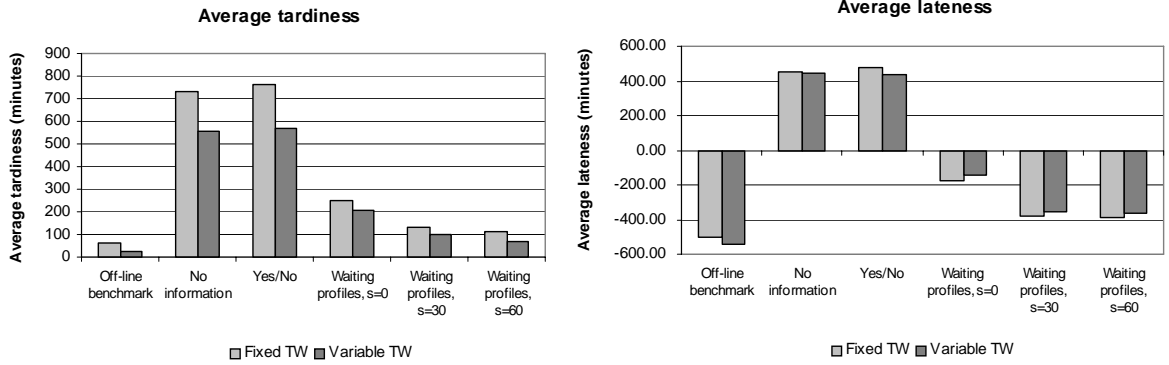


Figure 6: Difference in average tardiness and average lateness compared to the off-line benchmark. The results are averaged over all considered scenarios.

Utilization degree	Fixed TW	Variable TW
50%	0	30-60
75%	0-30	60
90%	30-60	60

Table 12: The best amount of slack in minutes (considering the percentage of barges leaving the port too late) broken down into fixed and variable time windows and the different utilization degrees considered (results are aggregated for all network layouts and number of quays per terminal)

more slack to be added to the profiles (see Table 12).

The reason more slack is needed for variable time windows compared to fixed time windows is because of the length of the time window of each barge related to the number of terminals it has to visit. Considering fixed time windows, the barges that have to visit a lot of terminals face more problems meeting the time window restrictions. Considering variable time windows, this problem is moved to the barges with short rotations, since they face on average more waiting time per terminal than barges with longer rotations (see also Section 6.2 for a detailed analysis). This means that (for variable time windows) fewer barges are delayed if the barges with short rotations face less average waiting time per terminal. This is realized by increasing slack. Increasing slack in the waiting profile means that the maximum waiting time for barges increases. Since barges optimize their rotation based on the worst case (they expect that they need to wait the maximum waiting time), the long rotations (with many terminals) become more spread in time, which leads to less dense terminal schedules, which leads to more opportunities for other barges to be processed in the short run.

With respect to the controls based on issuing waiting profiles, we see that both terminals and

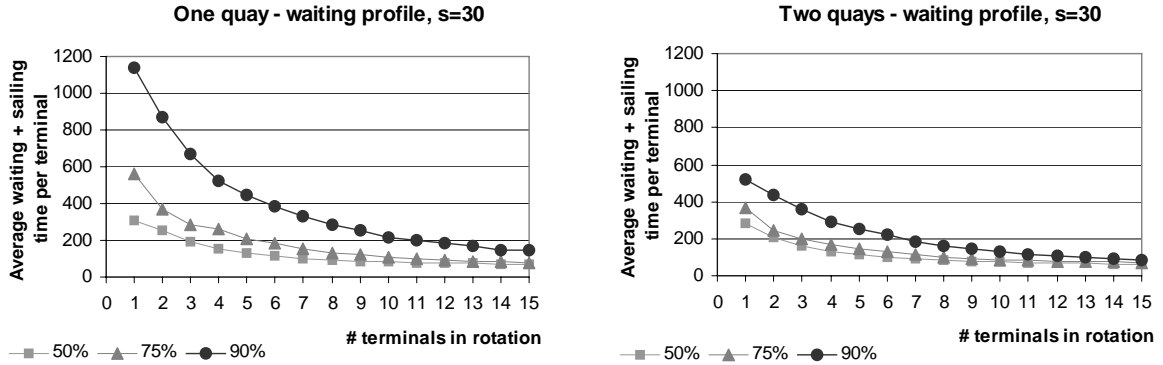


Figure 7: Average waiting plus sailing time (in minutes) at a terminal faced by barges for different rotation lengths. Results are presented for different utilization degrees for a network with three regions in a line and nine terminals per region. Results are based on issuing waiting profiles with slack $s=30$.

barges benefit from adding slack to the waiting profile. The increased flexibility gives the terminal more opportunity to dense its quay schedules, resulting in a higher quay utilization and a higher throughput. Finally, we see that the relative performance of the different controls in the network configurations ‘three networks in line’ and ‘three networks in a triangle’ are similar. However, the average performance in the ‘line network’ is slightly better than in the ‘triangle network’. This is because barges visit region B only once in the latter network and twice in the ‘line network’.

6.2 Analysis of waiting and sailing times per terminal

Considering the average waiting time of barges at every terminal, we also have to look at the sailing time of the barge. The reason is that barges sometimes prefer additional sailing time over additional waiting time, especially when the utilization degree of the network increases. Considering only the average waiting time per terminal confuses the analysis, since we cannot see to what extent waiting time is compensated by additional sailing time. We therefore analyze the average waiting plus sailing time.

The average waiting plus sailing time per terminal faced by a barge depends on the length of its rotation, i.e., the number of terminals it has to visit. Figure 7 shows that the average waiting plus sailing time per terminal decreases if a barge has to visit more terminals in the port. We see further that increasing terminal capacity reduces the average waiting plus sailing time per terminal significantly. Analysis of the results reveals that this reduction is mainly caused by a reduction in the average waiting time.

Information exchange	% barges too late		average lateness	
	Fixed TW	Variable TW	Fixed TW	Variable TW
No information	-17	-29	-814	-822
Yes/No	-18	-28	-740	-794
Waiting profiles, $s = 0$	-16	-27	-613	-621
Waiting profiles, $s = 30$	-20	-20	-360	-367
Waiting profiles, $s = 60$	-19	-19	-313	-320

Table 13: *The impact of increased terminal capacity on the fraction of barges leaving the port too late (in percent points) and the average lateness (in minutes). The results are specified for different levels of information exchange, for fixed and variable time windows, and are averaged for all scenarios considered.*

Based on Figure 7 one can imagine that for fixed time windows, especially barges with long rotations face troubles to stay within the time window. For variable time windows the barges with shorter rotations face more difficulties, since they need on average more waiting per terminal.

6.3 General results and insights

Considering the results for all scenarios and different levels of information exchange, we see that increasing the capacity of terminals (from one quay to two quays) with equal utilization degrees, has a major impact on the results. Table 13 presents the average decrease in the percentage of barges leaving the port too late and the average lateness when only the capacity of terminals is increased. The results are presented for the different controls over all scenarios.

From Table 13 it is clear that the number of quays per terminal has a significant effect on the average performance of all evaluated controls. This is because terminals have more flexibility to plan barges on the available quays.

Considering the average waiting time that successive barges face at terminals, we see (besides higher average waiting times) great fluctuations if the utilization degree of the network increases. This is to be expected considering theory about the relation between utilization degree, the lead time, and the work-in-progress level of a system [8]. When the utilization degree of the system grows, the lead time and/or the work-in-progress level will grow either. Combined with the variability in the system (different barges arriving over time) the system becomes more vulnerable which results in fluctuations in both lead times and work-in-progress levels. This means that the total amount of waiting time as part of the total sojourn time in the port increases when the utilization degree in the network grows. This is of importance in practice when barge operators set time windows for barges.

Analyzing the average sailing and waiting time per rotation, we see that barges in scenarios with high utilization degrees sometimes prefer extra sailing time over additional waiting time. As we explained before, one hour sailing is the same as one hour waiting in our model. Specifically, we see that barges sometimes sail twice from one region to another, though it results in a lot additional sailing time, their average waiting time is reduced that way.

7 Conclusions

In this paper we look at the barge rotation planning and quay scheduling problem. This problem is about the alignment of barge rotations (sequence of terminal visits) and terminal quay capacity. This alignment has to be done such that terminals achieve high quay utilization and barges leave the port according to their sailing schedule.

For this problem a central solution, with one overall supervisor that coordinates all activities in the port, is not acceptable by the parties involved. One of the reasons is that parties compete with each other and are reluctant to share information that possibly deteriorates their competitive position. We therefore apply a multi agent based approach since this enables a distributed planning where all actors involved have the freedom to make decisions which are highly in their own benefit, without exposing too much information about their operations.

We investigate three levels of information exchange varying in the extent to which information about the state of the network is revealed. Information is exchanged between an arriving barge and the terminals it has to visit. The barge uses this information to optimize its rotation. In our approach terminals do not strive for optimizing their resource utilization, but act in such a way that the barge handling process runs smoothly, i.e., by providing opportunities for the visiting barges to comply with their individual sailing schedules.

We evaluate the performance of the multi agent based controls by means of simulation. Besides, we compare the results with an off-line benchmark, which is a central optimization algorithm that assumes all information over the planning horizon to be known in advance. We consider a dynamic setting, i.e., barges arrive over time and plans are made and updated in real-time.

From the results we conclude the following. If we take into account that the off-line benchmark has all information in advance and weighs the interests of different barges against each other, we think that the decentralized control based on issuing waiting profiles is promising as a control

structure enabling an efficient barge handling process without facing the unpleasantness of sharing information which is sensitive for competitive reasons. Besides we see that issuing waiting profiles clearly outperforms control structures in which less information is exchanged. For the problem owners (the terminals and barges) this might mean a huge improvement over the current situation.

8 Further research

This research has thrown up many questions in need of further investigation. We mention some of these. First, in the presented model, the terminal operator is indifferent about which barge to process first. Suppose the terminal operator would give more priority to the barge that has the highest probability to leave the port delayed, then the results in terms of total number of barges delayed might be better. Second, we aim to consider more complex scenarios with, among others, limited opening times of terminals, sea vessels, closing times of containers (specified times at which containers need to be at a specific terminal) and unbalanced networks (terminals with different utilization degrees and different capacities). Third, we aim to apply the model to realistic data based on the port of Rotterdam. Fourth, in a future study we aim to introduce disturbances during operation which might lead to a need for e.g. replanning and robust scheduling.

Acknowledgement *This research is performed within a TRANSUMO program. TRANSUMO is a Dutch foundation which supports research to realize a transition to sustainable mobility. We are grateful to TRANSUMO for financing this research.*

References

- [1] B.H. Ahn and J.Y. Shin. Vehicle-routeing with time windows and time-varying congestion. *The Journal of the Operational Research Society*, 42(5):393–400, May 1991.
- [2] A. M. Campbell and M. Savelsbergh. Incentive schemes for attended home deliveries. *Transportation science*, 40(3):327–341, 2006.
- [3] Connekt. Eindrapport approach. participants: Port of rotterdam, initi8, illyan and the free university amsterdam. Technical report, Connekt, Delft, 2003. Report is in Dutch.
- [4] J.F. Cordeau, G. Laporte, P. Legato, and L. Moccia. Models and tabu search heuristics for the berth-allocation problem. *Transportation science*, 39(4):526–538, 2005.

- [5] E.L. Demeulemeester and W.S. Herroelen. A branch-and-bound procedure for the multiple resource-constrained project scheduling problem. *Management science*, 38(12):1803–1818, 1992.
- [6] B. Fleischmann, M. Gietz, and S. Gnutzmann. Time-varying travel times in vehicle routing. *Transportation Science*, 38(2):160–173, 2004.
- [7] L. Van Groningen. Performance measurement in barge planning. Master’s thesis, RSM Erasmus University, 2006.
- [8] W.J. Hopp and M.L. Spearman. *Factory Physics*. McGraw Hill, New York, 3 edition, 2000.
- [9] S. Ichoua, M. Gendreau, and J.Y. Potvin. Exploiting knowledge about future demands for real-time vehicle dispatching. *Transportation Science*, 40(2):211–225, 2006.
- [10] A. Imai, E. Nishimura, and S. Papadimitriou. The dynamic berth allocation problem for a container port. *Transportation research part B*, 35(4):401–417, 2001.
- [11] R. Kolisch and A. Drexel. Adaptive search for solving hard project scheduling problems. *Naval Research Logistics*, 43(1):23–40, 1996.
- [12] C. Malandraki and R.B. Dial. A restricted dynamic programming heuristic algorithm for the time dependent traveling salesman problem. *European journal of Operational Research*, 90:45–55, 1996.
- [13] M. Melis, I. Miller, M. Kentrop, B. van Eck, M. Leenaarts, M. Schut, and J. Treur. *Intelligent Logistics Concepts*, chapter Distributed Rotation Planning for Container Barges in the Port of Rotterdam, pages 101–116. Eburon Publishers, 2003.
- [14] H. Paessens. The savings algorithm for the vehicle routing problem. *European journal on Operational Research*, 34:336–344, 1988.
- [15] M. Schut, M. Kentrop, M. Leenaarts, M. Melis, and I. Miller. Approach: Decentralised rotation planning for container. In *Proceedings of the 16th European Conference on Artificial Intelligence*, Valencia, August 2004. Prestigious Applications Intelligent Systems.
- [16] J.M.J. Schutten. List scheduling revisited. *Operations Research Letters*, 18:167–170, 1996.
- [17] W. Shen and D.H. Norrie. Agent-based systems for intelligent manufacturing: A state-of-the-art survey. *Knowledge and Information Systems, an International Journal*, 1(2):129–156, 1999.

- [18] D. Steenken, S. Voß, and R. Stahlbock. Container terminal operation and operators research - a classification and literature review. *OR Spectrum*, 26:3–49, 2004.
- [19] M. Wooldridge. *An Introduction to Multi Agent Systems*. John Wiley and Sons, Ltd., 4th edition, 2005.