Transportation Research Part C 17 (2009) 133–148

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Waiting profiles: An efficient protocol for enabling distributed planning of container barge rotations along terminals in the port of Rotterdam

Albert Douma *, Marco Schutten, Peter Schuur

University of Twente, School of Management and Governance, P.O. Box 217, 7500 AE Enschede, The Netherlands

ARTICLE INFO

Article history:
Received 13 July 2007
Received in revised form 14 March 2008
Accepted 27 June 2008

Keywords:
Barge planning
Quay scheduling
Multi-agent based control
Simulation
Dynamic assignment

ABSTRACT

We consider the problem of aligning 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. 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 to 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, in spite of the limited information available, our distributed approach performs quite well when compared to the central approach. In addition, our experiments indicate that an information exchange based on waiting profiles reduces the average tardiness per barge with almost 80% when compared to the situation with no information exchange. We therefore think that waiting profiles provide a promising protocol to tackle this problem.

1. Introduction

The container 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 it handled about 6.5 million containers in 2007. Most of these containers are shipped to the hinterland by means of train, truck, and barge. Barges are self-propelled boats operated by a barge shipper. In this study, we focus on the barge container sector, which has become an increasingly important transportation mode over the past decades.

The barge rotation planning and quay scheduling problem is rather complex. In the port, there are about 30 terminals, whereas 75–100 barges visit the port daily, visiting, on average, eight terminals. Some of the terminals also handle sea going vessels and some are closed during the night, which limits the possibilities to handle barges. Terminals are not necessarily dedicated container terminals. Currently, the planning of a barge rotation, i.e., a sequence of terminal visits, is done by the barge operator that has contracted the barge. The quay planning is done by the terminal operator of the corresponding terminal. We define a quay as a combination of resources (berth, crew, crane) needed to handle a barge. On the 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 much time and due to changes and disturbances, it is often not possible to execute a barge rotation

* Corresponding author. Tel.: +31 534895480.
E-mail address: a.m.douma@utwente.nl (A.M. Douma).

0968-090X/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.trc.2008.06.003

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.

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 for 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 between 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 undermines 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. It is important that the solution is acceptable to and optimizes the operations of all the parties involved. Acceptance is possibly even more important than optimization. Optimization is a hard concept in this context, since parties have different interests and we deal with a dynamic environment, which means that decisions are made without full knowledge about the future.

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 consider only barges and terminals, which are both opportunistic, meaning that opportunities are exploited for their own benefit with no regard for the consequences for other players. Barges arrive over time with stochastic inter-arrival times. Decisions of both barges and terminals have to be made in real-time and we assume that never two barge operators plan rotations concurrently, but one after another. With respect to a terminal we assume that it handles only barges (no sea vessels), has fixed capacity, is never closed, and only has information about barges that already arrived in the port. Arrival times and other characteristics of barges that did not arrive yet in the port are unknown to the terminal. The time to handle a container, the mooring time, and the sailing time between terminals are deterministic. With respect to a barge we assume that, on arrival in the port, it needs to make a decision about the sequence in which it visits the concerned terminals. On arrival in the port, the barge has information about the terminals it has to visit, the number of container it has to (un)load at each terminal, and the mooring time at and the sailing time between terminals. It has no information about the state of the network, such as waiting times at terminals. We consider no capacity or stowage constraints for the barge. Barges visit terminals only once and preemption at a terminal is not allowed. We assume that all terminals have the same objectives and the same holds for all barges. We do not consider disturbances and the effect on the operations of barges and terminals. These assumptions allow barges and terminal operators to make reliable appointments, since no disturbances occur during operations that result in unexpected delays. Introducing disturbances require the introduction of a kind of penalty, e.g., penalty costs, if appointments are not met by one of the parties. This is part of future research.

We propose a new protocol, using waiting profiles, to support the alignment of barge and terminal operations. We introduce the idea of including *slack* in the waiting profiles to increase the planning flexibility of terminals. We compare different multi-agent based protocols and compare the results (obtained by simulation) with an off-line benchmark, which is a central optimization algorithm based on the assumption that information over the whole planning horizon is known a priori. The information concerns the arrival times of barges, the terminals each barge has to visit, the containers the barge has to (un)-load, the terminals and their capacity, and the sailing times in the network. In the multi-agent based control, this information is revealed gradually during the simulation.

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 propose a decentralized approach of the problem and explain the way we model the communication among barge and terminal operators. We describe different levels of information exchange and discuss the barge operator's and terminal operator's intelligence. In Section 4, we describe our off-line benchmark model and the algorithm used to solve the associated problem. Section 5 describes the conceptual simulation model and Section 6 the experimental settings we considered. Section 7 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 work

Except for a few earlier studies, the problem we consider has not been studied in the literature before. The most related fields are the berth allocation problem and the ship scheduling and routing problem, but both fields do not capture the characteristics of our problem. Also other related problems, like the attended home delivery problem and the hospital patient scheduling problem, show similarities but also important differences. We discuss these issues successively.

In 2003 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 (Connekt, 2003; Melis et al., 2003; Schut et al., 2004). The focus was on creating an off-line planning system, where barge rotations were planned one day in advance and not updated during execution. The main concern in the studies was to create feasible plans since this would mean a major improvement in practice. From the study it became clear that a decentralized control structure, besides the expected improvement, offers an acceptable solution for the parties involved (Moonen et al., 2007). In this paper we focus on a real-time planning system and, besides feasible plans, we aim at optimizing the operations of both barges and terminals.

A related problem is the berth allocation problem (BAP). It concerns the assignment of berths to ships (for an overview, see, e.g., Cordeau et al., 2005; Steenken et al., 2004; Stahlbock and Voss, 2008). However, there are two major reasons why the BAP is not applicable to our problem. First, in the static and dynamic 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., Imai et al., 2001; Cordeau et al., 2005). In our problem, however, arrival times of barges are uncertain and terminals have to plan their quays taking into account possible future barge arrivals. Second, the BAP considers the operations of a single terminal. In our problem we deal with multiple terminals which are mutually depending on each other due to the barges that, in general, have to visit more than one terminal. This means that the arrival time of a barge at a terminal is also a result of decisions made at other terminals.

Another related field is the ship routing and scheduling problem (SR&SP). For a recent overview we refer to Christiansen et al. (2004). SR&SPs are considered separately from similar problems in other transportation modes (like vehicle routing problems), due to the specific conditions under which ships operate (Ronen, 1983). Although our problem can be considered as an SR&SP, there are some major differences with the existing literature. First, the routing of ships is mainly along ports instead of terminals within a port. Although this seems a similar problem at a different level, the level of freedom in the sequence ports are visited might be much more limited due to the geographical dispersion than this is the case within a port. Moreover, we need to take into account the availability of berths, whereas the majority of literature in the SR&SP only focuses on the route the ship sails and assumes that (when a port or terminal is not closed) quay space is available to process the ship. Second, nearly all papers that appeared in the field of SR&SP focus on the static problem, i.e., all information is known in advance. Moreover, these problems are solved using a single objective function, which cannot be done in the problem we consider.

A third related field is the vehicle routing problem and especially the attended home delivery problem (AHDP) (Campbell and Savelsbergh, 2006), 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, Campbell and Savelsbergh (2006) consider incentive schemes to influence the preferred time windows of a customer. We consider a similar problem, however, having multiple carriers. We take explicitly into account the interest of the customer (the terminal), which might want to plan carriers (barges) close to each other in order to decrease the time they need to be 'at home'. This makes our problem different from the AHDP.

A fourth related field is the hospital patient scheduling problem (HPSP). The HPSP is about the scheduling of patients which can have multiple appointments that have to be scheduled at different resources (see, e.g., Decker and Li, 2000). Especially in the diagnostic phase, the sequence in which tests needs to be done is relatively free. This makes the HPSP comparable to the barge handling problem. However, there is a major difference with respect to the arrival time of barges and patient. In our problem, barges sail from the hinterland to the port and are not willing to delay their arrival time with days or weeks. In the HPSP, the arrival time of patient at the hospital is still variable and part of the decision at what (part of) the day appointments with the patient can be made to reduce the time the patient needs to be in the hospital.

In this paper we approach the problem using multi-agent systems (see for an introduction Wooldridge and Jennings, 1995). In an extensive survey on existing research on agent based approaches in transport logistics, Davidsson et al. (2005) state that especially applications of agents in transportation via water are scarce and most papers have focussed on the alignment of activities at a terminal. Contributions in this field are, e.g., Bürckert et al. (2000), Thurston and Hu (2002), and Franz et al. (2007). A few papers take a broader perspective, like a macroscopic study of Sinha-Ray et al. (2003) modeling the container business in the UK within the global container shipping context. We agree with Davidsson et al. (2005) that most agent based approaches (especially in the maritime industry) are not evaluated properly and comparisons with existing techniques are rare. Examples of two recent papers that apply a quantitative evaluation of their multi-agent model are Henesey et al. (2006) and Lokuge and Alahakoon (2007). Most papers stay, however, at the level of a conceptual agent model and sometimes draw conclusions about the (expected) performance of the model without presenting experimental results. Among the literature on multi-agent based approaches in transport logistics, we found no application similar to the problem we consider (except for the papers already mentioned).

For the modeling of the multi-agent based controls and the off-line benchmark, we apply algorithms obtained from the literature on traveling salesman problems and scheduling problems. In the course of this paper, we refer more specifically to literature in this field.

The contribution of this study to the existing literature is the following. First, we present a business problem where decentralized control, in contrast 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. Fourth, we provide insight in the way a decentralized system functions containing multiple competitive parties that are mutually dependent on each other. Fifth, we provide a quantitative evaluation and analysis of a multi-agent based approach and compare the results with traditional Operations Research techniques. To this end, we compare the results of the simulations with an off-line algorithm, which solves the problem centrally.

3. Decentralized approach

In this section, we present a multi-agent based approach to the barge rotation planning and quay scheduling problem. All actors in the network are opportunistic. 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 apply a *physical decomposition* of the problem, which means that agents represent entities in the physical world (Shen and Norrie, 1999). To simplify our design, we model only two types of agents: barge operator agents and terminal operator agents. The agents are competitive and have both reactive and proactive properties. Reactive are the terminal operator agents that respond to requests of barges, proactive are barge operator agents in the sense that they (can) anticipate on expected congestion in the network. All agents from the same type (like barge operator agents) have identical intelligence, meaning that every agent would make the same decision if it would be in the same situation as an arbitrary other agent of the same type.

Communication among agents can be organized in various ways (see, e.g., Wooldridge, 2005). We propose a direct communication mechanism (negotiation), which means that agents contact each other directly. The reason is that direct communication mirrors daily practice and is probably easier accepted by barge and terminal operators. The barge operator agent plays a central role in the negotiation and we distinguish two phases in its decision process. In the first phase, the barge determines a convenient rotation based on information revealed by the terminals and information it already has about the network, like sailing times. In the second phase, it makes appointments for the 'best' rotation found in the first phase. In the remainder of this paper, we discuss this in more detail. Let us clarify what we mean with an appointment.

An *appointment* made by a barge and a terminal is an agreement from two sides. The barge promises the terminal to be present at the terminal at a certain time, i.e., the latest arrival time. The terminal in turn guarantees the barge a latest starting time, if the barge keeps its promise. If the barge does not keep its promise and arrives later than the announced time, it has to make a new appointment. In making appointments, the barge uses the guaranteed latest starting times at preceding terminals.

Remark that barges in our model do not estimate the waiting time, but use the maximum waiting time to plan their rotation. This maximum waiting time is guaranteed by the terminals. Recall that no disturbances occur during operations and that sailing, mooring, and handling times are deterministic. This means that a barge is ever able to arrive in time at a terminal, i.e., not later than the agreed latest arrival time. Penalty costs are therefore no issue. In a future study it might be interesting to let barges estimate the waiting time instead of using the maximum waiting time to plan their rotation. However, this probably requires the introduction of a penalty to force parties to meet agreements.

3.2. Performance indicators

Typical for multi-agent systems is the existence of players with different (conflicting) interests. In-depth interviews with barge and terminal operators revealed that the main interest of barge operators is to stick to the sailing schedule as much as possible (Van Groningen, 2006). The sailing schedule defines on a yearly basis the specific dates, at which barge services are offered from the hinterland to Rotterdam and vice versa. Ocean carriers use the sailing schedules to contract barge 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.

In our model, terminals have fixed capacity, which means that the utilization degree of a terminal is determined by capacity requirements of the barges that visit this terminal during a certain period. To evaluate the system performance, we therefore focus on barge performance, which reflects the terminal performance as well. Remark that terminals have fixed capacity in our model. Utilization of the terminal is not a decision variable, but a result of the number of barges that arrive during a certain time period. However, the more efficient terminals operate their quays, the smaller the waiting times at terminals and the shorter barges sojourn in the port. The key system performance indicators we use are:

1. *Fraction of barges leaving the port late*. Leaving the port 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 tardiness*. We measure the average tardiness of all barges, to evaluate to which extent barges leave the port late. The tardiness of a barge is equal to the maximum of zero and the *lateness* of the barge. 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 lateness*. To get an impression how 'early' or 'late' the barges leave the port on average, we measure also the average lateness of all barges.

For the off-line benchmark, we took the minimization of indicator 2 as primary objective and the minimization of indicator 1 as secondary objective. The 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. To evaluate the value of sharing information, we consider different levels of information exchange in the first phase of the barge decision process. We define a *level of information exchange* as the extent to which a terminal gives insight to a barge in the occupation of the terminal during a certain horizon, i.e., the next couple of hours, days, weeks. The more insight a barge has in the occupation of terminals, the better it can determine a rotation minimizing its expected sojourn time in the port.

We consider three levels of information exchange:

1. *No information.* Terminals reveal no information about their occupation. The barge operator therefore determines its rotation by finding the shortest path along all the concerned terminals.
2. *Yes/no.* A barge operator can ask terminals repeatedly whether a certain arrival time is convenient and the terminal replies only yes or no. To find a convenient rotation the barge operator can delay terminal visits or visit terminals in different order. The basic idea is that barges can retrieve information about the occupation of terminals, but the information is very limited. See Section 3.4.3 for more details.
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.

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 then 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. The maximum waiting times determining 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 turn 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 future 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.

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.

We like to mention again that the levels of information exchange relate to the first phase in the decision process of the barge (see Section 3.1), where a barge determines the 'best' rotation. The second phase in the decision process is independent of the level of information exchange used in the first phase.

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. We describe the mathematical model and propose a way the barge operator agent makes decisions based on the available information. Recall that we do not consider capacity and stowage constraints.

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. 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. It is important that barges make decisions 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. 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 that both the sailing and the handling time are 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. We have therefore defined the secondary objective as minimize the total sailing time. This means that if two solutions are equal in terms of total sojourn time in the port, the solution with the least sailing time is preferred. 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 it has to 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 at terminal j be denoted by $a_j(d_i)$. Clearly, $a_j(d_i) = d_i + s_{ij}$. Furthermore, the time dependent travel time $\tau_{ij}(d_i)$ is given by:

$$\tau_{ij}(d_i) = s_{ij} + w_{j,a_j(d_i)} + h_j$$

Consequently, $\tau_{ij}(d_i)$ 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

In the last decades, several studies have been devoted to the time dependent traveling salesman problem (TDTSP) and the time dependent vehicle routing problem (TDVRP). We mention especially the contributions of Malandraki and Dial (1996), Fleischmann et al. (2004), and Ichoua et al. (2006). 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. Malandraki and Dial (1996) 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 Ahn and Shin (1991). Fleischmann et al. (2004) present an extension of the Clark-and-Wright algorithm using a modified savings formula (based on the work of Paessens, 1988). 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, Ichoua et al. (2006) present (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. For convenience, we have defined this (on a Pentium IV 3.4 GHz dual core computer) as less than 200 ms, although this is by no means restrictive. For that reason, we did not consider the algorithm of Ichoua et al. (2006), but first implemented a savings algorithm with the following savings formula (the zero denotes the depot):

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

and we constructed several routes for $g \in [0,3]$ and $f \in [0,1]$ (see Paessens, 1988). 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 about seven to eight terminals. Second, the savings-based algorithm resulted in solutions with an optimality gap varying from 10–40%, which is not satisfying.

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 and Dial (1996), where in every stage at most H best partial tours are retained. We compared the algorithm with an exact branch-and-bound algorithm for different experimental settings (for more details, see Appendix). In the experiments, we took $H = 1000$. From the results we conclude the following:

- The DP-heuristic of Malandraki and Dial (1996) yields a solution within 1% of the optimum within 200 ms for all experimental settings we considered.

- For instances up to seven to eight terminals, depth-first branch-and-bound is faster than the DP heuristic of Malandraki and Dial (1996).

We therefore use depth-first branch-and-bound for instances with at most seven terminals and the DP-heuristic of Malandraki and Dial (1996) for instances with more than seven terminals.

3.4.3. How decisions are made by the barge operator

In our experiments, we compare the levels of information exchange by terminals (see Section 3.3) and we implement this as follows:

1. *No information.* 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.* As in option 1, the barge operator has no information about the state of the network. 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 last considered 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 (Connekt, 2003) and is inspired by practice, where barge improve their rotation by contacting terminals iteratively, until they are satisfied with a certain rotation or they simply stop putting effort in finding improvements if it takes too much time.
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.

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 that 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 at a time. We assume that the handling time of each container is deterministic.

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

An appointment is an agreement between the barge and the terminal operator (see Section 3.1). By making an appointment, the barge promises the terminal to be present at a certain time, namely the *latest arrival time (LAT)*. Based on the waiting profile, issued in the first phase of the barge' decision process, the terminal derives the *maximum waiting time (MWT)* for the LAT and guarantees the barge a *latest starting time (LST)*. It holds that LST = LAT + MWT. After the appointment is made, the terminal schedules the barge tentatively such that the appointment with the barge is not violated, nor the appointments made with other barges. In the schedule (which is hidden for the barge) every barge has a *planned starting time (PST)* and an *expected departure time (EDT)*. The EDT is equal to the PST and the *processing time (PT)* of the barge. The PT is equal to the handling time (h_i) of the barge, as introduced in Section 3.4.1 and is revealed by the barge in the first phase of its decision process.

To give an illustration, consider the agreements made with barge B1 in Table 1. Barge B1 has promised to arrive not later than $t = 10$ and it has a guarantee that its processing will be started no later than time $t = 20$ (if it arrives in time). Hidden for the barge is the schedule of the terminal, denoting the PST, PT and EDT, respectively 10, 10 and 20 for this barge. Barge B2 has made similar appointments as shown in Table 1.

Table 1
Example of a quay schedule and the corresponding agreements

Barge	Agreements with the barges		Schedule (hidden for the barges)		
	Latest arrival time (LAT)	Latest starting time (LST)	Planned starting time (PST)	Processing time (PT)	Expected departure time (EDT)
B1	10	20	10	10	20
B2	30	40	30	10	40

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 (see Table 1 for an example). 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 latest arrival time. This is possible when a barge does not need to wait the maximum waiting time at previous terminal(s).

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 LST. 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 apply list-scheduling in combination with depth-first branch-and-bound (see Schutten, 1996). We define the *terminal lateness* of a barge as the planned starting time of this barge minus its latest starting time. Our 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 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. 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.

3.5.3.1. 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. In our example, barge b has a PT equal to 15.

To determine the quay waiting profile, we first determine all start intervals for barge b 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 ith barge in the schedule ($i = 0$ means insertion before all scheduled barges). At every insertion point, all barges before the insertion point are replanned as early as possible (i.e., without violating any appointment) and barges after the insertion point as late as possible. Remark that we do not resequence the barges. 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. 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. 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.

Once we have determined the start intervals, we need to evaluate whether the start intervals are disjoint. 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.

Table 2
The start intervals in the example

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

To determine the waiting 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 maximum waiting time. We do this as follows. If arrival time t is in one of the start intervals, the corresponding waiting time $w_t = 0$. If arrival time t is not element of one of the start intervals, the corresponding maximum waiting time w_t is equal to the time until the start time of the first start interval after t. The latter means that maximum 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 maximum 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 $t_0 = 0$.

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

3.5.3.2. 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 maximum waiting times at each of the quays. To do so we only need to evaluate the times t at which the maximum waiting time changes at one of the quays and to calculate the corresponding maximum waiting time at the other quays. Consider the following example with two quays and two different waiting profiles (see Table 4).

In the example it is clear that the maximum 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 easily be obtained (see Table 5 and Fig. 2).

The minimum maximum 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 can add some slack s to the maximum waiting time, which means that we augment the maximum waiting time at every time t with an amount s. Note that $s = 0$

Table 3
Data representation of a waiting profile

Time	Maximum waiting time	Insertion point
0	0	0
5	15	1
25	15	2

Fig. 1. Example waiting profile for a single quay.

Table 4
Example of a waiting profile of two quays

Quay 1			Quay 2		
Time	Max. waiting time	Insertion point	Time	Max. waiting time	Insertion point
0	0	0	0	0	0
5	15	1	10	15	1
25	15	2	25	5	2

Table 5
Data representation of the terminal waiting profile

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

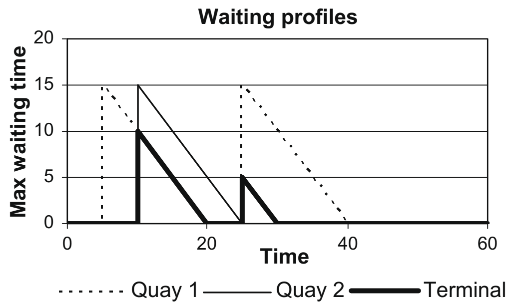

Fig. 2. The terminal waiting profile derived from the quay waiting profiles.

resembles the issuing of time windows. 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 6.1).

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, i.e., we know a priori the arrival times of barges, the terminals each barge has to visit, the containers a barge has to (un)load, the terminals and their capacity, and the sailing times in the network. We call this situation the off-line problem. We use the off-line problem to benchmark the simulation results. We model the off-line problem as a Resource Constrained Project Scheduling Problem (RCPSP; see, for example, Demeulemeester and Herroelen, 1992) 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 processors of each resource (for example, 2 processors from resource 1 and 5 processors 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 Fig. 3 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 the handling time h_i of the associated terminal visit (see Section 3.4.1).

We need to change the properties of the activity-on-node network in a number of ways to deal with characteristics that are not modeled 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 average *tardiness*. The secondary objective is to minimize the fraction of barges leaving the port late. The average tardiness is the sum of the tardiness over all barges considered in the model divided

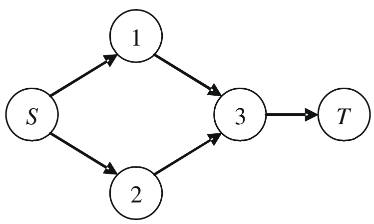

Fig. 3. Example of an activity-on-node network.

by the number of barges. Since it is possible that the total tardiness for specific scenarios is zero, we also implemented the standard RCPSP objective to the problem, i.e., the primary objective is to minimize the maximum lateness and the secondary objective is to minimize the average lateness. In fact we have two benchmarks; one based on total tardiness and one on maximum lateness.

To solve the resulting problems, we used an algorithm that is based on the adaptive search algorithm by Kolisch and Drexl (1996). 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.

5. Simulation model

Before going to the experimental settings let us briefly describe the simulator used to simulate and evaluate the performance of the multi agent system. For the simulator we apply an object oriented, discrete event simulation. The system we simulate comprises the terminals and barges. The course of the simulation is event based. This means that after an event the barge or terminal can perform an action resulting in a state transition, afterwards the state of the system remains unchanged until the next event. The state of the system can be described by the state of all the barges and terminals in the system. A barge can be in three states namely sailing, waiting, or handling. A terminal can be in two states, namely handling a barge, or being idle. The state definitions might be augmented in the future with, e.g., handling sea vessels, closing of the terminal, et cetera. Events in our model are (i) the arrival of a barge in the port, (ii) the arrival of a barge at the terminal, (iii) start handling, and (iv) finish handling. Remark that events in our model only refer to a physical change in the system, arrival of information is not seen as an event. On an event an action can be undertaken by a barge or a terminal. This requires a decision of either (or both) the barge and the terminal operator.

The decision which action has to be done by a barge or terminal is made by the barge operator and terminal operator agent respectively. These agents are modeled as decision making objects which can be invoked during the simulation. The agents can communicate with each other if necessary, e.g., to plan a rotation. To give an illustration of a decision, consider the event *terminal X finishes handling of barge A*. This is an event for terminal operator agent *X* and barge operator agent *A*. The former has to decide what terminal *X* has to do next, handle another barge (say *B*) or stay idle. The latter has to decide what barge *A* has to do next, e.g., sail to terminal *Y* or leave the port. Based on the decisions of the barge and terminal operator agents a state transition takes place and the system proceeds to the next event.

6. 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. To obtain insight in the functioning of the multi-agent system, we consider also other port configurations besides the port of Rotterdam. 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. To speed up the simulation we let the terminals reconsider their quay schedules not on every arriving barge, but after every ten arriving barges.

6.1. Scenarios

We have created 72 different scenarios varying along the dimensions presented in Table 6. Remark that the average 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.

Every scenario is evaluated using the off-line benchmark and simulations 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 profile 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.

All scenarios have a run length of 100 days. We apply a warm-up period of 10 days (which proves to be sufficiently long) and a cool-down period of 3 days. We length of these periods are determined by a graphical procedure (the method of Welch (1983)). We evaluated the development of the average lateness by taking the moving average over the successive barges until the average lateness becomes more or less stable. The length of the warm-up period is especially important for higher

Table 6
Dimensions varied in the experiments

Variable	Value
Network layout	Three variants (see Section 6.2)
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 6.4)

utilized networks. In that case it takes some time before the system is fully loaded. To calculate the results of the off-line benchmark, we take only 19 days, with the same warm-up and cool-down period as in the simulations. The reason we consider nineteen days is that this is still computationally viable to compute with the off-line benchmark, whereas the number of days that can be used for the analysis, after subtracting the warm-up and cool-down period, is still large enough.

6.2. Network layouts

We consider three different network layouts (see Fig. 4), which are inspired by the geographical structure 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 network layouts fit these ports exactly, but they are reasonable approximations. Layout I is added to evaluate the effect of regions on the performance of the system.

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 min. Sailing through a region (without visiting a terminal) takes 40 min. Sailing times between terminals are given by Table 7 on a regional level. So, from Table 7 we can see that traveling from a terminal in region A to a terminal in region C takes 240 min in a line network configuration.

6.3. 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.

The call size (sum of the containers to load and unload) at a terminal is drawn from a normal distribution with mean 30 containers and a standard deviation of 10 containers. We discretize the normal distribution by rounding to the nearest integer with a minimum value of one. The number of terminal visits (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 given in Table 8. The distributions used for the call size and the number of terminals in a rotation are chosen in the absence of real data.

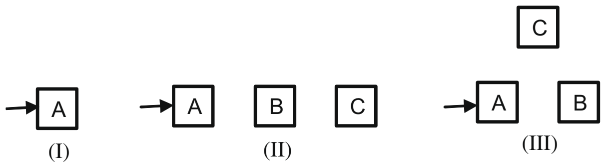

Fig. 4. Three network layouts; one region, three regions in line, and three regions in a triangle. The arrows represent the port entrance and exit point.

Table 7
Sailing times (in min) between terminals belonging to specific regions, specified for different network layouts

From/to a terminal in region	Line			Triangular		
	A	B	C	A	B	C
Port entrance and exit point	20	140	260	20	140	140
Region A	20	120	240	20	120	120
Region B	120	20	120	120	20	120
Region C	240	120	20	120	120	20

Table 8
Parameter settings

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

6.4. 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, i.e., the time window of all barges have equal length. 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 as follows. We assume an average number of calls and an average call size per call. We assume that an average 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 (1.8). This factor is determined experimentally, 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 windows of all barges have the same length (in one single experiment). The size of the time window does not depend on the chosen average utilization degree.

The *variable time windows* are 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 set to 4% 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 that 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 $1 + 10\% + 8 \cdot 4\% = 1.42$.

7. Results

If we consider the performance of different levels of information exchange, we find major differences. In Figs. 5 and 6 we show the differences in average tardiness. We averaged the results over all considered scenarios, specified for the number of quays (one and two), and the utilization degree of the terminal (50% and 90%).

The aim of Figs. 5 and 6 is not to give an estimation of the average tardiness of the barges, but to depict patterns that arise in each of the scenarios. Looking at Figs. 5 and 6, we make the following observations. First, the average tardiness highly depends on the utilization degree of the terminals. Second, the average tardiness reduces significantly when all terminals increase their capacity (from one to two quays) for the same utilization degree. Third, the effect of adding slack *s* to the waiting profiles (see Section 3.5.3) depends on the average utilization degree of the terminals.

We depicted the results for the average tardiness, but similar results are found considering the fraction of late barges and the average lateness. If we take a closer look at the results of each of the experiments, we conclude 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 late, but not necessarily in terms of average lateness or average tardiness. Besides, for the network layout II and III in Fig. 4, our experiments show that the relative performance of the different controls are similar, although 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'.

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 only the scenarios with nine terminals per region, since they are the most complex and realistic ones. From these scenarios, we considered only the ones with network layout I and II, since the results of network layout II and III are comparable. We analyzed the simulation results for the same set of days as was considered by the off-line benchmark. Fig. 7 depicts the average tardiness and Fig. 8 the average lateness of every

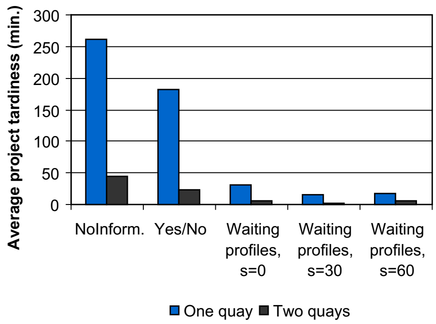

Fig. 5. The average tardiness per barge averaged over all considered scenarios, specified for the number of quays per terminal and a utilization degree of 50%.

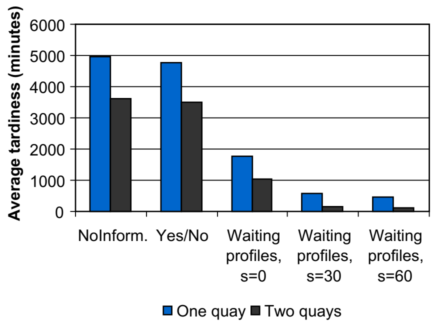

Fig. 6. The average tardiness per barge averaged over all considered scenarios, specified for the number of quays per terminal and a utilization degree of 90%.

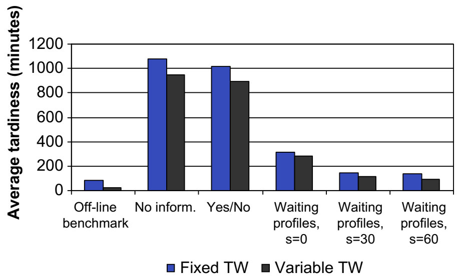

Fig. 7. Average tardiness per barge averaged over all scenarios and compared with the off-line benchmark.

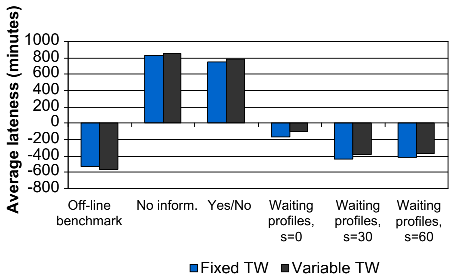

Fig. 8. Average lateness per barge averaged over all scenarios and compared with the off-line benchmark.

simulated level of information exchange, compared to the off-line benchmark. If we consider the fraction of barges leaving the port late, we find a similar pattern as in Figs. 7 and 8, namely that issuing waiting profiles clearly outperforms 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 issuing 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 more slack is to be added to the profiles (see Table 9).

Table 9
The best amount of slack in min (considering the total tardiness)

Utilization degree (%)	Fixed TW	Variable TW
50	0–30	30–60
75	30–60	60
90	60	60

We considered all 72 scenarios and specified the results for fixed and variable time windows, and the different utilization degrees considered.

The reason why 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. This means that, if the average waiting time per terminal for barges with short rotations can be reduced, the fraction of late barges reduces as well. 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 compact 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 barges benefit from adding slack to the waiting profile. The increased flexibility gives the terminal more opportunity to compact its quay schedules, resulting in a higher quay utilization and a higher throughput.

8. 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 model, terminals do not primarily 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 with stochastic inter-arrival times and plans are made and updated in real-time.

Based on the result we think that our decentralized control based on issuing waiting profiles (compared to centralized control) is promising as a control structure enabling an efficient barge handling process. We think of several reasons. First, we can reduce the communicational burden for practitioners by reducing the communication necessary to make appointments, and by speeding up the communication through automation. Second, we can significantly improve outcome of the communication. Third, when we take into account that our off-line benchmark (resembling central control) has a priori information, we see that, especially the exchange of waiting profiles, performs quite well. Besides, our experiments indicate that an information exchange based on waiting profiles reduces the average tardiness per barge with almost 80% when compared to the situation in which no information is exchanged. We therefore think that waiting profiles provide a promising protocol for our problem. For the problem owners (the terminals and barges), this might mean a huge improvement over the current situation.

9. Further research

This research has raised 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.

Acknowledgements

We thank INITI8 B.V. for providing us with the case of the barge handling and quay scheduling problem in the Port of Rotterdam and initiating the discussion of approaches to solve this problem. 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. We also thank the anonymous referees for their constructive comments and critics.

Appendix. Ad Section 3.4.2

To evaluate the performance of the DP-heuristic of Malandraki and Dial (1996), we performed several off-line experiments. For the experiments we created several instances, which are solved using the DP-heuristic of Malandraki and Dial (1996), a branch-and-bound algorithm, and the savings-based algorithms with a 2-opt procedure as described in Section 3.4.2. Every experiment is a combination of the following variables. First, the number of regions (1, 2, or 3). Second, the number of terminals per region (varying from 3 to 12) with a maximum of 12 terminals for the whole network. Third, randomly generated waiting profiles for varying average waiting times.

Terminals are positioned uniformly distributed over a square area of 60 times 60 min sailing. Distance between regions is 60 min and the regions are positioned in a line. Barges arrive with exponentially distributed inter-arrival times. Every experiment is repeated 1000 times to get statistically significant results.

The waiting profiles are created as follows. We first compute the number of times the waiting time is zero during a certain planning horizon. We do this by multiplying a sufficient long planning horizon with different values of $k \in \{0.05, 0.025, 0.017, 0.0125\}$, where k is the probability that the waiting time is zero for a certain time unit. After determining the number of times the waiting time is zero, we distribute these points uniformly over the planning horizon. At each of these points the waiting time is zero. At other points in planning horizon the waiting time is equal to the time remaining to the first point on the planning horizon where waiting time is zero. Remark that the aim of these experiments is go get a feeling for the quality of the algorithms, without giving specific performance guarantees. We are mainly interested in the performance of comparable algorithms on instances with time dependent travel times comparable to the problems barges need to solve.

References

Ahn, B., Shin, J., 1991. Vehicle-routing with time windows and time-varying congestion. The Journal of the Operational Research Society 42 (5), 393–400.
Bürckert, H., Fund, P., Vierke, G., 2000. Holonic transport scheduling with Teletruck. Applied Artificial Intelligence 14 (7), 697–725.
Campbell, A.M., Savelsbergh, M., 2006. Incentive schemes for attended home deliveries. Transportation Science 40 (3), 327–341.
Christiansen, M., Fagerholt, F., Ronen, D., 2004. Ship routing and scheduling: status and perspectives. Transportation Science 38 (1), 1–18.
Connekt, 2008. Eindrapport approach. Technical Report, Connekt, Delft (Report is in Dutch).
Cordeau, J., Laporte, G., Legato, P., Moccia, L., 2005. Models and tabu search heuristics for the berth-allocation problem. Transportation Science 39 (4), 526–538.
Davidsson, P., Henesey, L., Ramstedt, L., Törnquist, J., Wernstedt, F., 2005. An analysis of agent-based approaches to transport logistics. Transportation Research Part C 13 (4), 255–271.
Decker, K., Li, J., 2000. Coordinating mutually exclusive resources using gpgp. Autonomous Agents and Multi-Agent Systems 3 (2), 133–157.
Demeulemeester, E., Herroelen, W., 1992. A branch-and-bound procedure for the multiple resource-constrained project scheduling problem. Management Science 38 (12), 1803–1818.
Fleischmann, B., Gietz, M., Gnutzmann, S., 2004. Time-varying travel times in vehicle routing. Transportation Science 38 (2), 160–173.
Franz, T., Voss, S., Rölke, H., 2007. Market-mechanisms for integrated container terminal management. Proceedings of the 6th International Conference on Computer and IT Applications in the Maritime Industry, 234–248.
Henesey, L., Davidsson, P., Persson, J., 2006. Agent based simulation architecture for evaluating operational policies in transshipping containers. Lecture Notes in Computer Science, 73–85.
Ichoua, S., Gendreau, M., Potvin, J., 2006. Exploiting knowledge about future demands for real-time vehicle dispatching. Transportation Science 40 (2), 211–225.
Imai, A., Nishimura, E., Papadimitriou, S., 2001. The dynamic berth allocation problem for a container port. Transportation Research Part B 35 (4), 401–417.
Kolisch, R., Drexl, A., 1996. Adaptive search for solving hard project scheduling problems. Naval Research Logistics 43 (1), 23–40.
Lokuge, P., Alahakoon, D., 2007. Improving the adaptability in automated vessel scheduling in container ports using intelligent software agents. European Journal of Operational Research 177, 1985–2015.
Malandraki, C., Dial, R., 1996. A restricted dynamic programming heuristic algorithm for the time dependent traveling salesman problem. European Journal of Operational Research 90, 45–55.
Melis, M., Miller, I., Kentrop, M., van Eck, B., Leenaarts, M., Schut, M., Treur, J., 2003. Distributed rotation planning for container barges in the port of rotterdam. In: Verduijn, T., van de Loo, B. (Eds.), Intelligent Logistics Concepts. Eburon Publishers, pp. 101–116.
Moonen, H., van de Rakt, B., Miller, I., van Nunen, J., van Hillegersberg, J., 2007. Agent technology supports inter-organizational planning in the port. In: de Koster, R., Delfmann, W. (Eds.), Managing Supply Chains – Challenges and Opportunities. Copenhagen Business School Press, Copenhagen.
Paessens, H., 1988. The savings algorithm for the vehicle routing problem. European Journal of Operational Research 34, 336–344.
Ronen, D., 1983. Cargo ships routing and scheduling: survey of models and problems. European Journal of Operational Research 12, 119–126.
Schut, M., Kentrop, M, Leenaarts, M., Melis, M., Miller, I., 2004. Approach: decentralised rotation planning for container. In: Proceedings of the 16th European Conference on Artificial Intelligence. Prestigious Applications Intelligent Systems, Valencia.
Schutten, J., 1996. List scheduling revisited. Operations Research Letters 18, 167–170.
Shen, W., Norrie, D., 1999. Agent-based systems for intelligent manufacturing: a state-of-the-art survey. Knowledge and Information Systems an International Journal 1 (2), 129–156.
Sinha-Ray, P., Carter, J., Field, T., Marshall, J., Polak, J., Schumacher, K., Song, D., Woods, J., Zhang, J., 2003. Container world: Global agent-based modelling of the container transport business. In: Müller, J., Seidel, M. (Eds.), Proceedings 4th Workshop of Agent-Based Simulation.
Stahlbock, R., Voss, S., 2008. Operations research at container terminals: a literature update. OR Spectrum 30, 1–52.
Steenken, D., Voss, S., Stahlbock, R., 2004. Container terminal operation and operators research – a classification and literature review. OR Spectrum 26, 3–49.
Thurston, T., Hu, H., 2002. Distributed agent architecture for port automation. In: Proceedings of the 26th Annual International Computer Software and Applications Conference.
Van Groningen, L., 2006. Performance measurement in barge planning, Master's Thesis, RSM Erasmus University.
Welch, P., 1983. The statistical analysis of simulation results. In: The Computer Performance Modeling Handbook. Academic Press, New York.
Wooldridge, M., 2005. An Introduction to Multi-agent Systems, fourth ed. John Wiley & Sons.
Wooldridge, M., Jennings, N., 1995. Intelligent agents: theory and practice. The Knowledge Engineering Review 10 (2), 115–152.