

Aligning barge and terminal operations using service-time profiles

Albert M. Douma · Peter C. Schuur ·
J. Marco J. Schutten

© The Author(s) 2011. This article is published with open access at Springerlink.com

Abstract We consider a key issue in hinterland container navigation in ports, such as Rotterdam and Antwerp, namely the barge handling problem: how to optimize the alignment of barge and terminal operations in a port. We make a major step in solving the barge handling problem for practical settings. Specifically, we consider restricted opening times of terminals, unbalanced networks, the presence of sea vessels, and closing times of containers. Consequently, at a terminal a barge faces time dependency in: (1) the waiting time until the start of handling and (2) the handling time itself. The concept of waiting profiles which we introduced in an earlier paper only deals with (1). To deal with (1) and (2) together we introduce a more comprehensive concept, namely that of service-time profile. To establish how well our approach works, we evaluate the performance of our distributed planning approach extensively by means of simulation. We compare our results with those based on centralized planning by using an off-line benchmark resembling it. We show that the Multi-Agent system that we introduce enables barge and terminal operators to align their operations efficiently. Hence, it can be seen as a promising solution approach for solving the barge handling problem, since it enables (competing) companies to collaborate in a competitive way.

Keywords Barge handling problem · distributed planning · Multi-Agent system · simulation

A. M. Douma
Dinalog, Princenhagelaan 13, 4813 DA Breda, The Netherlands

P. C. Schuur · J. M. J. Schutten (✉)
School of Management and Governance, University of Twente,
P.O. Box 217, 7500 AE Enschede, The Netherlands
e-mail: m.schutten@utwente.nl

1 Introduction

Nowadays, the barge handling problem is a serious problem in ports like Rotterdam and Antwerp. For these ports, barges (self-propelled boats operated by a barge shipper) are an important means to transport containers between terminals in the port and the hinterland. Barges become even increasingly attractive due to congesting road and rail infrastructure and the growing attention for sustainable mobility. In the Port of Rotterdam barges visit on average eight terminals for loading and unloading containers. The sequence in which a barge visits the terminals determines to a large extent its sojourn time in the port. However, a poor alignment of barge and terminal operations is a serious problem nowadays. Currently, barge operators (companies that contract barges) and terminal operators (companies that operate a terminal) try to align their operations by making appointments using communication means such as phone, e-mail, fax, and, in case of the Port of Rotterdam, PortBase (the port community system). However, this process is communicationally demanding and results in inefficient use of quay resources and long (and uncertain) waiting times of barges at terminals, among others due to unreliable appointments, disturbances, and strategic behavior. The barge handling problem concerns the optimization of the alignment of barge and terminal operations in a port. The problem is to decide in which sequence barges visit the terminals in the port aligned with the decision on how barges are scheduled at the terminals. A solution to the problem is difficult for various reasons which we discuss below.

In the past, several solutions to the problem have been proposed. One such solution was to introduce a central (trusted) party that coordinates the activities of all barges and terminals in the port. However, both from a business and an optimization point of view this approach has some important drawbacks. From a business point of view, central coordination would require terminal and barge operators to share operational information with a central (trusted) party and to let this central party decide on their operations. This turned out to be not acceptable. The reason is that terminal operators compete with each other and so do barge operators. Both are reluctant to share information and to give up their autonomy, to avoid the risk of undermining their competitive position in the chain. From an optimization point of view, central coordination has its drawbacks, since it requires that the central trusted party weighs the (conflicting) interests of the players, which also requires the sharing of gains and losses. One can imagine that this could lead to a lot of discussion and makes the system instable. Moreover, information changes over time which makes it hard to define an ‘optimal’ solution.

A few studies therefore propose an alternative to central coordination, namely decentralized coordination by means of a Multi-Agent system (Connekt 2003; Melis et al. 2003; Schut et al. 2004). The idea is that every barge operator and every terminal operator gets a corresponding (software) agent. The agent represents its principal (the barge or terminal operator), which means that it is empowered to make decisions on behalf of its principal and to communicate with other agents about convenient handling times. Discussions with practitioners revealed that they

expect that a Multi-Agent system could offer a solution for the barge handling problem, since it does justice to the specific requirements and interests of the players concerned. However, in designing a Multi-Agent system, acceptance of the solution is at least as important as optimization of the actors' operations. These two issues (acceptance and optimization) direct the choices in the design of the Multi-Agent system. The design of a Multi-Agent system is generally about the strategy of the agents and the interaction protocol (Sandholm 1999; Wooldridge 2005; Zlotkin and Rosenschein 1996). We assume that the strategy of the agents is to behave opportunistically, meaning that they exploit opportunities for their own benefit with no regard for the consequences for other agents. This reflects the current situation in the port. We define the interaction protocol as a protocol that prescribes the way agents communicate, the content of the communication, and the aimed outcome of the communication.

In a previous paper (Douma et al. 2009), we developed a first Multi-Agent system for the barge handling problem, which serves as *base model* in this paper. The proposed interaction protocol is based on the concept of waiting profiles, which proves to be an efficient protocol to align the operations of terminals and barges in simplified port settings. The basic idea behind the concept is that terminal operators give every barge insight in the maximum amount of time it has to wait until its processing is started after it has arrived. This information is provided by means of a waiting profile, giving for every possible arrival moment during a certain time period the maximum waiting time. Based on this information, a barge operator can determine a rotation, i.e., a sequence of terminal visits, that minimizes its sojourn time in the port. The aim of the communication between a barge and a terminal operator is to make an *appointment*. Douma et al. (2009) define an appointment as follows. An appointment—made by a barge and a terminal operator—is an agreement from two sides. The barge promises the terminal to be present at the terminal before a certain time, i.e., the latest arrival time. The terminal in turn guarantees the barge a maximum waiting time until the start of processing, 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 waiting times at preceding terminals to calculate the arrival time at a succeeding terminal. Section 2 describes the base model in detail.

The base model has been evaluated for simplified port settings. This evaluation showed that the base model is a promising alternative to central coordination. In workshops with practitioners (including barge and terminal operators) we have presented the base model in a game setting (see also, Douma et al. 2008). We got positive feedback on our model. However, practitioners also expressed a need to address various practical aspects to make the system applicable in practice and to see how the system would perform in more practical settings. The following practical aspects are considered to be the most relevant:

- *Restricted opening times of terminals* Terminals can be closed during a certain period of the day
- *Sea vessels* Terminals process, besides barges, also sea vessels, which have priority over barges

- *Closing times of containers* Certain containers have a specific time at which they need to be at the terminal, e.g., because they need to be loaded on a sea vessel that is about to leave
- *Unbalanced networks* Terminals can have mutually different capacities and utilization degrees, and in a port we can have clusters of busy and quiet terminals

Addressing these four aspects means a major step towards practice. However, the question is whether and how we have to change the base model and how it will perform in more practically oriented and thus more complicated port settings. All four practical aspects do have an impact on the model and its performance. Restricted opening times of terminals imply that the handling of a barge may be interrupted by the closing of the terminal. The handling time of a barge will thus depend on the handling time windows of the terminal. The processing of sea vessels results in long periods of time during which quay resources of terminals are not available for barges. Moreover, sea vessels have absolute priority, which means that the handling of barges may not change the schedule of sea vessel handling. Both restricted opening times and sea vessels can have a major impact on the performance of barges and the way they decide upon the sequence of terminals to visit. Closing times of containers result in additional constraints for the barge operator and may limit the number of possible sequences of terminal visits. Finally, it is likely that different terminals having different capacities, opening times and other characteristics impact the performance of all barges and terminals. For instance, a cluster of terminals that are closed during the night (such as some terminals near the city in Rotterdam) will force barges to visit other terminals during the night. The question is thus how a Multi-Agent system should be designed in more practically oriented settings and whether it is still an attractive alternative to central coordination.

The goal of the paper is to provide a model that solves the barge handling problem in practical settings and leads to significant improvements of the operations of barge operators, terminal operators, and the system as a whole. In this paper we make the following contributions. First, we adapt the base model, to deal with the four practical aspects. We therefore introduce the concept of service-time profiles (as an extension of the waiting profile concept) and explain how they can be constructed dynamically and in real-time. Moreover, we change the barge operator intelligence to deal with closing times on containers. Second, we change the performance measurement to also deal with the new constraint imposed by closing times of containers. Third, we evaluate the performance for various experimental settings of our new model and compare it with central coordination.

The outline of this paper is as follows. In Sect. 2 we provide a description of the base model and our assumptions. Section 3 discusses recent work related to the barge handling problem. In Sect. 4 we introduce the concept of service-time profiles. We explain why waiting profiles do not apply to the practical settings and why the concept of service-time profiles does. We also show how service-time profiles can be constructed dynamically, based on existing terminal quay schedules and characteristics of the requesting barge. Section 5 discusses the impact of closing times on the

model. In particular, it discusses performance indicators and explains how the barge operator algorithms and our off-line benchmark have to be adjusted to properly deal with closing times of containers. To evaluate the performance of our model we perform a simulation study. In Sect. 6 we describe the experiments we perform and the results of these experiments. Finally, we draw some conclusions in Sect. 7.

2 Base model and assumptions

In this section we first describe the base model proposed in Douma et al. (2009), which serves as starting point for the model in this paper, and second we describe the assumptions we make.

The base model consists of a Multi-Agent system in which two types of decision making actors are represented, namely barge operators and terminal operators. The aim of the barge operator is to minimize the sojourn time of a barge in the port. Every time a barge enters the port, its barge operator has to determine the sequence in which terminals are going to be visited (a rotation). This decision is based on information the barge operator has (a priori) and information about waiting times at terminals. The barge operator is assumed to have (a priori) the following information: (1) the set of terminals \mathcal{N} that has to be visited, (2) the number of containers to load and unload at each terminal, and (3) the sailing time $s_{j_1j_2}$ between each pair of terminals (j_1, j_2) , where $j_1, j_2 \in \mathcal{N}$. Waiting profiles are issued by terminals on request of the barge operator. The barge operator solves a time-dependent traveling salesman problem (TDTSP—see, e.g., Malandraki and Daskin 1992) minimizing the barge's sojourn time in the port. In the TDTSP, we define the time dependent travel time $\tau_{j_1j_2}(d_{j_1})$ as the sailing time going from terminal j_1 to terminal j_2 , where d_{j_1} is the departure time at terminal j_1 , plus the handling and waiting time at terminal j_2 . Consequently, $\tau_{j_1j_2}(d_{j_1})$ gives the maximum amount of time between leaving j_1 and leaving j_2 . Note that, apart from waiting profiles, also other 'levels of information exchange' are considered, which we discuss below.

The terminal operator has to decide when a barge will or can be handled. A terminal consists of a set of berth-crane-team combinations Q . Every $q \in Q$ is called a quay and consists of a set of resources required to handle a barge. A barge is assigned to one quay and every quay handles at most one barge at a time. Terminal operators treat all barges equally, which means that a terminal accepts an appointment with a specific barge if the barge can be handled. In fact, terminal operators make appointments on a first-come first-served basis. In practice they may have different considerations, such as giving priority to barges that have agreed on arriving during specific fixed (weekly) time windows. In our model, terminal operators (re)schedule their quays using list-scheduling (see, e.g., Schutten 1996) in combination with depth-first branch-and-bound. The primary objective for the terminal for scheduling the barges is to minimize the maximum *terminal lateness* and the secondary objective is to minimize the average terminal lateness of all expected barges. We define the terminal lateness as the amount of time the barge leaves the terminal earlier or later than the time that was agreed on when the

appointment was made. In other words, the terminal plans barges as early as possible, such that the barge that is scheduled most closely to its latest departure time (LDT) still departs from the terminal before its LDT. Note that the terminal lateness always needs to be less than or equal to zero, since all appointments with barges need to be met.

Douma et al. (2009) consider three levels of information exchange, to evaluate the value of sharing information between barge and terminal operators. A *level of information exchange* is defined as the extent to which a terminal gives insight to a barge in the occupation of the terminal during a certain horizon. 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. The following three levels of information exchange are considered:

1. *No information* Terminals reveal no information about their occupation. The barge operator therefore determines its rotation by finding the shortest path along all terminals concerned.
2. *Yes/no* A barge operator can ask terminals repeatedly whether a certain start handling time is ok for the terminal and the terminal only replies yes or no. To find an acceptable 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. This level of information exchange is the most close to the current situation. We speak of an ‘acceptable’ rotation, since it is quite likely that the planned rotation is not optimal.
3. *Waiting profiles* Terminals give barges information about the maximum waiting time given a certain arrival time. This information is provided for every possible arrival moment during a certain time horizon. After the barge operator has determined its ‘best’ rotation, it communicates its arrival times to the terminals. Each such arrival time is, based on the definition of an appointment, the latest arrival time for the barge at the terminal concerned.

Note that the levels of information exchange only relate to the phase in the decision process where the barge collects information about the availability of terminals. The aimed outcome of the communication between a barge and a terminal operator is always an appointment (as defined in Sect. 1), irrespective of the level of information exchange used.

In this paper, we make the following assumptions. Barges arrive at the port over time with stochastic interarrival times. On arrival in the port, the barge operator decides which rotation a barge is going to execute. Every barge enters and leaves the port via a *port entrance and exit point*. Based on practice, we assume that every barge has a time window in which it has to complete all its activities in the port. This time window is, in practice, determined by the sailing schedule of the barge operator. We use these time windows to measure the performance of barges in the port. Decisions of barge and terminal operators have to be made in real-time and we assume that two barge operators never plan their rotation simultaneously. The reason why decisions have to be made in real-time, is to deal with the fact that information on the availability of terminals is perishable due to, e.g., actions of

other barge operators. With respect to barges, terminals only have information about barges that arrived in the port. Terminals have fixed capacity and can have restricted opening times during which barges can be processed. Terminals do not work in overtime. The time to handle a container, the mooring time and the sailing time between terminals are deterministic. We make this assumption to get a basic understanding of the functioning of the model. Arrival and handling times of sea vessels are known to the terminal prior to the planning horizon, as well as the quay where the sea vessel berths. Sea vessels have absolute priority over barges, which means that sea vessels have fixed start and completion times. With respect to a barge, we consider no capacity or stowage constraints. Barges visit terminals only once. We define the closing of the terminal as a *preemptive downtime*, and the processing of a sea-vessel and a barge as a *non-preemptive downtime*. Preemptive downtime means that the handling of a barge may start before the downtime and finish after it (Schutten 1998), i.e., the handling of the barge is preempted for closing of the terminal. Non-preemptive downtime means that the handling of a barge may not be interrupted by a downtime, i.e., the handling of a barge should be completed before the downtime starts or start after the downtime ends. In practice, terminal operators may have different objectives, but we assume them to be the same (the same holds for all barge operators). We do not consider disturbances and their effect on the operations of barges and terminals. These assumptions allow barge and terminal operators to make reliable appointments, since no unexpected delays occur. Introducing disturbances may require the introduction of, e.g., a kind of penalty if appointments are not met by one of the parties. This is part of future research.

3 Related work

In this section we discuss earlier and related studies to the barge handling problem. Additionally we introduce the concept of Multi-Agent systems and mention literature on the benchmark we use for our Multi-Agent system.

The barge handling problem already emerged in the 1980's, when the first containers were transported by barge. Through the years the problem has become more urgent due to the rapid growth of container transportation. Several attempts have been made to solve the problem. RIL Foundation (1998) was one of the first studies to investigate the bottlenecks in the barge handling process at the Port of Rotterdam. Result of the study was a covenant between barge operators and a major terminal operator in which they made agreements about the handling of barges at the terminal and about performance levels. However, this did not solve the problem. A European project (RIL Foundation 2000), called 'Barge Planning Support', investigated the added value of publishing the quay schedules of terminals on the internet. Barge operators valued this information, but terminals saw little value added from this information and considered the technical solution as too labor intensive. To evaluate whether both barges and terminals lived up to the agreements in the covenant, an application (BargePlanning.nl) was developed. This application is still in use for the planning of barges at the two major container terminals in Rotterdam. These attempts resulted in more insight in the barge handling process, but they did not provide a solution to the barge handling problem (Melis et al. 2003).

Several ideas have been proposed to solve the problem using central coordination. For reasons mentioned in Sect. 1, this turned out to be unacceptable for players. In 2003, a study was performed to investigate a distributed approach to the barge handling problem. The aim of the study—called APPROACH 1—was to investigate the possibilities of 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 was to create feasible plans, meaning a major improvement in practice. From the study it became clear that a decentralized control structure offers a promising solution to the players involved (Moonen et al. 2007). However, the study has important limitations. First, it focuses on creating feasible (not necessarily optimal) plans for all actors involved. Second, the Multi-Agent system is an off-line system and does not allow for replanning if appointments have become infeasible due to disturbances. Third, the interaction protocol results in a huge communicational burden and is not robust against strategic behavior of the actors. Regrettably, no experimental results were presented about the functioning of the Multi-Agent system and the expected improvement in practice. Based on the results with the Multi-Agent system (Connekt 2003) recommended doing research into a system which is capable to plan in real-time, to be able to deal with the dynamic nature of the problem. In 2009, a Multi-Agent system was proposed by Douma et al. (2009) to plan barge rotation and terminal capacity dynamically and with a focus on acceptance and optimization. In Sect. 1 we discussed this study and the contribution we make in this paper.

A different approach is taken by Caris et al. (2010) and Konings (2007). They investigate possibilities to solve (or at least relieve) the barge handling problem by reducing the need for a barge to visit multiple terminals in the port. To realize this, barge operators have to either consolidate (or bundle) freight for specific terminals in the hinterland or at hubs closer to the port. Currently, a barge operator has not enough volume to consolidate freight in the hinterland for one terminal and still offer frequent services between the port and the hinterland. This can only be realized when barge operators cooperate, which is not likely to happen in the short term for reasons of competitiveness. Consolidating freight at hubs closer to the port seems promising. However, whether this concept is economically viable depends on the costs of the extra handling of containers and the transport tariffs. Another development are extended gate concepts initiated by terminal operators (see, e.g., Notteboom 2004). However, it is expected that only a part of the hinterland container volumes can be handled via extended gates. Although these concepts might relieve the barge handling problem, both terminal and barge operators have expressed the need for a coordination system that helps them to align their operations dynamically, that improves the reliability of barge handling, and that enables them to respond quickly to events or opportunities.

The barge handling problem relates to several other problems in the literature. These are the berth allocation problem, the ship scheduling and routing problem, the attended home delivery problem, and the hospital patient scheduling problem. Although these problems have similarities to the barge handling problem, they do not fully capture the characteristics of our problem. For an extensive discussion on the differences with the barge handling problem we refer to Douma et al. (2009).

For an overview of the application of operations research to container terminals we refer to Stahlbock and Voß (2008) and Steenken et al. (2004).

In this paper we solve the barge handling problem through distributed planning. Centralized planning often fails to provide a satisfying solution, since it creates difficulty to weigh the (conflicting) interests of (competing) companies satisfactorily, requires the sharing of information about an actor's operations, and may be sensitive to information updates. We use the concept of a Multi-Agent system to implement a distributed planning approach. There are several ways to realize distributed planning. However, when looking at the system we intend to implement in practice, we find Multi-Agent systems the most appropriate and useful concept. One might debate whether the system we propose in this paper is already a Multi-Agent system, since certain aspects are not yet implemented, such as dynamically responding to disturbances or new opportunities. There are many definitions on Multi-Agent systems around in the literature which we will not discuss here. In our opinion, the system we propose is a Multi-Agent system (see also the definition of Durfee and Lesser (1989) and Jennings et al. (1998)). At least, one would agree that in the system we propose, a Multi-Agent system is present in a rudimentary form. We therefore will describe our system as a Multi-Agent system.

Let us briefly describe the concept of Multi-Agent systems and the Resource Constraint Project Scheduling problem (RCPSp) we use as a benchmark for our Multi-Agent system. A Multi-Agent system is a system in which multiple agents interact to achieve local or global goals. Every agent is a piece of software and can act autonomously to make decisions in the best interest of its principal. Multi-Agent systems are widely studied in different fields and from different perspectives. We apply Multi-Agent systems as an alternative to central coordination. Multi-Agent systems allow for distributed planning (see for an introduction Wooldridge and Jennings 1995) and may provide solutions where central coordination cannot. Moreover, Multi-agent systems (MAS) are believed to be particularly suited for decentralized systems in real-time and dynamic environments (Mes et al. 2008). For an extensive survey on existing research on agent-based approaches in transport logistics we refer to Davidsson et al. (2005).

For an off-line benchmark of our Multi-Agent system (see also Sect. 5.3), we model the barge handling problem as a Resource Constrained Project Scheduling problem (RCPSp). The RCPSp is a well-studied problem (see, e.g., Brucker et al. 1999; Demeulemeester and Herroelen 2002). In the classical RCPSp a single project has to be scheduled, which consists of a number of activities. The objective is to minimize the makespan, i.e., the time to complete all activities in a project. Although the problem is \mathcal{NP} -hard in the strong sense (Demeulemeester and Herroelen 2002), heuristics exist for finding high quality solutions (see, e.g., Kolisch and Drexel 1996).

4 Service-time profiles

In this section we extend the waiting profile concept used in the base model to the concept of a service-time profile. We successively provide a definition of a service-time profile and show how service-time profiles can be constructed dynamically based on terminal quay schedules.

4.1 The service-time profile concept

If the handling of a barge can be interrupted by a preemptive downtime, then a barge may need to stay longer at the terminal than the time required for the transshipment of containers only. This means that waiting profiles are not sufficient anymore to align barge and terminal operations. The reason is that waiting profiles provide information about the guaranteed maximum waiting time until the start of handling at a terminal, but not about the handling time itself. When barge operators want to make reliable appointments with succeeding terminals, then they need information on the handling time at the terminal. We therefore suggest that terminal operators provide *time dependent service time* information by means of a *service-time profile*. We define service time as the sum of waiting time and handling time at a terminal. Given a certain terminal, a service-time profile is defined as a function denoting the maximum service time $S(t)$ of a barge given a certain arrival time t during a time period $[0, T]$. A service-time profile can easily be generated if the terminal is not occupied. However, if the terminal has scheduled other barges as well, the service-time profile becomes complex. Note that the change from waiting to service-time profiles also implies that terminals, by making an appointment, guarantee a latest departure time instead of a maximum waiting time until the start of processing.

Let us illustrate the service-time profile concept with an example. In the example we aim to construct a service-time profile to be issued by a terminal (with a specific quay schedule) on request of a barge. In the next section we show step by step how we create the service-time profile.

The following example is used throughout the paper. Consider a specific terminal, which has already made appointments with barges B1 and B2, and let us assume for simplicity that the terminal has one quay. The terminal is closed during the interval $[30, 50]$. In Table 1 we denote the quay schedule of the terminal and the appointments made with the two barges. Let us consider the agreements made with barge B1. Barge B1, which needs a processing time (PT) equal to 15, has promised to arrive not later than $t = 5$ and it has a guarantee that its processing will be completed no later than time $t = 25$ (if it arrives in time). Hidden for the barge is the schedule of the terminal, denoting the planned start time (PST) and the expected departure time (EDT), respectively 5 and 20 for this barge. Barge B2 has made similar appointments as shown in Table 1. In the example, the slack in the appointments is 5 or 10 min. Assume that we have to make a service-time profile for

Table 1 Example of a quay schedule and the corresponding agreements

Barge	Processing time (PT)	Agreements with the barges		Schedule (hidden for the barges)	
		Latest arrival time (LAT)	Latest departure time (LAT)	Planned start time (PST)	Expected departure time (EDT)
B1	15	5	25	5	20
B2	10	55	75	55	65

barge b , with a processing time $PT = 15$, that now enters the port. The question is how this can be done. We explain this in the following section.

4.2 Construction of the service-time profiles

The construction of a service-time profile consists of two steps. In the first step we create a waiting profile and a service-time function. The second step is the merger of the waiting profile and the service-time function into a service-time profile. The reason why we use two steps is that we have to determine (1) the possible handling start times of the barge and (2) the service time given a certain start time. For the former we use (and extend) the concept of waiting profiles and for the latter we create a service-time function (as can be found by Christiansen and Fagerholt 2002). In this section we describe the two steps successively.

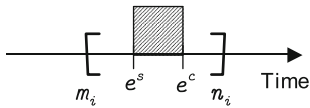
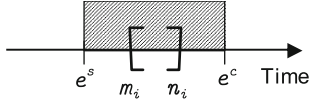
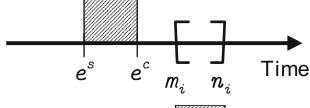
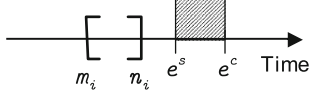
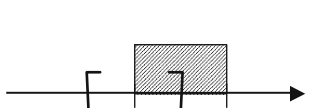
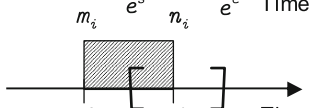
4.2.1 Construction of the waiting profile

The construction of the waiting profile depends only on the non-preemptive events, which in our model are barges and sea vessels. We apply the following procedure to construct a waiting profile that is compatible with sea vessels and restricted opening times of terminals. We start with determining all *start intervals*, by considering all possible insertion points i in the quay schedule. Insertion point i corresponds with insertion after the i th ship (barge or sea vessel) in the schedule ($i = 0$ means insertion before all scheduled ships). We define a start interval as a time interval in which the handling of the concerning barge *can be started* (not necessarily completed) such that no appointment with other barges or sea vessels is violated. Based on the start intervals we create a quay waiting profile and based on all the quay waiting profiles we construct a terminal waiting profile. This is done analogously to procedures used for the base model. In this stage we add no slack to the waiting profile. Let us describe the way the start intervals are determined in more detail.

To determine a start interval we take two steps. In the first step we consider insertion point i and we determine the corresponding start interval in the following way. We schedule all ships (barges and sea vessels) before insertion point i as early as possible, and all ships after insertion point i as late as possible (without changing the sequence of handling ships). By doing so, we take into account the restricted opening times of the terminal. Let m_i and n_i be the respective begin and end of start interval i . Then m_i becomes equal to the EDT of ship i (the last scheduled ship (barge or sea vessel) before insertion point i), and n_i is equal to the $PST_{i+1}^l - PT$, where PST_{i+1}^l denotes the planned (and latest possible) start time of the first scheduled ship after insertion point i . If $n_i < m_i$ then the start interval is not feasible.

Once we find a feasible start interval, we proceed to step two. In this step we evaluate whether the processing of a barge is interrupted by the closing of the terminal when we start the handling of a barge during start interval i . This could mean that we have to adjust start interval $[m_i, n_i]$ resulting in a new start interval $[m'_i, n'_i] \subset [m_i, n_i]$. If $n'_i < m'_i$ then the adjusted start interval is not feasible.

Table 2 Six possible elementary relations between start intervals and closing of the terminal

Possible situation	Condition	New start interval
	$m_i \leq e^s < e^c \leq n_i$	$m'_i = m_i$ and $n'_i = n_i$
	$e^s \leq m_i \leq n_i < e^c$	Infeasible interval
	$e^s < e^c \leq m_i \leq n_i$	$m'_i = m_i$ and $n'_i = n_i$
	$m_i \leq n_i < e^s < e^c$	If $e^s < n_i + PT$ then $m'_i = m_i$ and $n'_i = n_i + e^s - e^c$ If $n_i + PT \leq e^s$ then $m'_i = m_i$ and $n'_i = n_i$.
	$m_i < e^s \leq n_i < e^c$	$m'_i = m_i$ and $n'_i = n_i + e^s - e^c$
	$e^s \leq m_i < e^c \leq n_i$	$m'_i = e^c$ and $n'_i = n_i$

^a Using that $PST_{i+1}^L \notin [e^s, e^c]$ and $e^s < n_i + PT$, we find that $PST_{i+1}^L \geq e^c$ and therefore $n'_i = n_i + e^s - e^c$.

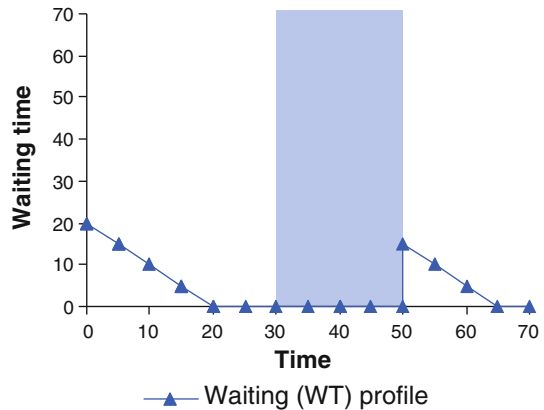
In Table 2 we give the possible elementary relations between a start interval and closing of the terminal. We denote the interval in which the terminal is closed with $[e^s, e^c]$. Note that $PST_{i+1}^L \notin [e^s, e^c]$. To give an illustration, consider the fifth situation in Table 2. We see that in this situation $m_i < e^s \leq n_i < e^c$. Assuming that we derived the start interval $[m_i, n_i]$ in the way we describe above, then $n_i = PST_{i+1}^L - PT$. However, if we start barge b in interval $\langle PST_{i+1}^L - e^c + e^s - PT, n_i \rangle$, the completion time of barge b will be greater than PST_{i+1}^L . This means that we have to decrease n_i such that $n'_i + PT + e^c - e^s \leq PST_{i+1}^L$. Rewriting the expression gives $n'_i = n_i + e^s - e^c$. This adjustment could result in $n'_i < m_i$, which is an infeasible start interval. The example is just one illustration of how a start interval has to be adapted. In practice all kind of (nested) situations can exist.

To clarify the above, let us revisit the example of Sect. 4.1. After executing the procedure described above we find the start intervals as given by Table 3. Start interval i corresponds with insertion point i . In the example it is clear that only start

Table 3 The start intervals in the example, with m_i respectively n_i the begin and end of start interval i

Start interval i	m_i	n_i	Feasible
0	0	-5	No
1	20	50	Yes
2	65	∞	Yes

Fig. 1 The waiting profile in the example



intervals 1 and 2 are feasible. Additionally, start interval 1 overlaps with the closing of the terminal (situation 1 in Table 2), which means that we possibly have to adjust the interval. However, in this situation the start interval can be maintained, since the barge handling can be completed without any problem before the next scheduled barge. During the interval the service time will vary. The resulting terminal waiting profile is depicted in Fig. 1.

4.2.2 Construction of the service-time function

The construction of the service-time function is only depending on the preemptive events, which in our model correspond with the closing of the terminal. This means that for constructing the service-time profile we assume that no other barges or sea vessels are processed at the terminal (these are non-preemptive events). Service-time functions are introduced by Christiansen and Fagerholt (2002) for the robust ship scheduling problem concerning the scheduling of port visits for a fleet of ships while minimizing the total costs. Since most terminals close in repetitive patterns, the service-time function can easily be constructed based on the opening times of the terminal. For convenience we assume that the handling of a barge never lasts longer than the time a terminal is opened per day. This is a realistic assumption. In general the service-time function looks as follows. On opening of the terminal the service time is equal to the processing time of the barge. However, when the terminal is about to close, the service time increases as the barge handling is interrupted by the closing of the terminal. Consider the example of Sect. 4.1 in

Fig. 2 The service-time (ST) function in the example

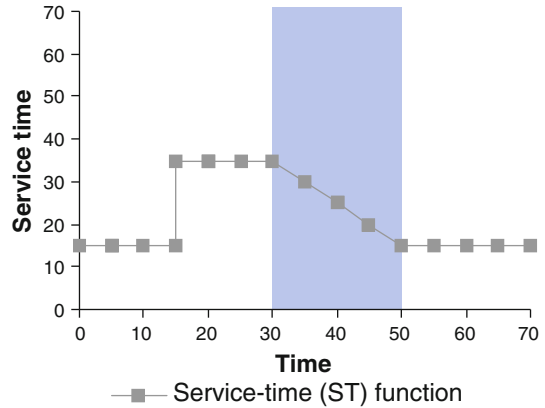


Fig. 2. In the example, the service-time function $S(t)$ is equal to $PT = 15$ for $t \in [0, 15]$. On the interval $t \in (15, 30]$ the service time is equal to 35, namely PT (15) plus the duration of the closing (20). During the interval $t \in (30, 50]$, the service time is equal to $PT + 50 - t$ and on the interval $t \in (50, 70]$ it is equal to $PT=15$ again.

4.2.3 Construction of the service-time profile

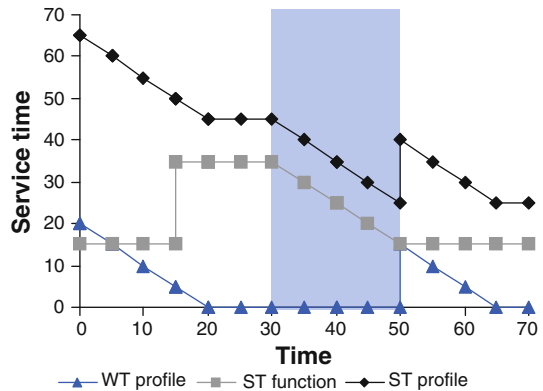
To determine the service-time profile we evaluate, starting from the current time t_0 , for all possible arrival times t in the time interval $[t_0, T]$ the corresponding maximum service time. The maximum service time $\tilde{S}(t)$ at time t can be calculated by $\tilde{S}(t) = w(t) + S(t + w(t)) + s$, with $w(t)$ the waiting time at time t given by the waiting profile, $S(t)$ the service time, given by the service-time function, and s the amount of slack that is added to the maximum service time to increase scheduling flexibility. The rationale behind this function is that a barge, arriving at time t , has a waiting time of $w(t)$ time units before it can be handled. The barge handling will therefore be started not prior to time $t + w(t)$ and its time dependent service time is then equal to $S(t + w(t))$ time units.

The service-time profile of the example presented at the start of this section is given in Fig. 3. In the example service-time profile we included 10 time units slack, which means that we augment the maximum service time at every time t with 10 min slack. Figure 3 shows both the waiting profile and the service-time function.

5 Impact of closing times on containers

The fact that containers may have closing times, creates the need to define corresponding performance indicators, to reconsider the barge operator intelligence, and to adapt our off-line benchmark model. We discuss these three issues successively.

Fig. 3 The waiting (*WT*) profile, service-time (*ST*) function, and the (*ST*) profile in the example



5.1 Performance indicators

In our study we use performance indicators to reflect the fraction of late barges and the extent to which barges are able to leave the port in time. In addition, we use a performance indicator reflecting the extent to which closing times of containers are met. The performance indicators focus on the average performance of barges and terminals, i.e., we do not consider the performance of a single barge or terminal. We assume that the capacity of terminals is fixed. This implies that the average utilization of the terminal depends on the workload of arriving barges. By processing these barges efficiently, the terminal can reduce the average waiting times at the terminal and thus influence the average sojourn time of barges in the port. However, as long as terminals have a fixed capacity and process all arriving barges, the utilization degree cannot be influenced by the terminal operator. We therefore use barge related performance indicators, since the average performance of barges also reflects the performance of the terminals. The indicators are derived from the scheduling literature and are related to the off-line benchmark (see Sect. 5.3). That is why we use the words *project* (for barge rotation) and *activity* (for the transshipment of containers at a terminal). The key performance indicators we use are:

1. *Fraction of late projects* Leaving the port late means that a barge was not able to complete its activities within the available time window. This indicator measures the fraction of barges that did not manage to leave the port in time.
2. *Average project tardiness* The project tardiness of a barge is equal to the maximum of zero and the *project lateness* of the barge. The project lateness is equal to the actual time the barge leaves the port minus the time the barge was supposed to leave the port according to its sailing schedule. Note that project lateness can be negative.
3. *Average activity tardiness* If an activity (the loading and unloading of containers at a terminal) has a due date, we also measure the activity tardiness. The activity tardiness is the maximum of zero and the activity lateness. Activity lateness is equal to the actual time an activity is completed minus the due date of that activity.

4. *Average project lateness* To get an impression how ‘early’ or ‘late’ the barges leave the port on average, we also measure the average project lateness of all barges.

Now that we have introduced the performance indicators, let us consider the barge operator intelligence.

5.2 The barge operator intelligence

The introduction of closing times of containers forces barges to weigh ‘sticking to the sailing schedule’ and ‘meeting the closing times of containers’, i.e., to weigh *project* tardiness against *activity* tardiness. In this section we discuss the barge operator objective and the way the best rotation is found. In accordance with Douma et al. (2009), we use depth-first branch-and-bound for instances with at most seven terminals and the Dynamic Programming (DP)-heuristic of Malandraki and Dial (1996) for instances with more than seven terminals. Douma et al. (2009) tested the quality of the DP-heuristic and reported solutions within 1% of the optimum within 200 ms, for different experimental settings considered. In this section we adapt (the objectives of) both algorithms to be able to deal with activity and project tardiness.

5.2.1 Objective

To deal with closing times of containers we have to adapt the objective function of the algorithms we use. Consider a specific barge for which a rotation has to be planned. Let ω_j^A denote the activity lateness of our barge at terminal j , c_j be the earliest closing time of all containers that have to be unloaded at terminal j , and d_j be the (planned) latest departure time of our barge at terminal j . If containers at terminal j have a closing time, then $\omega_j^A = d_j - c_j$, otherwise $\omega_j^A = 0$. The project lateness is denoted by ω^P and is equal to the actual time the barge leaves the port minus the planned departure time. To weigh project and activity tardiness we introduce $\gamma \geq 0$, the so-called project tardiness penalty. The primary and secondary objective functions are then equal to

$$\begin{array}{ll} \text{primary objective:} & \min \left[\gamma \cdot (\omega^P)^+ + \sum_{j \in \mathcal{N}} (\omega_j^A)^+ \right] \\ \text{secondary objective:} & \min \omega^P \end{array}$$

So, we consider all possible scenarios and select the rotation that minimizes the weighted sum of the project and the total activity tardiness. If two or more rotations are equal in this respect, we minimize the project lateness, i.e., we prefer to let the barge leave the port as soon as possible.

5.2.2 Selection criterion in the DP-heuristic

In the base model, the DP-heuristic of Malandraki and Dial (1996) is used to find the best rotation for a barge that has to visit more than seven terminals. The DP-heuristic

constructs rotations by adding to each partial tour in the previous stage a (yet unvisited) terminal, which becomes the new end terminal of the partial tour. A state is a set of terminals that are in the partial tour, and an end terminal. The value of a state is the minimum time to travel from the port entrance point, along all terminals in the partial tour and ending in the end terminal. To limit the growth of the state space and to keep the computation time of the DP-heuristic tractable, Malandraki and Dial (1996) retain in every stage the best H states, i.e., the states that most likely result in the best rotation.

To deal with container closing times, we have to adjust the DP-heuristic of Malandraki and Dial (1996), with respect to the selection of the best H states in every stage of the DP. To select the best H states, the two objectives of Sect. 5.2.1 are not sufficient. Suppose we apply these two objective functions in the selection phase, then the algorithm will add terminals with a closing time at the end of the tour. Moreover, project tardiness is (most probably) zero at the start of the rotation construction, which means that in early stages of the DP all states have similar values. This is not desirable. We therefore use a different selection criterion such that it is likely that a near-optimal solution is part of the H solutions retained in every stage.

The selection criterion we use is as follows. Suppose we are in a certain stage in the DP. Let \mathcal{N}^{int} be the set of terminals that is part of the rotation at this stage and let \mathcal{N}^{Nint} be the terminals that still need to be added. Let the current duration of the rotation be denoted by t . The activity lateness ω_j^A at terminal $j \in \mathcal{N}^{int}$ is calculated as we described above. The activity lateness $\hat{\omega}_j^A$ at terminal $j \in \mathcal{N}^{Nint}$ is equal to $t - c_j$ if one or more of the containers that have to be transhipped at terminal j has a closing time, and is zero otherwise. The selection criterion we use is then given by:

$$\text{selection criterion: } \min \left[\gamma \cdot \omega^P + \sum_{j \in \mathcal{N}^{int}} (\omega_j^A)^+ + \sum_{j \in \mathcal{N}^{Nint}} (\hat{\omega}_j^A)^+ \right]$$

By using this selection criterion we penalize postponing the visit of a terminal with a closing time to the end of the tour. In the selection phase we use project lateness instead of project tardiness, since this leads to a better distinction in the sojourn time of the port in early stages of the DP.

We have compared the performance of the adapted DP-heuristic with branch-and-bound for similar off-line experiments as Douma et al. (2009). In the experiments we used a project tardiness penalty of one. The experiments reveal that the DP-heuristic yields solutions within 2% of the optimum on average with respect to the primary objective.

5.3 The off-line benchmark

For the off-line benchmark, we model our problem as a Resource Constrained Project Scheduling Problem (RCPSP; see, for example Brucker et al. 1999; Demeulemeester and Herroelen 1992), as a base model. The off-line benchmark is an off-line model to benchmark the simulation results. The model differs from the

Multi-Agent system in two respects. First, it is a static model, i.e., it assumes that all information over the whole planning horizon is known in advance, whereas in the Multi-Agent system we assume that information is revealed over time (dynamically). Second, it solves the optimization problem for a single objective function, whereas in the Multi-Agent system the performance is the result of interacting self-interested agents. To solve the resulting problems we use a randomized construction algorithm that is based on the adaptive search algorithm (Kolisch and Drexel 1996). In this algorithm a (large) number of schedules is generated using a randomized priority rule scheduling heuristic. For the basic RCPSP this heuristic finds schedules that are close to optimal very fast.

To deal with restricted opening times of terminals and closing times of containers we have to make some adjustments to the model. Restricted opening times of terminals (resources) are represented by so-called (*quay*) *resource profiles*. A resource profile denotes the available capacity over time. Since we assume that the arrival and processing time of a sea vessel are known prior to the planning period, as well as the quay where the sea vessel berths, we include sea vessels in a quay resource profile as well. This has the advantage that sea vessels are not seen as activities and therefore do not need to be planned.

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 we are interested in minimizing the total tardiness of all projects and all activities. To achieve this, we distinguish two objectives. As primary objective we minimize the total activity tardiness augmented with a project tardiness penalty (γ) times the total project tardiness. As secondary objective we minimize the fraction of barges leaving the port late. The project tardiness penalty $\gamma \geq 0$ is used to weigh the total project tardiness compared to the total activity tardiness (see also Sect. 5.2.1).

6 Experiments and results

In this section we provide insight in the effect of the four practical aspects mentioned in Sect. 1 and the performance of the service-time profile concept compared with other levels of information exchange (see Sect. 2) and the off-line benchmark. In particular we compare the performance of the waiting profile concept with that of the service-time profile concept. In this section we describe successively the set-up of the experiments (Sect. 6.1), the general experimental settings (Sect. 6.2), and four experiments (Sect. 6.3 to 6.6).

6.1 Experimental set-up

Considering all four practical aspects at once complicates the analysis. We therefore choose to consider the effect of two practical aspects, namely restricted opening times and unbalanced network settings, in separate experiments (Experiment 1 and 2 respectively). We do not consider sea vessels in isolation, since we expect that they have a similar impact as restricted opening times. In both cases capacity is blocked for a considerable amount of time. Based on the outcomes of Experiments 1

and 2 we set-up Experiment 3 in which we consider the effect of all four practical aspects together. In Experiment 3 we consider the impact of closing times on containers by varying the project tardiness penalty. Experiments 1 to 3 are based on fictitious data inspired by the Port of Rotterdam. To see also how the service-time profile concept would perform in practice, we perform experiments based on realistic data of the Port of Rotterdam (Experiment 4).

We use simulation to evaluate the performance of the different levels of information exchange. In Experiments 3 and 4 we also make a comparison with the off-line benchmark to get insight in the relative performance of distributed planning compared to central planning.

6.2 General experimental settings

We use the following general experimental settings in the experiments based on fictitious port data. These settings are similar to the settings used in Douma et al. (2009). We consider three different port layouts (see Fig. 4) which are inspired by ports around the world, e.g., Rotterdam (layout II) and Antwerp (layout III). We vary the number of terminals per region (either four or nine). The sailing time between terminals is given in Table 4 and is based (due to the connecting waterways) on the region each of the terminals belongs to and not on the Euclidian distance.

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 barge interarrival time is exponentially distributed. The call size (number of containers to load and unload) is drawn from a normal distribution with a minimum value of one. We discretize the normal distribution by rounding to the nearest integer. The mooring time on arrival and departure at the terminal is 10 min. The time to move a container from the ship to the quay or vice versa) is 3 min. The number of terminal visits in a rotation is drawn from a triangular distribution, with a minimum of $a = 1$, a maximum b equal to the minimum of (1) the maximum number of terminal visits (equal to 15) or (2) the maximum number of terminals in the port, and a mode $c = (a + b)/2$.

As we explained in Sect. 2, barge operators determine in advance the time window in which a barge has to complete its activities in the port in order to meet the issued sailing schedules. In our experiments we consider two extremes, namely a *fixed* and a *variable* time window. A fixed time window means that all barges get the same time window irrespective of the number of terminals they have to visit (their

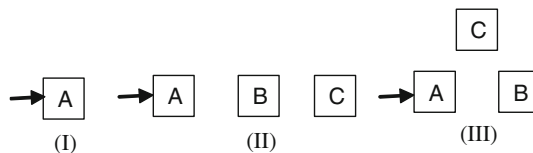


Fig. 4 Three port layouts: one region (I), three regions in *line* (II), and three regions in a *triangle* (III). The *arrows* represent the port entrance and exit point

Table 4 Sailing times (in minutes) between terminals belonging to specific regions, specified for different port layouts

From/to a terminal in region	<i>Line</i>			<i>Triangle</i>		
	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	120

rotation length). A variable time window means that the time window of a barge depends on its rotation length and is thus barge specific. For both fixed and variable time windows we add slack to compensate for waiting times in the port. The fixed time window is calculated based on the sailing and handling time an average barge needs to complete its activities in the port multiplied with a slack factor. The variable time window is calculated based on a slack per terminal (x) and a fixed percentage of slack per rotation (y) as follows. The variable time window is equal to $(1 + y + x \cdot |\mathcal{N}|)$ times the expected total handling and sailing time in the rotation, with \mathcal{N} the set of terminals to visit in the rotation. The slack factors have been determined experimentally, such that a reasonable number of barges can leave the port in time. For each of the experiments we denote the slack factors we have used.

6.3 Experiment 1: Restricted opening times

To evaluate the effect of restricted opening times we generate 36 scenarios, varying along the following dimensions: the number of terminals per region (either four or nine terminals), three different port layouts (see Fig. 4), three different utilization degrees of the network (50, 75, and 90%), and variable and fixed time windows for the barges. We successively describe the experimental settings and the results of the experiments.

6.3.1 Experimental settings

In the experiments all terminals are identical, except for the opening times. In all scenarios half of the terminals in a region (rounded down to the nearest integer) is closed during the night (6 pm–6 am) and the rest is opened 24 h a day. Terminals have one quay. The utilization degree of a terminal is only calculated based on the time the terminal is open, which implies that terminals with restricted opening times handle fewer barges per day than terminals that are opened 24 h a day. The slack factor used for the fixed time window is 1.8. The fixed slack y (slack per rotation) for the variable time window is equal to 0.5, 1.0, and 1.5 for the different utilization degrees of 50, 75, and 90% respectively, and the variable factor x (slack per terminal) is equal to 10% for all utilization degrees. We use a mean call size of 30 containers and a standard deviation of 10 containers. We evaluate the performance

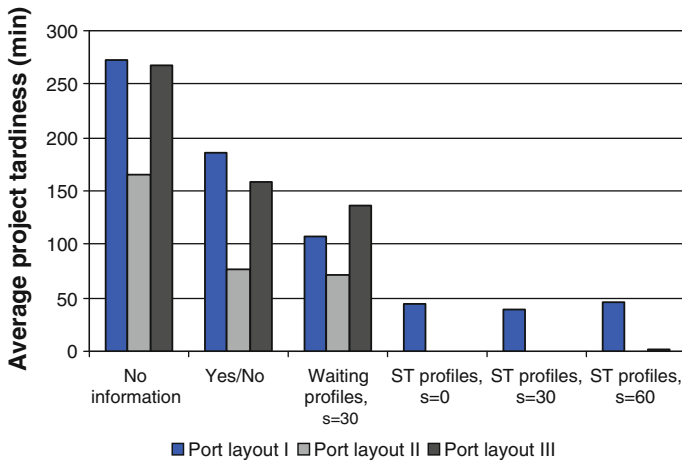


Fig. 5 Average project tardiness. Results for a utilization degree of 50%, fixed time windows

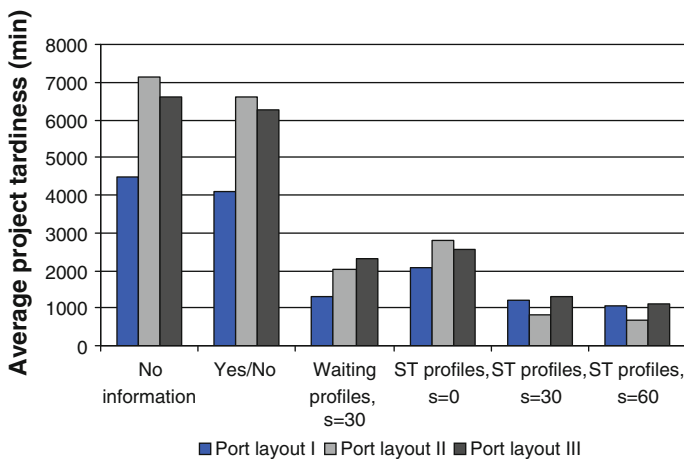


Fig. 6 Average project tardiness. Results for a utilization degree of 90%, fixed time windows

of all levels of information exchange denoted in Sect. 2 as well as service-time profiles. For the latter we vary the amount of slack $s \in \{0, 30, 60\}$ (in minutes) added to the service-time profile¹. In case of waiting profiles we add 30 min slack, since this led to the best results in Douma et al. (2009). Every scenario has a run length of 100 days, and we apply a warm-up and cool-down period of 10 and 3 days respectively.

¹ In Experiment 3 we also consider alternative slack policies by varying the amount of slack per terminal in one scenario

6.3.2 Results

In Figs. 5 and 6 we show the impact of restricted opening times on the average project tardiness, specified for scenario attributes, namely different port layouts (I, II, III), different utilization degrees (50, 90%), and different levels of information exchange (including different slack policies), assuming that barges have fixed time windows. Figures 7 and 8 show similar results as Figs. 5 and 6 but now for variable time windows.

When analyzing the figures we make the following observations.

1. For all scenarios it holds that service-time profile communication (for specific slack policies) performs better than waiting profiles (applying the same slack

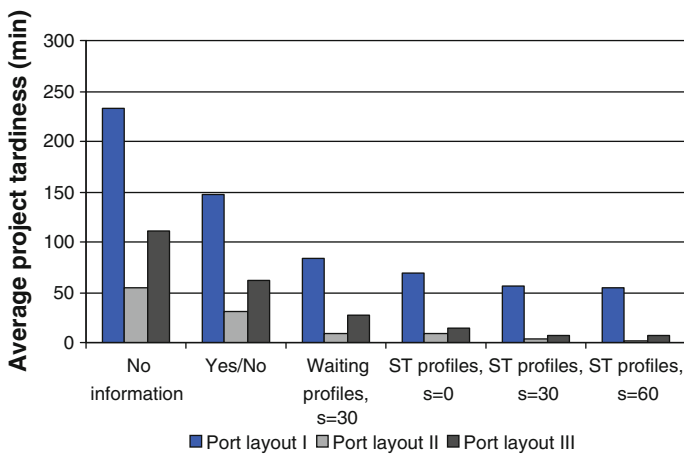


Fig. 7 Average project tardiness. Results for a utilization degree of 50%, variable time windows

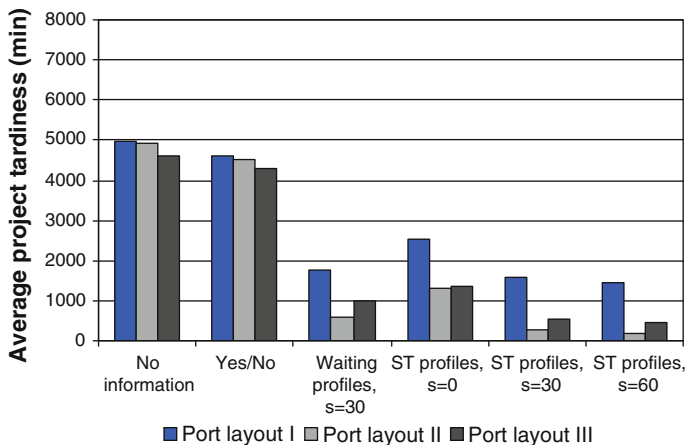


Fig. 8 Average project tardiness. Results for a utilization degree of 90%, variable time windows

- policies), which in turn performs better than yes-no information exchange, which in turn performs better than no information exchange.
- Waiting profiles with a slack policy $s = 30$ (ignoring restricted opening times of terminals) can perform better than service-time profiles with a slack policy $s = 0$. Adding slack to the service-time profiles is apparently more important for the average performance of barges than the awareness of restricted opening times of terminals.
 - The effect of service-time profiles (compared to waiting profiles) is larger for ports consisting of three regions than for ports consisting of one region. The reason is that with waiting profiles a barge has an optimistic view on the service time at terminals with restricted opening times. The barge might therefore accept more sailing time (by sailing between regions) as it expects that this will result in less waiting time. Sailing back and forth between regions is relatively time consuming compared to sailing in a region.
 - No information exchange and yes/no information exchange perform significantly worse than service-time profiles. The reason is twofold. First, barges plan a rotation without full knowledge of the availability of terminals. The resulting rotations therefore contain relatively much waiting time. Second, appointments are made without any slack. This means that an appointment with a specific barge cannot be moved, which reduces the planning flexibility of terminals.

Looking at the way barges deal with restricted opening times in case of different levels of information exchange, we find that in case of service-time profiles, barges avoid to visit terminals with restricted opening times at the end of the day, i.e., right before the terminal closes. This holds especially for lower utilization degrees (see Fig. 9). The reason is that barges limit the risk to stay overnight at a terminal which would lead to a significantly longer sojourn time in the port. In contrast, barges arrive during the night at these terminals so that the handling is started as soon as the terminal opens again.

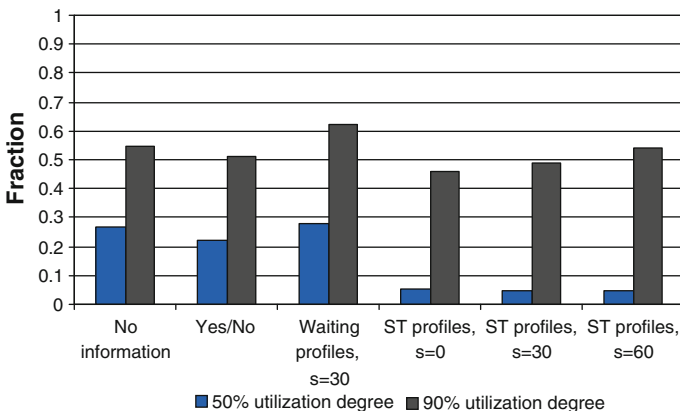


Fig. 9 The fraction of nights that terminals with restricted opening times have interrupted a barge handling due to closing of the terminal. Results averaged over all network layouts

6.4 Experiment 2: Unbalanced networks

To evaluate the effect of unbalanced networks we generate six scenarios, varying along the following dimensions: three different port layouts (see Fig. 4), and variable and fixed time windows. We aim to get insight in the impact of unbalanced networks on the (relative) performance of the different levels of information exchange. Note that the results of ‘waiting profiles’ are similar to the results of ‘service-time profiles’, since we do not consider terminals with restricted opening times in this experiment.

6.4.1 Experimental settings

In the experiments terminals have different characteristics, i.e., a different number of quays, different utilization degrees, and processing times. We distinguish four terminal types (see Table 5). We simulate different port layouts with specific configurations of terminal types, as given by Table 6. The reason we have chosen this distribution of terminals over the regions is that it is comparable with the Port of Rotterdam. We have two relatively quiet regions and one very busy region (region C) in the network layouts with three regions.

Terminals are opened 24 h a day in all scenarios. The slack factor used for the fixed time window is 0.75. The fixed slack y (slack per rotation) for the variable time window is equal to 0.5 and the variable slack x (slack per terminal) equal to 0.03. All experiments are replicated 10 times and each replication has a length of 75 days.

To gain additional insight we also simulate the scenarios using service-time profiles with varying amounts of slack per terminal. We consider six different policies (see Table 7). The choice for the slack values is based on previous experiments (see Douma et al. 2009).

Table 5 Different types of terminals, with corresponding number of quays, utilization degree, and average and standard deviation of the number of containers handled for every barge

Terminal type	#Quays	Utiliz. degree (%)	Mean call size	Stdev call size
Alpha	1	50	15	5
Beta	2	50	15	5
Gamma	3	75	40	20
Delta	6	90	60	30

Table 6 The number of different terminal types per port layout

Port layout-region	I		II and III		
	A	B	A	B	C
Alpha	6	6	6	8	3
Beta	2	1	1	1	3
Gamma	1	2	2	0	2
Delta	0	0	0	0	1

Table 7 The amount of slack for different terminal types

Terminal type	Slack policy					
	1	2	3	4	5	6
Alpha	0	30	60	0	0	0
Beta	0	30	60	0	0	0
Gamma	0	30	60	30	30	30
Delta	0	30	60	30	60	90

6.4.2 Results

In Figs. 10 and 11 we depict the average tardiness for respectively fixed and variable time windows for different scenarios.

Analyzing the figures we make the following observations.

1. In both Figs. 10 and 11 we see the same pattern in relative performance between the ‘levels of information exchange’ irrespective of the port layout
2. Comparing the six slack policies we observe that, in case of fixed windows, slack policies 4–6 (with varying slack per terminal) outperform slack policies 1–3. This does not hold for variable time windows, where it seems that the more slack the lower the average project tardiness. Interestingly, if we look at the average waiting time at the different terminal types for the six slack policies in port layout II (see Table 8) then we find that slack policy 5 leads to the shortest waiting time per terminal. The average waiting time is calculated as the difference between the arrival time and the start time of handling. Note that the average waiting time at a terminal is irrespective of fixed and variable time windows, since barges do not use these in their decisions. Table 8 confirms that slack policies 4–6 perform similar and outperform slack policies 1–3 in terms of average waiting time at the terminal.
3. An explanation for the fact that, in case of variable time windows, more slack leads to better results is rather technical and has to do with the type of barges that in particular run a risk to have non-zero project tardiness. In Fig. 14 we depict the average sojourn time of a barge in the port as a function of its rotation length. It is clear that the sojourn time increases nonlinearly with the rotation length. In case of variable time windows, one can imagine that especially barges with small rotation lengths run a risk to have non-zero project tardiness. In case of fixed time windows this holds for barges with large rotation lengths. For barges with small rotation lengths more slack is beneficial, since it creates more possibilities in the short term to be handled. We prefer to look at the average waiting time at the terminals, since in practice barge operators will use a mix of fixed and variable time windows.
4. Slack policies 4–6 perform similarly and have a significant lower average project lateness that slack policies 1–3 in case of both fixed and variable time windows (not depicted here).

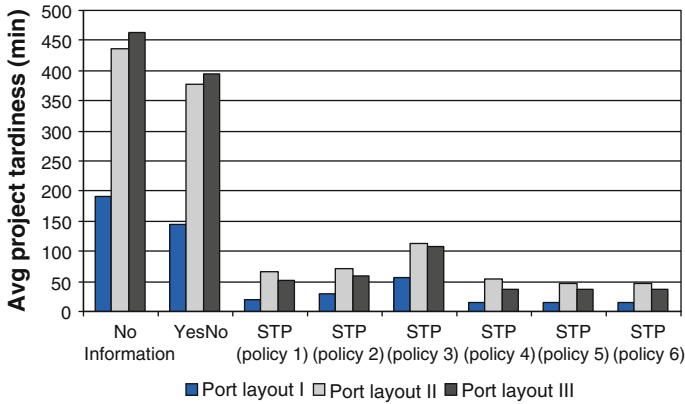


Fig. 10 Average project tardiness, fixed time windows

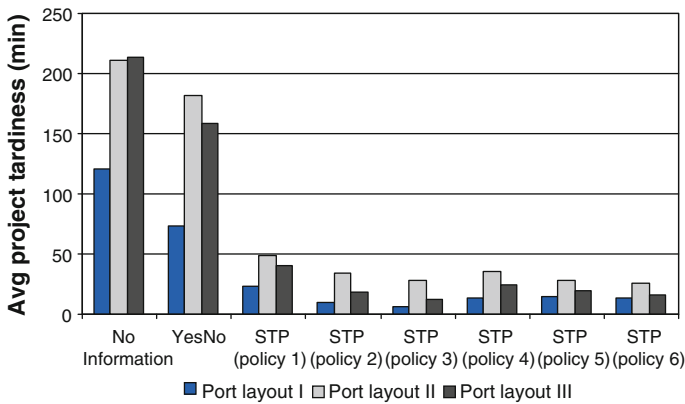


Fig. 11 Average project tardiness, variable time windows

Table 8 Average waiting time per terminal type for the alternative slack policies mentioned in Table 7

Terminal type	Slack policy						No information
	1	2	3	4	5	6	
Alpha	19	19	24	15	14	15	63
Beta	11	15	21	10	10	11	28
Gamma	38	41	60	32	30	31	164
Delta	141	155	183	129	117	118	519

In general it would be interesting to know which amount of slack results in the best performance of the system. Based on the simulated data sets it is hard to draw a conclusion about that. However, we expect that there is a relation between average

waiting time at the terminal and the best amount of slack to add to the service-time profile.

6.5 Experiment 3: Restricted opening times, unbalanced networks, sea vessels, and closing times of containers

In this experiment we consider all practical extensions (introduced in Sect. 1) in combination. However, a full-factorial analysis (varying all parameters in all possible combinations) is out of the scope of our study, since the simulation effort is computationally intractable. Moreover, in the previous experiments we have seen that the relative performance of different levels of information exchange has a similar pattern for different port layouts. We therefore choose to consider the settings in Sect. 6.5.1 which are inspired by the Port of Rotterdam. Section 6.5.2 presents the results of the experiments.

6.5.1 Experimental settings

We extend the settings of Experiment 2, with respect to the introduction of restricted opening times, sea vessels, and closing times of containers. We use one port layout (layout II) with every region containing nine terminals. The port configurations are similar to Table 6. However, the terminal types have some additional properties. The alpha terminals are closed from 6 pm–6 am and do not handle sea vessels. The other terminal types are opened 24 h a day and do handle sea vessels. The fraction of time they work on sea vessels per day is 40%. Sea vessels are handled at one quay, like barges.

The processing time of sea vessels is based on historical data of 23,000 sea vessels that visited the Port of Rotterdam at the major terminals in the period 2005–2007. Based on this data we decided to apply a beta distribution with parameters β_1 and β_2 , equal to 1.14 and 8.3, respectively. The corresponding mean and standard deviation is equal to 770 and 645 min. Based on these settings, we calculate the number of sea vessels that have to arrive during a certain time period in order to realize that terminals work 40% of their time on sea vessels. Based on the same historical data, we assume that sea vessels arrive with exponentially distributed interarrival times in the port. This is to be expected as liner shipping alliances determine their sailing schedules independently. Moreover, the assumption is in accordance with results found by Kuo et al. (2006).

Sea vessels are generated per terminal. The mean interarrival time depends on the number of sea vessels the terminal has to process to realize the utilization degree we aim for. If a sea vessel is about to arrive during processing of another vessel and a terminal has no capacity available to process the vessel, we delay the arrival of the sea vessel until the handling of the first sea vessel is completed. The arrival time of the next sea vessel, however, is calculated using the initial arrival time of the previous vessel.

Closing times of containers are assigned uniformly to 20% of the activities of a barge that are handled at terminals where also sea vessels are handled. A closing

time is drawn uniformly distributed from the time window of the barge (both for fixed and variable time windows).

We consider all levels of information exchange considered in Experiment 1 including service-time profiles. In case of waiting profiles we apply slack policy 5 and in case of service-time profiles we apply slack policies 1–3 and 5 (see Table 7). We vary the penalty for the project tardiness $\gamma \in \{0, 10\}$. We simulate 30 replications for every scenario, and every replication has a length of 100 days. We apply a warm-up period of 10 days and a cool-down period of 3 days. We also compare the performance of the different levels of information exchange with the off-line benchmark. This comparison is based on 10 replications of 15 days and we apply a warm-up and cool-down period of 4 respective 3 days. The reason for this choice is that the off-line benchmark can compute a solution for a time horizon of 15 days within a reasonable amount of time and the remaining number of days (after subtraction of the warm-up and cool-down period) is still acceptable for our analysis.

6.5.2 Results

When analyzing the simulation results we find the same pattern in the relative performance of the different levels of information exchange as in Experiments 1 and 2. We therefore focus on the comparison with the off-line benchmark. In Fig. 12 we compare the performance of the alternative controls in terms of the *average activity tardiness* (for a project tardiness penalty of zero) and in Fig. 13 in terms of the *average project tardiness* (for a project tardiness penalty of ten). For service-time profiles we only depicted the results for slack option 5, since it outperforms slack policy 1 in Table 5 with slack policy 5, we find that adding slack to higher utilized terminals reduces the average waiting time at these terminals with about 2–6% and improves the performance of barges in general. An explanation for the relatively small reduction in average waiting time is that only a few terminals act as bottleneck terminal due to a high utilization degree and many sea vessel visits. These terminals

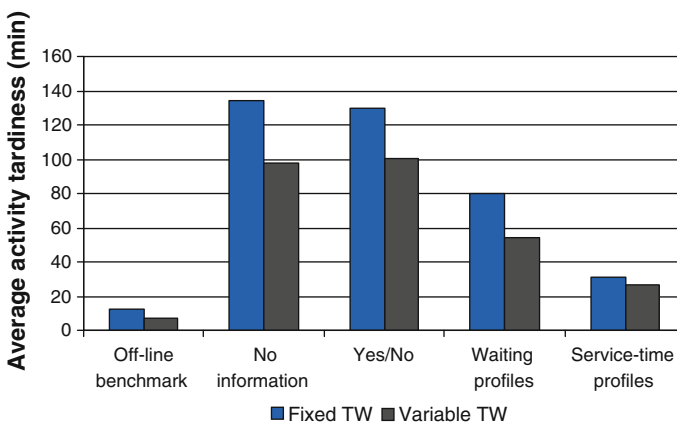


Fig. 12 Average activity tardiness. Simulation results in comparison with the off-line benchmark, $\gamma = 0$

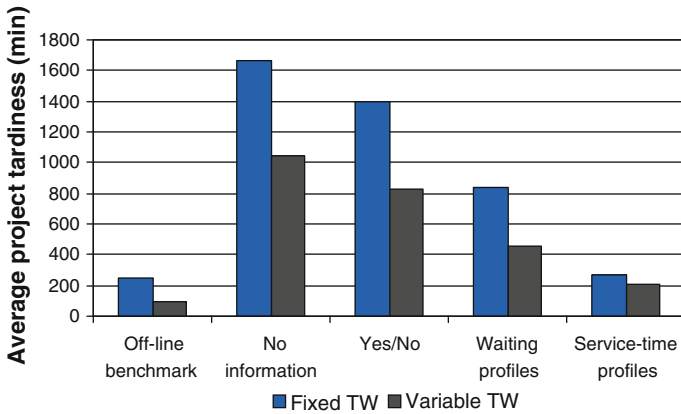
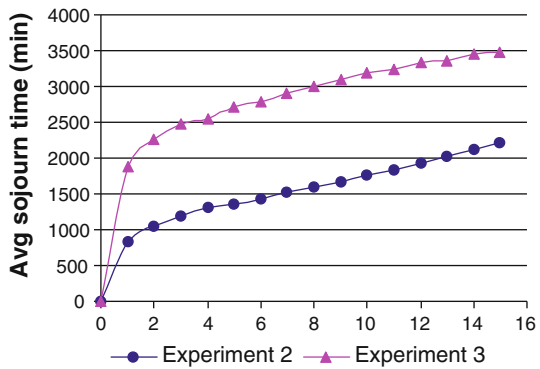


Fig. 13 Average project tardiness. Simulation results in comparison with the off-line benchmark, $\gamma = 10$

Fig. 14 Average sojourn of barges in Experiment 2 and 3 for different rotation lengths (port layout II)

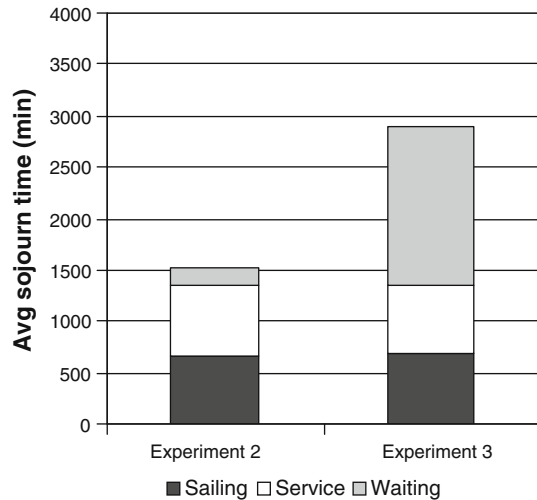


can plan barges one after another and during the daily operations they usually have a queue of barges to choose from (since most barges arrive earlier than their latest arrival time) allowing them to operate efficiently. Nevertheless, the results indicate that in practical settings a slack policy that differentiates based on the utilization degree of terminals is preferable. For instance, terminals with a lower utilization degree (50%) require less flexibility since they are already flexible in making appointments.

Generally, there is not much difference in the performance of the service-time profiles when we raise the project tardiness penalty from 0 to 10, except that the average activity tardiness slightly increases with about 5%. It seems that our Multi-Agent system has difficulty to minimize the activity lateness. This can be explained by the fact that barges are planned dynamically and that there are only minor possibilities to swap activities with other barges (which is similar to practice). Moreover, terminals in our model make appointments on a first-come first-served basis, i.e., they do not reserve capacity for containers with a closing time.

From Fig. 13 it is clear that the off-line benchmark outperforms the decentralized control options. However, the performance of service-time profiles gets very close

Fig. 15 Average service, sailing, and waiting time for a rotation with 7 terminal visits in Experiment 2 and 3 (port layout II)



to the off-line benchmark. Note, that the off-line benchmark assumes all information over the planning horizon, to be known a priori, whereas in the simulations information becomes available over time (dynamically).

In Fig. 14 we take a closer look at the sojourn time of barges in Experiments 2 and 3. Figure 14 depicts the average sojourn time of barges for different rotation lengths using service-time profiles in port layout II. It is clear that the sojourn time of barges increases significantly as a result of sea vessels, restricted opening times, and closing times of containers. The reason for this increase is mainly an increase in waiting time at terminals, as indicated by Fig. 15.

6.6 Experiment 4: Realistic data

The previous experiments (Experiments 1 to 3) are based on fictitious data to investigate the performance of different levels of information exchange for various scenarios with increasing practical complexity. In this experiment (Experiment 4) we provide insight in the performance of the different levels of information exchange for realistic data. Unfortunately, the system has not been implemented yet in practice, so we can not present results on the performance of our Multi-Agent system in practice. However, we can provide simulation results based on a realistic data set of the Port of Rotterdam in the period 2006–2007.

Creating a realistic data set of the Port of Rotterdam is not trivial, since we had to deal with the following limitations. First, it turned out that registrations of activities in the port are limited, ambiguous, incomplete, distributed, and sometimes even contradicting. Second, structural changes in the port take place frequently, which means that a model of the port today may not be valid anymore tomorrow. Third, our model can deal with only part of the practical aspects that may be relevant for the problem. Despite these limitations, we think that we managed to create a (fairly) realistic model of the Port of Rotterdam and provide valuable insights in the performance of our Multi-Agent system. The reason why we think so, is that we can

still validate our model on certain levels of abstraction by checking the internal consistency and by making a comparison with figures published by, e.g., the Port of Rotterdam. We note, however, that considering the quality of the information we have at our disposal, we do not claim to quantify the savings that can be realized using a Multi-Agent system.

We apply the same model as used for the previous experiments for which we made the assumptions as denoted in Sect. 2. Assuming that no-information exchange and yes-no information exchange are reasonable approximations of how barge rotations are currently planned, we can provide insight on the approximate room for improvement in practice as well as in comparison with central coordination (the off-line benchmark).

6.6.1 *Experimental settings*

We obtained the following practical data for the period 2006–2007. First, information about the terminals that handle containers in the Port of Rotterdam, such as the estimated utilization degree, number of quays, opening times, and the average and standard deviation of the barge call size. Second, data about the mooring time of barges, the average time to move a container from a barge to the quay (and vice versa), the average number of terminals in a rotation (specified for different hinterland regions), and the sailing time between terminals. Third, data on the annual number of containers handled by the larger container terminals and the Port of Rotterdam as a whole. Information on the container sea vessels that visited the major container terminals in the Port of Rotterdam in the period 2005–2007. This data has been obtained using different sources, such as data provided by a consulting company, the Port of Rotterdam, interviews with experts, master theses, and web resources. The data obtained from these sources has undergone several adjustments to get a setting, which is internally consistent and consistent with the different sources of information and, to our opinion, a reasonable approximation of the activities performed in the Port of Rotterdam in the period 2006–2007.

The practical data set we used contains 33 container terminals spread over three regions in a linear port layout. Terminals handle 2 million containers per year, which is about 700,000 moves more than reported by the Port of Rotterdam. These moves concern the handling of, e.g., empty containers, which are not included in the statistics of the Port of Rotterdam. On average 48 barges visit the port every day, having a utilization degree of 80% (assuming a capacity of 120 TEU). Annually about 123,500 appointments are made by all barges together. Barges visit on average seven terminals and have call-sizes depending on the terminals they visit. The terminals that a barge visits are randomly assigned based on the average workload of these terminals.

We consider, apart from service-time profiles, three levels of information exchange, namely (1) no information, (2) yes/no, and (3) waiting profiles. We consider 30 replications of 100 days each. We apply a warm-up period of ten days and a cool-down period of three days. We again consider both fixed and variable time windows. For the fixed time window we use a slack factor of two. For the

variable time window we use a fixed slack (y) per rotation equal to one, and a variable slack per terminal (x) equal to 12%. These values are determined experimentally. We do not consider closing times of containers. The objective of the barges is to minimize the sojourn time in the port. For the simulation of the waiting profiles and service-time profiles we apply slack policy 5 of Table 8. In addition to the Multi-Agent control we also evaluate the performance of the off-line benchmark. For the comparison with the off-line benchmark we consider 20 replications of 10 days each, and a warm-up period of 4 days and a cool-down period of 3 days.

6.6.2 Results

When looking at the results we find the same relative pattern in performance between the different levels of information exchange as in the previous experiments. We therefore focus on the comparison with the off-line benchmark in Figs. 16 and 17. We find again that the Multi-Agent system with service-time profiles outperforms the other levels of information exchange and performs well compared to the off-line benchmark. Moreover, it is clear that waiting profiles can not deliver the same performance as service-time profiles.

The average sojourn time of barges with a rotation length of seven terminals is about 36 h (in case of service-time profiles). This consists of about 9 h handling, 9 h sailing, and 18 h waiting. The waiting time boils down to about 2.6 h per terminal on average. The sojourn time of a barge in case of ‘No Information’ is about 69 h, consisting of 7 h sailing, 9 h handling, and 54 h waiting. We see that the increased performance of the service-time profiles is mainly due to a decrease in the average waiting time at terminals.

We depict the arrival patterns of barges during the day for different terminals in Fig. 18. One can see that, e.g., terminal C19 (closed from 10 pm to 7.30 am) clearly has more barge arrivals during the day than in the night. Terminal C21, however, is

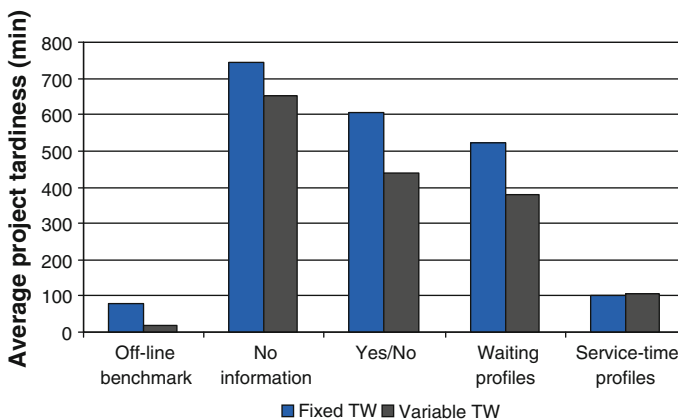


Fig. 16 Average project tardiness for different levels of information exchange in comparison with the off-line benchmark

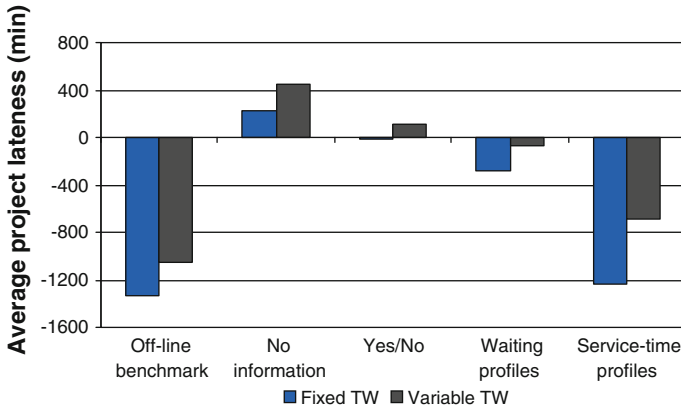
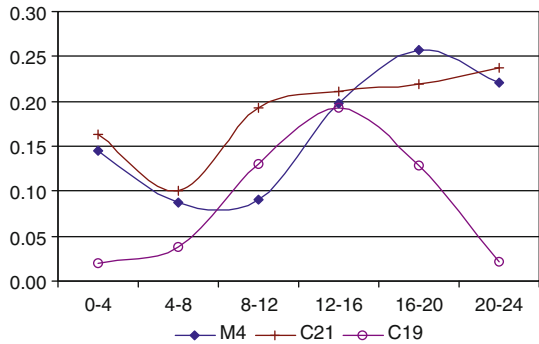


Fig. 17 Average project lateness for different levels of information exchange in comparison with the off-line benchmark

Fig. 18 Fraction of barges arriving at terminals M4, C21, and C19 during certain time intervals of the day



opened 24 h a day and we see that especially at the start of the day there is a decline in the fraction of barge arrivals. The reason why is that the terminal is surrounded by terminals that close during the night. When these terminals open, barges prefer to head to these terminals first. Terminal M4, on the other hand, is opened 24 h and *not* surrounded by terminals that close during the night. One can see that the arrival pattern has a wave-like shape during the day. The reason for this is that at the end of the day, several terminals in other regions close and that barges prefer to visit 24 h terminals during the night.

7 Conclusions and further research

We presented a Multi-Agent system for the barge handling problem that can be used in practical settings and enables barge and terminal operators to align their operations efficiently. The model we developed is a major step towards practice. We presented the model to practitioners (barge and terminal operators) and they became

enthusiastic about the potential of our Multi-Agent system for solving the barge handling problem.

In an earlier paper we presented a Multi-Agent system for the barge handling problem for simplified port settings, which serves as base model in this paper. In workshops with practitioners we got positive feedback on the base model, the used interaction protocol and the assumptions made regarding the behavior of agents. However, they also wondered whether the model would be applicable in practical settings and how it would then perform. The practical aspects that were considered to be the most relevant are: (1) restricted opening times of terminals, (2) sea vessels, (3) closing times of containers, and (iv) unbalanced networks. In this paper we showed how these aspects resulted in a need to redesign the interaction protocol and the agent intelligence. Additionally, by means of a simulation study, we showed how our Multi-Agent system performs in practical settings compared to exchanging less information and compared to central coordination.

To do so, we adapted the base model, to deal with the four practical aspects. For this we introduced the concept of service-time profiles (as an extension of the waiting profile concept) and explained how they can be constructed dynamically based on a terminal quay schedule. Additionally, we changed the barge operator intelligence to deal with closing times of containers. Second, we changed the performance measurement to also deal with the new constraint imposed by closing times of containers. Third, we evaluated our new model and compared the performance with central coordination.

We performed four experiments. The first three experiments are based on fictitious data inspired by the Port of Rotterdam. We used these experiments to study the impact of different practical aspects on the performance of our distributed planning approach. The fourth experiment is based on realistic data of the Port of Rotterdam in the period 2006–2007.

Based on the results of the four experiments we conclude that our distributed planning approach performs well in comparison with the off-line benchmark (resembling central coordination). Moreover, service-time profiles clearly outperform other levels of information exchange, including waiting profiles.

One striking result through all experiments is the value of slack. We conclude that adding slack to appointments leads to significant improvement of the performance of our distributed planning approach. The reason is that slack enables the terminal to vary the start time of the barge handling process. In this way the planning flexibility of the terminal improves, which can be observed by lower sojourn times of barges. Our experimental results indicate that the amount of slack added to the terminal depends, among others, on the utilization degree of the terminal. Although we evaluated different slack policies, additional research is required to study the relation between slack in an appointment and terminal performance in terms of utilization degree and average barge waiting time.

We also observed that restricted opening times and sea vessels have a major impact on the sojourn times of barges. Moreover, the opening times of terminals impact the arrival pattern of barges at terminals and thus the sequence in which a barge visits terminals. Generally, we observe that barges avoid visiting terminals with restricted opening times at the end of the day to avoid the risk of staying

overnight at the terminal, in particular when terminals have a low utilization degree. In unbalanced networks (ports with terminals that differ in, e.g., size or utilization degree) we found that barges plan their rotations based on the bottleneck terminals, which are usually highly occupied and less flexible than the other terminals. Bottleneck terminals determine to a large extent the sojourn times of barges in the port, which implies that a reduction of the rotation length not necessarily leads to a proportional decrease of the sojourn time of barges in the port.

Although our distributed planning approach can deal well with practical aspects, such as restricted opening times of terminals, unbalanced networks, and sea vessels, we also found that our approach has difficulty to minimize the activity tardiness in case containers have a closing time. The reason is that barges plan rotations successively and have minor opportunities to swap activities with other barges.

In our experiments we also made a start with exploring how our distributed planning approach performs in the realistic setting of the Port of Rotterdam. Our results indicate that our distributed planning approach can really make a difference in practice and might lead to significantly lower sojourn times of barges. Considering our first promising results, it would be worthwhile to study this in more detail.

We have presented our Multi-Agent system also to practitioners by means of a (simplified) simulation game. Both barge and terminal operators consider the system as a promising solution to solve the barge handling problem for the following reasons. First, they can stay autonomous and share only limited information. Second, they expect to improve the efficiency of their decision making process and the quality of their decisions. Third, the system improves the reliability of the barge handling process which has a lot of value for both terminal and barge operators. Fourth, they see opportunities to develop the system so that they can plan more dynamically which allows them to respond quickly to opportunities and events.

With the above practical issues addressed, the interaction protocol based on service-time profiles is found to be promising and is expected, once implemented, to yield significant benefits for barge and terminal operators. An interesting topic for further research concerns the question how to deal with (major) uncertainties and disturbances. Small disruptions can probably be dealt with, without reconsidering appointments that are made, but large disruptions (such as a crane break-down) probably require rescheduling of appointments. Terminal and barge operators have to make agreements on how often and when appointments may be cancelled or changed. Another topic is the tuning of the terminal and barge operator intelligence to the working practice of these companies. For instance, terminal operators may want to use a tactical plan that can be used by the agent to make appointments at operational level. Barge operators, on the other hand, may want to choose between different rotations depending on their preferences, such as the fastest or the most sustainable rotation. Finally we mention a topic that concerns one of the basic assumptions, namely that appointments are kept by all the parties involved. It would be interesting to examine how this assumption can be realized in practice. What mechanism can induce parties to keep appointments and avoid strategic behavior that undermines the functioning of the system as a whole? Though, in our model, we tried to realize self-regulation where possible, a supervising port authority may be

required to enforce appointment keeping behavior. One of the possibilities might be a rating list—made public by the port authority—that displays the extent to which parties keep their appointments. But surely, this has pros and cons. Further research, e.g. in the field of governance, will lead the way. It is certainly worthwhile, since this paper shows that—in a practical situation—the benefits of our appointment based Multi-Agent system, are considerable. This is also recognized by market players and the Port of Rotterdam. They have shown great interest in the solution and have requested the building of a prototype to make a next step towards implementation.

Acknowledgment 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 criticism and their useful suggestions.

Open Access This article is distributed under the terms of the Creative Commons Attribution Non-commercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

- Brucker P, Drexel A, Mohring R, Neumann K, Pesch E (1999) Resource-constrained project scheduling: Notation, classification, models, and methods. *Eur J Oper Res* 112:3–41
- Caris A, Macharis C, Janssens GK (2010) Network analysis of container barge transportation in the Port of Antwerp by means of simulation. *Journal of Transport Geography* (in press)
- Christiansen M, Fagerholt K (2002) Robust ship scheduling with multiple time windows. *Naval Res Logist* 49(6):611–625
- Connekt (2003) Eindrapport approach. Tech. report, Connekt, Delft (in Dutch)
- Davidsson P, Henesey L, Ramstedt L, Törnquist J, Wernstedt F (2005) An analysis of agent-based approaches to transport logistics. *Transp Res Part C* 13(4):255–271
- Demeulemeester EL, Herroelen WS (1992) A branch-and-bound procedure for the multiple resource-constrained project scheduling problem. *Manage Sci* 38(12):1803–1818
- Demeulemeester EL, Herroelen WS (2002) *Project scheduling: a research handbook*. Kluwer, Dordrecht
- Douma AM, Schutten JMJ, Schuur PC (2009) Waiting profiles: an efficient protocol for enabling distributed planning of container barge rotations along terminals in the Port of Rotterdam. *Transp Res Part C* 17:133–148
- Douma AM, van Hillegeersberg J, Schuur PC (2008) Using a management game to exemplify a multi-agent approach for the barge rotation and quay scheduling problem in the Port of Rotterdam. In: Blecker T, Kersten W, Gertz C (eds) *Management in logistics and nodes*. Erich Schmidt Verlag GmbH & Co, Boston
- Durfee EH, Lesser VR (1989) Negotiating task decomposition and allocation using partial global planning. In: Gasser L, Huhns MN (eds) *Distributed artificial intelligence: vol 2*. Morgan Kaufmann Publishers Inc, San Francisco, CA, pp 229–243
- Jennings NR, Sycara K, Wooldridge MJ (1998) A roadmap of agent research and development. *Auton Agents Multi Agent Syst* 1(1):7–38
- Kolisch R, Drexel A (1996) Adaptive search for solving hard project scheduling problems. *Naval Res Logist* 43(1):23–40
- Konings R (2007) Opportunities to improve container barge handling in the Port of Rotterdam from a transport network perspective. *J Transp Geogr* 15:443–454
- Kuo TC, Huang WC, Wu SC, Cheng PL (2006) A case study of inter-arrival time distributions of containers ships. *J Mar Sci Technol* 14(3):155–164

- Malandraki C, Daskin MS (1992) Time dependent vehicle routing problems: formulations, properties and heuristic algorithms. *Transp Sci* 26(3):185–200
- Malandraki C, Dial RB (1996) A restricted dynamic programming heuristic algorithm for the time dependent traveling salesman problem. *Eur J Oper Res* 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, Loo B (eds) *Intelligent logistics concepts*. Eburon Publishers, The Netherlands, pp 101–116
- Mes MRK, Van der Heijden MC, Van Hillegersberg J (2008) Design choices for agent-based control of AGVs in the dough making process. *Decis Support Syst* 44(4):983–999
- Moonen HM, Vande Rakt B, Miller I, Van Nunen J, Van Hillegersberg J (2007) Agent technology supports inter-organizational planning in the port. In: Koster RBM, Delfmann W (eds) *Managing supply chains—challenges and opportunities*. Copenhagen Business School Press, Copenhagen
- Notteboom TE (2004) Container shipping and ports: an overview. *Rev Netw Econ* 3(2):86–106
- RIL Foundation (1998) *Afhandeling Containerbinnenvaart Rotterdam*. Tech. report, B-RIL, CBRB, and TNO-Inro
- RIL Foundation (2000) *Functioneel ontwerp Bargeplanning.nl*. Technical report
- Sandholm T (1999) *Multiagent systems, a modern approach to distributed artificial intelligence*, 1st edn, ch. Distributed rational decision making. MIT press, Cambridge, pp 201–258
- Schut M, Kentrop M, Leenaarts M, Melis M, Miller I (2004) Approach: decentralised rotation planning for container barges. In: *Proceedings of the 16th European conference on artificial intelligence (Valencia)*, prestigious applications intelligent systems
- Schutten MJM (1996) List scheduling revisited. *Oper Res Lett* 18:167–170
- Schutten MJM (1998) Practical job shop scheduling. *Ann Oper Res* 83:161–178
- Stahlbock R, Voß S (2008) Operations research at container terminals: a literature update. *OR Spect* 30:1–52
- Steenken D, Voß S, Stahlbock R (2004) Container terminal operation and operations research—a classification and literature review. *OR Spect* 26:3–49
- Wooldridge MJ (2005) *An introduction to multi agent systems*, 4th edn. Wiley, New York
- Wooldridge MJ, Jennings NR (1995) *Intelligent agents: theory and practice*. *Knowl Eng Rev* 10(2):115–152
- Zlotkin G, Rosenschein JS (1996) Mechanism design for automated negotiation, and its application to task oriented domains. *Artif Intell* 86:195–244

Author Biographies

Albert M. Douma Until February 2011, Albert Douma was employed as a researcher at the School of Management and Governance at the University of Twente, The Netherlands. From this university he received a PhD in Industrial Engineering, and an MSc in Industrial Engineering & Management and in Business Information Technology. His research interests are in the fields of supply chain management, business information systems, and economics and include Multi-Agent systems, logistics, collaboration in supply chains, production planning, pricing in freight transport, multi-modal networks, simulation, serious gaming, main ports, change management, and simulation. He is currently employed by the Dutch Institute of Advanced Logistics (Dinalog).

Peter C. Schuur Peter Schuur received his Ph.D. in Mathematical Physics from the University of Utrecht. On the theoretical side his research interests include: packing and covering problems, as well as reinforcement learning and game theory. On the applied side his research interests include: (closed loop) supply chain management, more specifically: order acceptance, vehicle routing, distributed planning, reverse logistics, layout problems, and warehouse modeling.

J. Marco J. Schutten Marco Schutten is an Assistant Professor at the School of Management and Governance of the University of Twente, The Netherlands. He received his Ph.D. in 1996 from this university on the subject of job shop scheduling. His current research interests include transportation management, production and project planning and scheduling, and material handling systems.