Beta

Research School for Operations Management and Logistics

Aligning the Operations of Barges and Terminals through Distributed Planning

Albert Douma

Aligning the Operations of Barges and Terminals through Distributed Planning

Albert Douma

Research School for Operations
Management and Logistics

This thesis is number D-114 of the thesis series of the Beta Research School for Operations Management and Logistics. The Beta Research School is a joint effort of the departments of Technology Management, and Mathematics and Computing Science at the Technische Universiteit Eindhoven and the Centre for Telematics and Information Technology at the University of Twente. Beta is the largest research centre in the Netherlands in the field of operations management in technology-intensive environments. The mission of Beta is to carry out fundamental and applied research on the analysis, design, and control of operational processes.

Dissertation committee

Chairman	Prof. dr. ir. E.J. de Bruijn (University of Twente)
Secretary	Prof. dr. P.J.J.M. van Loon (University of Twente)
Promotor	Prof. dr. ir. J.H.A. de Smit (University of Twente)
Assistant Promotor	Dr. P.C. Schuur (University of Twente)
Members	Prof. dr. E. Hagdorn-Van der Meijden (VU University)
	Prof. dr. J. van Hillegersberg (University of Twente)
	Prof. dr. A.G. de Kok (Eindhoven University of Technology)
	Prof. dr. S. Voß (University of Hamburg)
	Prof. dr. ir. M.J.F. Wouters (University of Twente)

This research has been partly funded by Transumo. Transumo (TRANsition to SUstainable MObility) is a Dutch platform for companies, governments and knowledge institutes that cooperate in the development of knowledge with regard to sustainable mobility.

Transumo

Ph.D. thesis, University of Twente, Enschede, the Netherlands
Printed by Wöhrmann Print Service, Zutphen

ISBN 978-90-365-2735-4

ALIGNING THE OPERATIONS OF BARGES AND TERMINALS THROUGH DISTRIBUTED PLANNING

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof. dr. W.H.M. Zijm,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen
op dinsdag 9 december 2008 om 15:00 uur

door

Albert Menno Douma
geboren op 9 april 1980
te Achtkarspelen

Dit proefschrift is goedgekeurd door de promotor:

prof. dr. ir. J.H.A. de Smit

en de assistent promotor:

dr. P.C. Schuur

To God be the glory

Acknowledgements

Writing this thesis was a rich experience and a special opportunity to develop both professionally and personally. For that, I owe a great deal of gratitude to many people of whom I like to mention a few in particular.

First prof. Aart van Harten and Matthieu van der Heijden. Both put a lot of energy in setting up this research project. Sadly, Aart got ill soon after I was employed and passed away in December 2006. We still miss him. Due to the illness of Aart several task in our department were rearranged. The result was that prof. Jos de Smit and Peter Schuur took over my supervision.

I am grateful to prof. Jos de Smit for being my promotor, even after his retirement. His great experience and keen sense of language greatly improved my thesis. I thank Peter Schuur for his committed and personal supervision. He taught me much and gave me many opportunities to take advantage of his broad knowledge and experience in many fields. Peter, thanks a lot!

I thank Erwin Hans and Waling Bandsma for encouraging me, as Master's thesis supervisors, to consider a Ph.D. position. I am happy that they did so.

I thank my colleagues for the great working atmosphere and the nice activities after working hours. To my mind, only a few days passed without having laughed. I thank Marco Schutten in particular for his help with developing the off-line benchmark and being a sparring-partner. Special thanks also to Martijn Mes. Besides being a travel companion on two international conferences, I valued much that he was always available for discussing and commenting ideas.

Thanks also to Rama Jagerman and Saskia Kuipers who helped me developing an alternative agent model (used in Chapter 7) and a realistic model of the Port of Rotterdam.

Within my research I cooperated with several parties. In particular I thank Initi8 B.V. for sharing their broad practical knowledge about the Port of Rotterdam. In addition, I thank prof. Jos van Hillegersberg for the pleasant cooperation and the possibilities to develop several initiatives, like the management game. I also acknowlegde the financial support from Transumo.

Finally, special thanks to my parents for their patience and care in my life, especially during the last four years. Pa en ma, dank jullie wel daarvoor!

Enschede, December 2008
Albert Douma

Contents

Chapter 1

Introduction

1.1 Introduction and motivation

For many years, companies used to focus their attention on optimization of the business processes within their organization. Over the last decades, companies have started to realize the strategic importance of the linkages among supply chain activities. As a result, some companies started to integrate and coordinate the intricate network of business relations among supply chain members. However, the alignment of operations of different companies often requires the sharing of information or to give up part of the control over the operational processes. For many companies these are difficult issues, since misuse can threaten one's competitive position in the chain.

Nevertheless, companies that are part of a supply chain have to align their operations in one way or another. However, it seems that supply chain members have difficulty to find coordination mechanisms through which they can align the activities in the supply chain, especially in the absence of one powerful player. The coordination mechanisms should allow for coordination in such a way that synergy can be achieved, while at the same time the interest of each of the companies is guaranteed satisfactorily. Traditionally, techniques to optimize operational processes (the realm of Operations Research and Operations Management) mainly focused on optimization for a single objective function measuring the overall system performance (centralized optimization). This requires that the objectives of all supply chain members can be rephrased to one objective function, which in turn requires that companies agree on the weights given to their respective interests. The latter is another difficult issue.

Centralized coordination mechanisms, through which the activities of all companies in a supply chain are coordinated by one trusted party, are therefore not always accepted by the companies involved. Distributed control mechanisms, on the other hand, may be a promising alternative. Research into distributed control mechanisms has increased since the introduction of Multi-Agent systems. Multi-Agent systems provide a platform for distributed control or distributed planning. Applications of Multi-Agent systems can be found in different fields, such as economics, computer science, and logistics. A limited number of studies have been devoted to the design of Multi-Agent systems for aligning the operations of companies in supply chains and to the performance of these Multi-Agent systems in general and in comparison with traditional optimization techniques.

In this thesis we consider a specific problem -occurring in, among others, the Port of Rotterdam (The Netherlands)-, which we call the *barge handling problem (BHP)*. The problem is about aligning the operations of container barges and terminals in a port. In the problem we deal with competitive actors that have to cooperate, but only want to do so under specific conditions.

In Section 1.1.1 we explore the barge handling problem. Section 1.1.2 illustrates the relevance of the problem. Section 1.1.3 describes several factors that complicate the design of a solution to the barge handling problem. In Section 1.1.4 we describe a few earlier studies to the problem. Finally, we describe in Section 1.1.5 the outline of the chapter.

1.1.1 The barge handling problem

The barge handling problem concerns the alignment of container barge and terminal operations in a port. Throughout our study, we use the Port of Rotterdam as our major case of reference, although our model is applicable to general multi-terminal, multi-barge settings. To fix ideas, let us describe the barge handling problem by focusing on the Port of Rotterdam.

In the Port of Rotterdam, barges are used to transport containers from the port to the hinterland, and vice versa. Every time a barge arrives in the Port of Rotterdam it visits several terminals to load and unload containers. The sequence in which a barge visits these terminals depends, among others, on the availability of terminals. The availability of terminals, in turn, is depending on other barges and sea vessels that have to be processed.

In the remainder of this thesis we will mainly talk about barge operators and terminal operators. Barge operators are companies that offer and organize barge container transportation services to and from the hinterland. These companies usually do not own barges themselves, but contract barge companies, which are companies that own and operate barges. Terminal operators are companies that operate a terminal. Terminals are used for the transshipment and temporary storage of containers.

Nowadays, barge and terminal operators try to align their activities by making appointments. These appointments are made by telephone, fax, and e-mail.

The barge operator usually initiates the communication with the terminal operators, to determine the most convenient rotation (sequence of terminal visits) for the barge concerned. The barge operator makes these appointments together with the *stowage plan*, i.e., the plan indicating the locations of all the containers on a ship, since the sequence in which terminals are visited determines the sequence in which containers are (un)loaded. This implies that appointments are sometimes made already one or two days in advance.

Unfortunately, it happens frequently that appointments are not (or cannot) be met by either the barge or the terminal operator. There are several reasons (see, Melis et al., 2003; Moonen et al., 2007). For example, appointments are sometimes not even feasible at the time they are made. In addition, the fact that barges usually visit several terminals, creates dependencies between the activities performed at the terminals. Thus, a disruption at one terminal can quickly propagate through the port and disturb the operations of other barge and terminal operators. The result is that barge operators face uncertain waiting and handling times at terminals, and that terminals deal with uncertain arrival times of barges.

The uncertainty in the alignment process leads to several undesirable effects. For example, some barges try to influence their processing times at terminals by exhibiting strategic behavior. They reserve and cancel time slots, announce wrong numbers of containers, etcetera, to obtain convenient time slots for handling. Some terminals, on the other hand, respond by creating queues of barges to prevent idle times at their quays. These conducts make the alignment process even more uncertain and do not contribute to a good relationship between the terminal and the barge operators.

For a barge operator it is important that barges have short and reliable sojourn times in the port. The uncertainty in the sojourn time of a barge in the port nowadays is mainly determined by uncertain waiting and handling times at terminals. Barge operators anticipate uncertainty by stowing their barges such that they are able to deal flexibly with the actual waiting times at terminals. For example, they stack containers with the same terminal destination on top of each other, as much as possible. However, this flexibility is at the cost of a high utilization degree. Moreover, the uncertain sojourn times require more slack in the sailing schedules of barges, which means that a barge makes fewer round trips to the hinterland.

For a terminal operator it is important to utilize the terminal resources as efficiently as possible. The uncertainty in arrival times of barges implies uncertainty in the quay schedules and the risk of idle time of the quay resources. Moreover, uncertainty in the quay schedules causes uncertainty in the processes that precede the barge handling, e.g., the stacking of containers at the quays. Since barges sometimes visit terminals in a different order than planned, they do not always have enough capacity to pickup all the containers that were initially announced. Terminal operators then have to decide what to do with the remaining containers at the quay.

As one can see, the current alignment process leads to several inefficiencies for both barge and terminal operators. In the past, attempts have been made to establish a central trusted party that coordinates the activities of all barges and terminals. It turned out that this was not an acceptable solution. Terminal operators compete with each other (as do barge operators) and they are, e.g., reluctant to share information or to give up the autonomy over their operational processes. In addition, there are several other factors that complicate the design of a solution to the barge handling problem. We discuss these factors in Section 1.1.3. In Section 1.1.4 we mention some earlier initiatives to solve the barge handling problem.

The barge handling problem is not only an issue in the Port of Rotterdam, but also in ports such as the Port of Antwerp. Moreover, a similar problem as the barge handling problem can be found for other types of cargo or ships, such as the transportation of liquid bulk by short-sea vessels. These short-sea vessels also have to visit a number of terminals in the port to load and unload liquid bulk products. In this thesis we focus on the barge handling problem and we illustrate it with our case of reference, i.e., the Port of Rotterdam.

In the next section we first give an impression of the urgency of the barge handling problem in the Port of Rotterdam.

1.1.2 The urgency of the problem

The barge handling problem became very urgent for the Port of Rotterdam in 2007 (see Figure 1.1 for a collage of some press releases). The main causes mentioned are the growth in container transportation and the limited capacity at the major terminals in the port. This results in long waiting times for barges at the major terminals (up to 48 hours) and affects the transit times of the containers. The delays made barge operators decide to raise their transport tariffs about 10-20%. A quick solution to the problems is not expected (Lloyd's List, 2007).

Since December 2007, the CBRB (an organization for employers and entrepreneurs in logistics and inland navigation) reports, in cooperation with several barge operators, the so-called *haven verblijfindex* (the port sojourn index). For the Dutch readers: CBRB stands for Centraal Bureau voor de Rijn en Binnenvaart. The index denotes the average amount of time a container barge sojourned in the port per move, i.e., per container transshipment. The average 'haven verblijfindex' in the first half year of 2008 was about 8 minutes for Rotterdam and 7 minutes for Antwerp. By publishing sojourn times, barge

Figure 1.1: Some press releases that appeared in 2006 and 2007 expressing the problems in the Port of Rotterdam

operators try to increase the pressure on terminal operators to improve their services towards barges. An important problem nowadays is the lack of objective registrations. This makes it complicated to show who has caused certain delays. The result is that barge and terminal operators point at each other, without being able to show objectively who is at fault.

In view of the growing competition among European ports, the increasing waiting times at terminals in Rotterdam can affect the attractiveness of the port in the long run. For liner shipping companies it is important that they can transfer containers fast, reliably, and with short transit times. If the Port of Rotterdam gets more and more congested, liner shipping companies might shift their activities to other ports if they offer better services. However, other ports in the Hamburg - Le Havre range also had difficulties in 2007 to process the increasing container flows.

1.1.3 Complicating factors in solving the barge handling problem

To give an indication of the complexity of the barge handling problem, we mention several factors that are relevant for designing a solution. These factors are based on earlier studies (Connekt, 2003; CBRB, 2003; Moonen and Van de Rakt, 2005; Moonen et al., 2007) and on interviews with experts. In the remainder of this thesis we refer sometimes to *port system*, which concerns all parties and activities related to handling barges in the port. We mention the following main complicating factors:

- *Autonomy.* Every terminal and barge operator wants to remain autonomous, i.e., in control of his[1] own operations.
- *No contractual relationships.* There exist no contractual relationships between barge and terminal operators, which means that barge operators and terminal operators cannot contractually force the other to deliver a certain service or charge for poor services.
- *Limited information sharing.* Since barge operators compete with each other (as do terminal operators), they are reluctant to share information that possibly undermines their competitive position.
- *Many different players with conflicting objectives.* In the port system different players are involved, such as terminal operators and barge operators, which have different objectives. This means that a solution must meet the objectives of both groups of players, which are conflicting to a certain extent.
- *Ill-structured - loosely coupled network.* In the port system there is no clear hierarchy in business relations, it is more like a market. Everyone is more or less free to join and to leave. This holds both for barges and terminals. The consequence is that the actors involved in the port system can be different at different points in time.
- *Interdependency of activities.* Since barges visit several terminals in the port, the activities of terminals become interdependent. Delays at one terminal easily propagate through the port and affect the operations of other terminals. This is a kind of dominoes effect, which is re-enforced when barges have planned terminal visits close after each other.
- *Highly dynamic environment.* The problem is dynamic, which means that information becomes known over time. Planning of both barge rotations and quay schedules always needs to be done without full knowledge of the future.
- *Disturbances during execution.* During execution, events such as crane breakdowns or waterway blockages may happen, which affects the operations of barges and terminals. Another cause of disturbance is, e.g., when a barge operator adjusts the number of containers that have to be transshipped at a terminal, close before the barge is processed.
- *Barges and sea vessels are often handled by the same equipment.* At several terminals, barges and sea vessels are processed by the same equipment and crew. However, sea vessels have absolute priority over barges. This means that the scheduling of barges is done such that the handling of sea vessels is not disturbed. Sea vessels that arrive delayed will generally interfere with the barge handling process.

[1] If we refer in this thesis to an unspecified person with *he*, one can also read *she*.

- *Specific constraints in the operations.* Barge operators are restricted in the sequence in which the barge can visit its terminals, due to the stowage plan and capacity of a barge. Terminals, on the other hand, sometimes have restricted opening times. Additionally, certain containers have a closing time. A closing time means that a container has to be at a terminal before a certain time to be shipped with a sea vessel.

- *Position of barges and terminals.* Terminals have a more dominant position in the port than barges. Terminals are obliged (based on their contract with the liner shipping companies) to transship containers to a successive modality. This means that barges will be processed by the terminal, but they have no power to claim specific capacity at the quay. In fact, barges are depending on the terminals to get convenient time slots for handling. This is related to the fact that barge and terminal operators have no contractual relationships.

- *Requests for labor and capacity fixation.* If a terminal operator needs additional labor forces to operate the quay cranes, he can submit a request at the Harbor pool. The Harbor pool is a common initiative of the terminals to create a pool of labor forces that can be deployed flexibly. Requests for labor force have to be submitted to the Harbor pool 24 hours in advance. This means that the terminal fixes its capacity from that moment on for the next 24 hours. Terminal operators therefore want barges to announce their arrival preferably 24 hours in advance as well.

The above mentioned factors need to be taken into account when designing a solution to make it acceptable for the actors concerned. Acceptance is possibly even more important than optimization. Optimization is a difficult concept in this context, since parties have different interests and have to make decisions without full knowledge of the future.

1.1.4 Previous studies on the barge handling problem

The barge handling problem is related to several other problems in the literature. We discuss these in Section 1.5. In this section we describe the attempts that have been made in the past to create a solution for the barge handling problem in the Port of Rotterdam.

The barge handling problem already emerged in the 1980's, when the first containers were transported by barge. The problem, however, became urgent in the last ten years due to the rapid growth of container transportation. A first study to investigate the bottlenecks in the barge handling process was performed in 1998 (RIL Foundation, 1998). One of the results of this study was a covenant between barge operators and the ECT terminals in which they made agreements about the handling of barges at the ECT terminals and about performance levels. In 2000 a next step was taken in a European project called 'Barge Planning Support' to investigate the added value of publishing the quay schedules of terminals on the internet (RIL Foundation, 2000). 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. In the system, the actual arrival and departure times of barges at the ECT terminals are registered. In case of delays, also the cause of the delay is registered. The application also supports the planning of barges at the ECT terminals. 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).

In the past, some ideas have been proposed to establish a central trusted party to coordinate the activities of both terminals and barges. However, for reasons described in Section 1.1.3 this turned out not to be an acceptable solution. Instead of a centralized approach, in 2003 the companies Initi8 B.V. and Havenbedrijf Rotterdam B.V. took the initiative 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 parties involved (Moonen et al., 2007).

One of the contributions of the Approach 1 study was the identification of the causes of the problems in the barge handling process and the way these causes are related/enforce each other. Connekt (2003) reports the following causes (see also Moonen et al., 2007). The first cause is the inefficient use of terminal capacity, rather than the terminal capacity itself. Second, actors include slack in their operations since they expect delays and disturbances, which further worsens the situation. Third, time slots at quays are unilaterally assigned by terminals to barges and do not always correspond with the time requested by the barge. This can result in infeasible rotations, e.g., when barges have planned two terminal visits at the same time. Fourth, a long planning horizon and poor administrative processes worsen the situation further. Summarizing they conclude that terminal and barge operators keep each other 'captive' in a situation of increasing waiting times and decreasing capacity utilization. All actors try to protect themselves against the negative effects resulting from the actions of others, but these protective actions hinder an improvement of the overall situation.

Another contribution of the Connekt (2003) study was the development of a Multi-Agent system to evaluate whether distributed planning is suitable for aligning barge and terminal operations. 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. In Connekt (2003) every barge operator and every terminal operator is equipped with an agent. The basic idea of the system is that barge operator agents (which have a barge visiting the port in the next 24h) and terminal operator agents align their operations every day before a fixed moment (e.g. 7 a.m.) for the next 24h. The resulting appointments between barge and terminal operator agents are not updated during execution, even when major disturbances take place. The way the alignment is done is described in more detail in Section 2.6.1. Based on the results with the Multi-Agent system, Connekt (2003) recommends 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.

Connekt (2003) was a preliminary study to explore whether distributed planning through a Multi-Agent system offers a solution to the barge handling problem. The study and the proposed Multi-Agent system, however, have important limitations. These limitations both concern the optimization and acceptance of the Multi-Agent system. First, the study 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. In Chapters 2 and 3 of this thesis we analyze the proposed Multi-Agent system in more detail.

1.1.5 Outline of the chapter

The outline of the remainder of this chapter is as follows. In Section 1.2 we describe our research goal. Section 1.3 describes the scope of our study and the assumptions made. In Section 1.4 we formulate our research questions and describe briefly the research approach. Section 1.5 discusses several related fields in the literature. In Section 1.6 we describe the scientific and practical contributions of the research and we conclude in Section 1.7 with a description of the outline of the thesis.

1.2 Research objective

The previous sections indicate that the barge handling problem is not easy to solve. A centralized solution is not acceptable for the players concerned and a decentralized solution is complicated because of the constraining demands and opportunistic behavior of the players. Previous attempts to provide a solution to the barge handling problem suggest that decentralized planning is promising and possibly one of the few ways to solve the problem. The problem is highly relevant in practice, since inefficiencies, resulting from poor alignment of barge and terminal operations, lead to significant (in)direct costs. Additionally, these inefficiencies affect the attractiveness of the Port of Rotterdam as a node in global container transportation chains. In this study we develop and explore a decentralized planning system for the barge handling problem. Our research goal can be formulated as follows:

The aim of this study is to develop and evaluate an efficient and effective distributed planning system for the barge handling problem -concerning the alignment of container barge and terminal operations in a port-, and to gain insight in the way the proposed system functions.

With *efficient* we mean the extent to which barge and terminal operators can realize their objectives, given that decisions are made in real-time with minimum communication. *Effective* concerns the realization and implementation of the solution in practice. Even if the developed distributed planning system is efficient, if it is not acceptable to the players in the market, it will not be implemented and can therefore not be effective. We call the distributed planning system also a Multi-Agent system. Evaluation of the distributed planning (or Multi-Agent) system is done with respect to barge operator, terminal operator, and other over-all objectives. We aim to make a comparison between the distributed planning system and a central planning system, i.e., an off-line benchmark. In addition, we consider different scenarios to gain insight in the performance of the Multi-Agent system for different situations.

1.3 Scope of the research and assumptions

To simplify the design of our Multi-Agent system we model the barge handling problem at a certain level of abstraction. To clarify the level of abstraction, we now discuss the scope we apply and the assumptions we make.

In our model we focus on the barge handling problem. We consider only container barge and terminal operators as decision making actors. We assume that both actors are *opportunistic*, meaning that they exploit opportunities for their own benefit with no regard for the consequences for other players. Decisions of both barge and terminal operators have to be made in real-time and we assume that two barge operators never plan rotations simultaneously, but one after another.

With respect to a terminal we make the following choices and assumptions. Terminal operators in our model have to decide about convenient time slots for the handling of barges. We therefore focus on the planning of activities at the quays and we do not model other activities that take place at the terminal, such as the yard planning or the release of containers. Terminals handle both barges and sea vessels. With respect to barges, we assume that terminals only have information about barges that have been announced to the terminal, which is, in our model, on arrival in the port. The service of a barge is not preempted for the service of another barge. We abstract from individual containers and assume that a barge just needs a certain amount of processing time at a terminal to transship its containers. Terminals have fixed capacity and can have restricted opening times, during which barges can be processed. Opening times of terminals are fixed, i.e., no work is done in overtime. The time to handle a container, the mooring time, and the sailing time between terminals are deterministic. Sea vessels arrive with stochastic interarrival times at terminals and processing takes a stochastic amount of time. Arrival times of sea vessels are known to the terminal prior to the planning horizon. Sea vessels have absolute priority over barges.

With respect to a barge we make the following choices and assumptions. Barges arrive over time with stochastic interarrival times. On arrival in the port the barge operator decides which rotation a barge is going to execute, i.e., one must decide on the sequence in which the terminals concerned are visited. We assume that the barge has information about the terminals it has to visit, the number of containers it has to (un)load at each terminal, and the mooring time at, and sailing time between, terminals. It has no information about the state of the network, such as waiting times at terminals. We consider no capacity or stowage constraints for the barge. Each barge visits a terminal only once. We define closing of the terminal as *preemptive downtime*, and the processing of a sea vessel as *non-preemptive downtime*. Preemptive downtime means that the handling of a barge may start before the downtime and finish after it (Schutten, 1998). Non-preemptive downtime means that the handling of a barge may not be interrupted by a downtime.

We assume that all terminal operators have identical objectives. The same holds for all barge operators. We do not explicitly consider disturbances in our model, although we discuss the way disturbances can be introduced in our model and how they will influence the performance of the system. Leaving out disturbances in the operations of barges and terminals, allows for making reliable appointments, since no unexpected delays occur.

1.4 Research questions and approach

To reach our research goal we define a number of research questions. For each question, we indicate the chapter(s) in which the specific question is considered.

1. What is the role of barge hinterland container transportation in The Netherlands and as part of the worldwide container transportation?

Before starting with developing a solution to the barge handling problem, we first consider the role of barge hinterland container transportation in the national and worldwide transportation of containers. We describe developments that currently take place and will possibly affect the barge container transport in the future. In addition, we describe aspects that introduce or explain characteristics of the barge handling problem. We discuss this in Chapter 2.

2. What are key performance indicators of barge operators, terminal operators, and the Port of Rotterdam?

   To evaluate our Multi-Agent system we need to know the (key) performance indicators of barge operators, terminal operators, and the Port of Rotterdam. These indicators have been investigated by other researchers and we describe the findings in Chapter 2.

3. What is an efficient and effective Multi-Agent system for the barge handling problem?

   In Chapter 3 we discuss the design of our Multi-Agent system. In the design of a Multi-Agent system we have to focus specifically on two parts, namely the strategy of players and the interaction protocol. Both parts require careful consideration. First, they determine the extent to which the system can support optimization of the operations of the actors, i.e., the barge and terminal operators, involved. Second, they determine the conditions under which actors can participate. Third, they set the 'rules of the game' and determine, e.g., the robustness of the system against undesirable behavior of certain actors. In Chapter 3 we propose our Multi-Agent system and we specify the system in the Chapters 5 and 6.

4. How to evaluate the performance of our Multi-Agent system?

   One of the aims of this study is to evaluate and to gain insight in the performance of our Multi-Agent system. We therefore have to think about the way we evaluate the performance, how we compare the performance, and which scenarios we consider to gain insight in the functioning of the Multi-Agent system. In Chapter 4 we propose the way we evaluate our Multi-Agent system.

5. How does our Multi-Agent system perform in various port settings?

   After designing the Multi-Agent system and deciding on the way we evaluate its performance, we evaluate the performance of our Multi-Agent system. We consider various (fictitious) port settings that differ in the level of complexity. For example, in simplified port settings we assume that all terminals are identical and that they do not process sea vessels, whereas in more complex port settings we study more realistic situations. We consider general port settings, which are inspired by our case of reference, the Port of Rotterdam. We develop interaction protocols to deal *with the different degrees of complexity. Based on the results we try to gain insight in the functioning of the Multi-Agent system. This research question is studied in the Chapters 5 and 6.*

6. What are relevant extensions to the model?

   Based on the results of research question 5, we aim to consider some extensions to the model that provide more insight in the way the Multi-Agent system functions or that make the model more realistic. We again consider general (fictitious) port settings. The extensions are discussed in Chapter 7.

7. How does our Multi-Agent system perform in the Port of Rotterdam?

   So far we have treated the problem in an abstract way. For this research question we focus specifically on the Port of Rotterdam, to evaluate the performance of our Multi-Agent system in practice. We therefore model a realistic situation of the Port of Rotterdam. In addition, we consider scenarios to perform a sensitivity analysis. We discuss this in Chapter 8.

8. How can we effectively communicate our solution to practice?

   The design of a Multi-Agent system is a first step to solve the barge handling problem. The next step is to implement the system in practice. It is therefore important that there is a practical basis for implementation. In Chapter 9 we explore the use of a game to communicate our research to practice.

Let us briefly describe the research approach we take to design and evaluate our Multi-Agent system. In the design of our Multi-Agent system we aim to develop a system that can be implemented in practice. For that, it is necessary that the Multi-Agent system facilitates (near) optimization of the operations of barge and terminal operators and is acceptable for them as well. Moreover, the system has to facilitate real-time planning to deal with the dynamic nature of the problem. We evaluate our Multi-Agent system by means of simulations. Through simulation we can study the performance of the system as a whole, as the result of the interactions of the parts. We study different variants of our Multi-Agent system to evaluate the value of, e.g., exchanging less information between terminal and barge operator agents. In addition, we construct an off-line benchmark, which is a central optimization algorithm, to make a comparison between the performance of a distributed planning system and a central planning system. In our study we consider general (fictitious) port settings and realistic settings of the Port of Rotterdam. To communicate our solution to practice, we develop a game to see how practitioners respond to the Multi-Agent system we propose. Through this we get insight whether the solution we propose is an acceptable solution in practice.

1.5 Related literature

To the best of our knowledge, the barge handling problem has -except for a few studies mentioned in Section 1.1.4- not been studied in the literature before. The problem, however, relates to other problems in the literature. We discuss these problems in Section 1.5.1. We also discuss briefly the concept of Multi-Agent systems and we give some examples of studies that have investigated these systems. In Chapter 3 we discuss the notion of decentralized planning and Multi-Agent systems in more detail, including references to relevant literature.

A Multi-Agent system is a system in which multiple agents interact to achieve certain goals. For the modeling of the Multi-Agent system we apply algorithms obtained from the literature on, e.g., traveling salesman problems and scheduling problems. In the course of this thesis, we refer more specifically to literature in these fields. We have chosen to apply (where possible) existing and proven methods, instead of developing them ourselves, to concentrate on the performance of the decentralized control system. The reason why is that if we develop new methods ourselves, we first have to test the performance of the parts, before we can combine them and draw our conclusions. A poor performance of the decentralized control system might then be caused by poor methods implemented in one of the parts.

In fact, the barge handling problem can be modeled as a network of queueing systems. We do so in our simulation study. However, the existing theory on queueing networks is not sufficient to analyze the resulting queueing network analytically. Nevertheless, we can use several insights obtained from queueing theory to analyze and understand the barge handling problem. For example, the relation between the utilization of a server and the number of customers waiting in the queue (see, e.g., Gross and Harris, 1998; Ross, 2003).

1.5.1 Related problems in the literature

The most closely related fields to the barge handling problem are the berth allocation problem and the ship scheduling and routing problem, but both fields do not fully capture the characteristics of our problem. Other related problems and fields are the attended home delivery problem, and the hospital patient scheduling problem. We discuss these related problems successively.

The first related problem discussed is the berth allocation problem (BAP). It concerns the assignment of berths to ships (for an overview, see, e.g., Cordeau et al., 2005; Steenken et al., 2004; Stahlbock and Voss, 2008). However, there are two major reasons why the BAP is not applicable to our problem. First, the static and dynamic BAP is usually assumed to have expected arrival times of ships as well as the processing times known at the time the berth plan is made (see, e.g., Imai et al., 2001; Park and Kim, 2003; Cordeau et al., 2005; Imai et al., 2007). In our problem, however, arrival times of barges are uncertain and terminals have to plan their quays taking into account possible future barge arrivals. Second, the BAP considers the operations of a single terminal. In our problem we deal with multiple terminals which are mutually depending on each other due to the barges that, in general, have to visit more than one terminal. This means that the arrival time of a barge at a terminal is also a result of decisions made at other terminals.

A second related problem is the ship routing and scheduling problem (SR&SP). For a recent overview we refer to Christiansen et al. (2004). SR&SPs are considered separately from similar problems in other transportation modes (such as vehicle routing problems), due to the specific conditions under which ships operate (Ronen, 1983). Although our problem can be considered as an SR&SP, there are some major differences with the existing literature. First, the routing of ships is mainly along ports instead of terminals within a port. Although this seems a similar problem at a different level, the level of freedom in the order in which ports are visited might be much more limited due to the geographical dispersion than this is the case within a port. Moreover, we need to take into account the availability of berths, whereas the majority of literature in the SR&SP only focuses on the route the ship sails and assumes that (when a port or terminal is not closed) quay space is available to process the ship. Second, nearly all papers that appeared in the field of SR&SP consider a static problem, i.e., all information is known in advance. Moreover, these problems are solved using a single objective function, which is not an appropriate approach for the barge handling problem. Within the literature of SR&SP, we like to mention a contribution of Christiansen and Fagerholt (2002), who introduce a service-time function to represent the expected service time of a ship during a certain time horizon. The service-time function is influenced by the availability of the port and used to calculate robust ship schedules. In our study we extend the concept of a service-time function and apply it for the construction of barge rotations.

A third related problem is the vehicle routing problem and especially the attended home delivery problem (AHDP) (see, e.g., Campbell and Savelsbergh, 2005; Campbell and Savelsbergh, 2006; Asdemir et al., 2008). In the AHDP, a carrier offers attended deliveries of packages at the homes of customers. An attended delivery means that a customer has to be present when the package is delivered. To optimize the route the carrier travels, Campbell and Savelsbergh (2006) consider incentive schemes to influence the preferred time windows of a customer. We consider a similar problem, although we have multiple carriers. We take explicitly into account the interest of the customer (the terminal), who might want to plan carriers (barges) close to each other in order to decrease the total time needed to be present 'at home'. This makes our problem different from the AHDP.

A fourth related problem is the hospital patient scheduling problem (HPSP).

The HPSP is about the scheduling of patients which can have multiple appointments that have to be scheduled at different resources (see, e.g., Decker and Li, 2000; Paulussen et al., 2003; Vermeulen et al., 2007). Especially in the diagnostic phase, the sequence in which tests need to be done is relatively free. This makes the HPSP comparable to the barge handling problem. However, there is a major difference with respect to the arrival time of barges and patients. In our problem, barges sail from the hinterland to the port and are not willing to delay their arrival time with days or weeks. In the HPSP, the arrival time of patients at the hospital is still variable and part of the decision to make the most convenient appointments from the patients' perspective.

1.5.2 Distributed planning and Multi-Agent systems

The increasing importance of strategic linkages among supply chains and the increasing ease to connect information systems all over the world, also create a need for planning concepts and information systems that support the alignment of operations of different companies. Centralized algorithms, optimizing a problem for a single objective function, often fail to provide a satisfying solution. The reasons why are various (see, e.g., Mes, 2008). First, the algorithms have difficulty to weigh the (conflicting) interests of (competing) companies satisfactorily. Second, companies are reluctant to share information about their operations with one (trusted) party. Third, problems often have a dynamic nature (information becomes known over time) and centralized algorithms can be sensitive to information updates. Fourth, problem instances might become too large to solve in real-time, which makes the algorithms less useful in practice.

Multi-Agent systems allow for distributed planning (see for an introduction Wooldridge and Jennings, 1995). Recall that 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. We refer to Chapter 3 for a more extensive introduction of agents and Multi-Agent systems. Several applications of Multi-Agent systems can be found in transport logistics. See, e.g., Zhu et al. (2000), Böcker et al. (2001), Thangiah et al. (2001), and Kozlak et al. (2004). Some studies explicitly study the interaction between shippers and (road) carriers, see, e.g., Figliozzi (2004), 't Hoen and La Poutré (2004), and Mes (2008).

In an extensive survey on existing research on agent-based approaches in transport logistics, Davidsson et al. (2005) state that especially applications of agents in transportation via water are scarce and most papers have focused on the alignment of activities at a terminal. Contributions in this field are, e.g., Bürckert et al. (2000), Thurston and Hu (2002), Henesey (2006), and Franz et al. (2007). We agree with Davidsson et al. (2005) that most agent-based approaches (especially in the maritime industry) are not evaluated properly and comparisons with existing techniques are rare. Examples of recent papers that apply a quantitative evaluation of their Multi-Agent model are Henesey et al. (2006), Lokuge and Alahakoon (2007), Mes et al. (2007), and Mes, Van der Heijden and Van Hillegersberg (2008). Most papers stay, however, at the level of a conceptual agent model and sometimes draw conclusions about the (expected) performance of the model without presenting experimental results. Among the literature on Multi-Agent based approaches in transport logistics, we found no application similar to the problem we consider (except for the papers already mentioned in Section 1.5.1).

1.6 Contributions

Our study makes both a scientific and a practical contribution. We discuss these contributions successively.

Although a large body of literature on applications of Multi-Agent systems has appeared over the last years, few studies have applied a quantitative evaluation of the proposed Multi-Agent system or even made a comparison with traditional techniques. In our study, we provide more insight in the latter two aspects. In addition, the barge handling problem itself is a nice example of a problem where the application of a Multi-Agent system seems to be the only feasible solution due the specific business constraints. This might be inspiring for other researchers as well. We mention the following more specific contributions:

- We design an efficient Multi-Agent system (MAS) for the barge handling problem in various port settings.
  - Our MAS facilitates effective decision making by barge and terminal operators with respect to the available information.
  - Our interaction protocol supports an efficient negotiation between barge and terminal operators and requires the sharing of only a limited amount of information.
  - Our MAS allows for real-time alignment of barge and terminal operations.
  - Our MAS is designed such that it can be acceptable for the barge and terminal operators and is suitable for implementation in practice.
  - Our MAS is designed such that the propagation of disruptions is suppressed and that the operations of barges and terminals become more reliable.
  - The architecture of our simulation model allows for a connection with DSOL (see Section 4.3.1) to simulate several practical settings in the Port of Rotterdam.
- We evaluate the performance of our Multi-Agent system by means of simulation and we compare the results with a central optimization algorithm.
- We give insight in the value of exchanging different levels of information between barge and terminal operators.

- We give insight in the way our Multi-Agent system functions. We show how the system performance is affected by, e.g., the interaction protocol and by the appointments barge and terminal operators make.
- We develop a realistic model of container barge handling in the Port of Rotterdam and show that our Multi-Agent system might lead to a significant improvement of the current situation.
- We develop an interactive multi–player game as an effective means to communicate our research with practitioners, and we share our first experiences.

Our research has been supported by Transumo (see Section 2.8 for a description of the program). The objective of Transumo is to strengthen the competitiveness of the Dutch transport sector ('Profit'), and to preserve and improve spatial and ecological ('Planet') and social ('People') aspects of mobility (www.transumo.nl). Let us formulate the practical contribution of our study in terms of planet, profit, and people. In our study we contribute to a more efficient use of barge and terminal resources. This leads to a reduction of the operational costs for barge and terminal operators (profit), to less environmental damage (planet), and to a reduction of the costs of hinterland container transportation, which is interesting for the shipper (people). We also contribute to a more reliable barge hinterland container transportation. This allows for optimization of the operations of barge and terminal operators (profit) and higher customer satisfaction (people). Finally, we automate the negotiation between barge and terminal operators, so that they have more time to work on the non-routine tasks and have to spend less time on recovering from all kinds of disturbances that take place (people).

1.7 Thesis outline

The outline of the thesis is as follows (see also Figure 1.2). We start in Chapter 2 by exploring the background of the barge handling problem. We relate the problem to national and global container transportation. We describe in detail two related studies of the barge handling problem. The first one is Connekt (2003), which is the study preceding our research. The second study concerns the key performance indicators of different players in the port. Besides we provide in Chapter 2 some details about the project within which our research has been performed. Chapter 3 introduces the notions of decentralized planning and Multi-Agent systems and formulates requirements for the design of our Multi-Agent system. We assess the Multi-Agent system proposed in Connekt (2003) and we propose our Multi-Agent system. Chapter 4 describes the way we evaluate the performance of our Multi-Agent system. It describes successively the conceptual simulation model, the off-line benchmark, and the scenarios we consider.

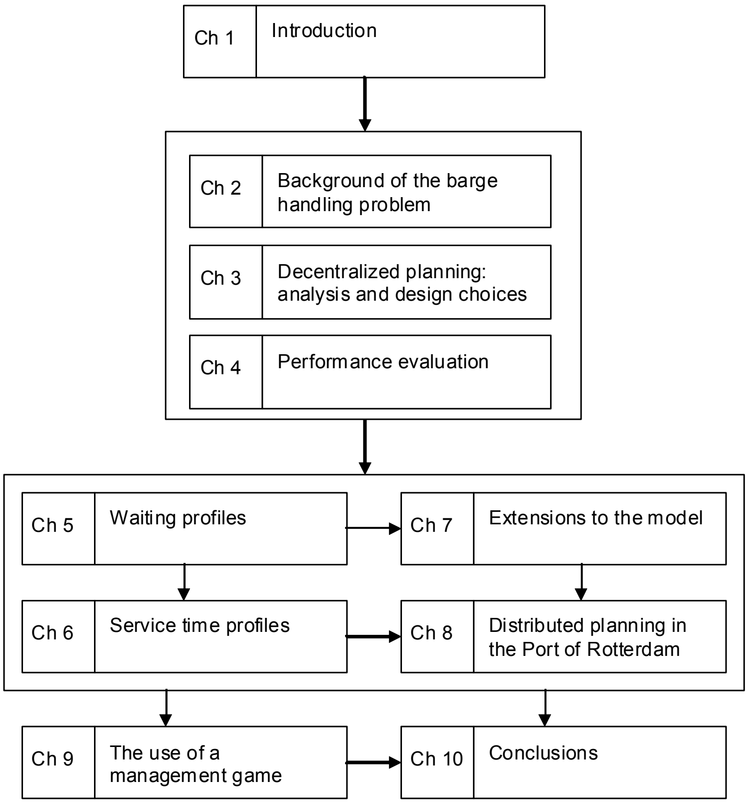


Figure 1.2: Illustration of the outline of the thesis

In Chapter 5 we specify the Multi-Agent system proposed in Chapter 3 and we evaluate its performance in the way described in Chapter 4. We focus in this chapter on simplified (fictitious) port settings to get insight in the functioning of our Multi-Agent system. In Chapter 6 we extend the models of Chapter 5 to be able to deal with more realistic port settings. We evaluate the performance of the Multi-Agent system and provide insight in the way it functions.

Chapter 7 considers two extensions to the model. A basic assumption in the previous chapters is that barge and terminal operators want to make appointments about convenient time slots for handling and are willing to share a certain amount of information. Another assumption is that no disturbances take place during the operations. In Chapter 7 we consider a Multi-Agent system in which no appointments are made and where terminals share limited information. In addition, we discuss the way our model has to be extended to deal with disturbances. In Chapter 8 we make a realistic model of container barge handling in the Port of Rotterdam. We evaluate how our Multi-Agent system, developed in Chapter 5 and 6, performs for this realistic model of the port. We perform sensitivity analysis and give insight in the way the Multi-Agent system functions and the factors that determine its performance.

In Chapter 9 we make a first step towards implementation, by developing a management game as means to communicate our research results and to create a basis for support of our system. We describe the game and share some first experiences from workshops with students and practitioners. Finally, we complete our thesis in Chapter 10 with conclusions and directions for further research.

Chapter 2

Background of the barge handling problem

2.1 Introduction

In this chapter we describe the background of the barge handling problem. In Section 2.2 we describe briefly the history and prospects of global container transportation. In Section 2.3 we try to give insight in the door-to-door transportation of a container by describing the role of liner shipping companies as deep-sea transportation provider. In Section 2.4 we describe the role of the Port of Rotterdam for container *transshipment* and in Section 2.5 the roles of barges in the *hinterland* container transportation. Transshipment is the transfer of a good from one conveyance to another for shipment. The hinterland is defined as an inland area supplying goods to and receiving goods from a port. In Figure 2.1 we illustrate the relation between the Sections 2.3, 2.4, and 2.5. We place these roles in a historical context and describe developments that are observed and expected in the container sector by academia and practitioners.

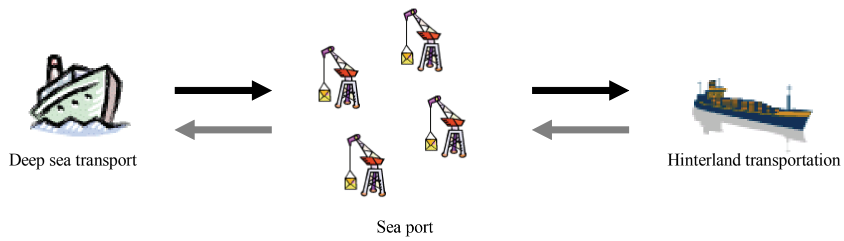


Figure 2.1: The container flows from sea to the hinterland and vice versa

In Section 2.6 we focus on the barge handling problem in the Port of Rot-

	To Rotterdam		From Rotterdam	
Trade lane	*Total*	*%*	*Total*	*%*
Europe	1,902,489	34.3%	2,082,402	39.6%
Africa	127,212	2.3%	122,078	2.3%
U.S. of America	963,014	17.3%	720,352	13.7%
Asia	2,540,953	45.7%	2,303,556	43.8%
Oceania	21,005	0.4%	29,640	0.6%
Total	**5,554,673**	**100%**	**5,258,028**	**100%**

Table 2.1: Origin and destination of containers (TEU) passing Rotterdam. Figures are for the year 2007. Source: www.portofrotterdam.com

terdam by discussing two earlier studies. Finally, we describe in Section 2.7 the key performance indicators we use in our model and in Section 2.8 some organizational details regarding our project.

2.2 Containerized transportation: history and prospects

Since the introduction of the container in the late 1960s, containerized transportation has been widely adopted as transportation means and the flow of containers worldwide has increased ever since. The main trade lanes today are between (East) Asia, (Western) Europe, and (North) America. These trade lanes are operated by several (groups of) liner shipping companies (Notteboom, 2004). To give an impression of the importance of different trade lanes for the Port of Rotterdam, we present in Table 2.1 the origin and destination of containers passing Rotterdam in the year 2007. It is interesting to note the growing imbalance in container flows between Europe and Asia. In 2006 about 53% of the containers shipped from Asia to Europe came back empty (ESPO, 2007). The expectation is that this fraction will increase to 59% in 2008.

The growth that has taken place in container transportation is impressive. The total number of TEU (twenty feet equivalent unit) passing the Port of Rotterdam has grown from 360,000 TEU in 1970 to 10.8 million TEU in 2007. In the period 1996 to 2006, the increase in TEU transshipped is 94%. For the period 2006-2020 an additional growth is expected of 64% to 15.9 million TEU in 2020 (Gemeente Rotterdam, 2004). From the containers transshipped in 2007 in the Port of Rotterdam, about 75% had a hinterland origin or destination (www.portofrotterdam.com). The remaining part is sea-sea transshipment. With respect to the hinterland container flows, the Port of Rotterdam reports a growth of 38% in the period 2003-2007.

In addition to the increase in container flows, also an increase in deep-sea vessel sizes can be observed over the years. Vessel sizes have grown from 1500 TEU in

1980 to about 14,500 TEU in 2006 (ESPO, 2007). The increase in vessel sizes also leads to an increase in *call sizes* at terminals, which has a major impact on the terminal facilities and hinterland transportation (Visser et al., 2007). A call size denotes the number of containers a barge or sea vessel has to load and unload at a terminal.

The increase in container flows has not been without consequences for the port and related facilities. Currently the major deep-sea terminals have reached their maximum capacity, which leads to delays in the transit times of containers. This affects especially the hinterland transportation means (truck and barge), who face long waiting times at terminals (see, e.g., Nieuwsblad Transport, 2007a). To be able to cope with the increased container flows, the Port of Rotterdam plans to build the *Tweede Maasvlakte.* This is a new container terminal area close to the sea where most of the container handling will take place. The transport of these containers to the hinterland via the existing infrastructure, especially road and rail, will form a serious challenge.

2.3 Liner shipping companies

In this section we describe the business and history of liner shipping companies. We also discuss some recent developments in the liner shipping market which impact barge container transportation. The discussion in this section is at a rather aggregate level. Section 2.5.2 describes in more detail the various actors that are specifically involved in the inland shipping of containers.

2.3.1 Business

The core business of liner shipping companies (carriers) is the ocean going transport of containers (CBRB, 2003). However, liner shipping companies sometimes also organize the hinterland transportation of containers. The situation that a liner shipping company only takes care of the ocean transportation is called *merchant haulage*. The merchant, which is the customer of the liner shipping company, then organizes the hinterland transportation. The situation that a liner shipping company organizes the door-to-door transportation of a container is called *carrier haulage*. In that case the liner shipping company organizes both the hinterland and the ocean transportation of the container. The two situations, carrier and merchant haulage, result in different contractual relationships between the liner shipping company, the merchant or the shipper, the terminal operator, and the barge operator.

The terminal is always contracted by the liner shipping company. In this contract the liner shipping company and the terminal make agreements about the transshipment of a certain amount of containers from a deep-sea vessel on a successive hinterland modality (truck, rail, or barge) and vice versa. The barge operator, on the other hand, is contracted by a carrier (in case of carrier haulage) or by a merchant (in case of merchant haulage). In both cases, carrier and merchant haulage, no contractual relationships between terminal and barge operators exist. This means that barge operators do not pay the terminal operator for the transshipment of containers, nor can both parties charge each other in case agreements are not carried out satisfactory.

2.3.2 History

Liner shipping companies have played an important role in intercontinental trade and the adoption of containerized transportation. Until the 1980s, the profits of liner shipping companies were relatively safe, since the powerful liner conferences looked after freight rates. These liner conferences were started in 1875. The advent of steam ships gave ship owners the possibility to offer regular scheduled services. To protect their business, ship owners established cartels (liner conferences) to control freight rates and the entrance of new shipping companies. These conferences have granted protection from national and international anti-trust legislations, because of their importance in realizing freight rate stability and traffic regulation (Franck and Bunel, 1991).

However, by the end of the 1970s the competition on intercontinental trade became fiercer. The main reasons were the entrance of low-cost (Asiatic) fleets and the introduction of the container (Franck and Bunel, 1991). Liner shipping companies have started to cooperate in the form of alliances to stay competitive in the changing market structures. These alliances consist of two to five liner shipping companies and change frequently in composition (see, e.g., De Souza et al., 2003).

Motivations for designing alliances are several, but mainly to secure economics of scale, to achieve critical mass in the scale of operation, and to maintain local and global market share (Glaister and Buckley, 1996; Ryoo and Thanopoulou, 1999). In today's business, liner shipping companies have developed a strong focus on reducing costs, while maintaining service levels in terms of sailing frequency, number of ports visited, reliability of the schedule, and transit times (Notteboom, 2004). Short transit times are necessary to offer attractive services, and reliable schedules are important to guarantee transit times. Notteboom (2007) states that a delay of a large container vessel holding 4,000 TEU with one day amounts for at least 57,000 euro.

2.3.3 Recent market developments

To stay competitive, liner shipping companies have restructured their business in different ways. A first development is the increasing size of deep-sea vessels as sketched before. The bigger and more fuel economic vessels have resulted in a reduction in the cost per TEU per mile (Notteboom, 2004). A second development is the acquisition of dedicated terminals at major nodes in the carrier's service network. The main reason is to secure transit times and schedule reliability, by guaranteeing transshipment capacity in ports (Notteboom, 2007). AP Møller - Maersk has, e.g., besides liner ships also several dedicated terminals (APM terminals), which means that APM terminals primarily handle containers shipped with Maersk Line. A third development is the reduction of port visits in, e.g., the Hamburg-Le Havre range. This means that carriers visit one port in a certain region and ship containers with another port destination in the same region over land. This results in large container flows between, e.g., Antwerp and Rotterdam (CBRB, 2003; Notteboom, 2007).

A fourth development is that liner shipping companies try to increase the percentage of carrier haulage at the costs of merchant haulage. This development has several reasons. First reason is, according to Notteboom (2004), that in a typical intermodal transport, the share of inland transportation costs in the total costs of container shipping ranges from 40-80%. Although liner shipping companies see this opportunity to reduce costs, there is little room to increase income out of inland logistics. If carrier haulage tariffs are higher than the open market rates, the merchant haulage becomes more attractive (Notteboom, 2002). A second reason is that in case of merchant haulage, carriers lose control of and information on their containers. Containers are mostly owned by the carrier, i.e., the liner shipping company, and the repositioning of empty containers is a significant cost factor. By choosing strategically positioned hubs of terminals in the hinterland, carriers try to gain control over the repositioning costs. However, the increasing number of inland terminals and the relatively large share of merchant haulage hinder the carriers' insight in the location of their containers and the dwell time at customers. Moreover, carriers are not eager to impose financial penalties on clients that hold containers too long, as they fear to upset and possibly lose the customer (Notteboom, 2004).

2.4 The Port of Rotterdam

In the previous section we looked at liner shipping companies and their role in the global shipment of containers. In this section we specifically focus on the Port of Rotterdam as our major case of reference. We explain the role of this port in the global container transportation and describe some developments that may affect the position of the port.

The Port of Rotterdam is the sixth largest container port in the world and the largest container port of Europe in 2007 in TEUs transshipped. Hamburg and Antwerp are respectively the second and third largest container port in Europe (www.portofrotterdam.com). Although Rotterdam has some major benefits over Hamburg and Antwerp, e.g., the accessibility of the port for (the largest) sea vessels and the connection to different hinterland transportation modes, it has to a large extent the same hinterland as Antwerp, Hamburg

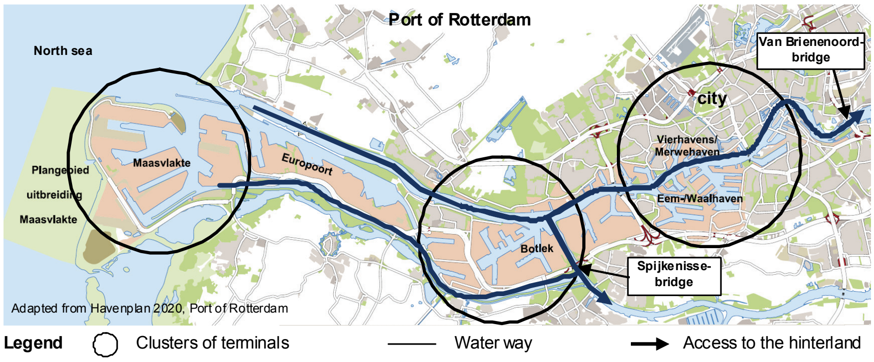


Figure 2.2: Layout of the Port of Rotterdam. Highlighted are the important container terminal clusters, the main waterways, and the hinterland access points. Adapted from Gemeente Rotterdam (2004)

and Le Havre. This leads to fierce terminal competition between the ports (Haralambides, 2002).

The Port of Rotterdam has a specific layout, since it has been developed and extended over time along the river towards the sea. In Figure 2.2 we depict the layout of the port and the three clusters of container terminals that can be distinguished. The port has become very large in the last decades. To sail, e.g., from the Van Brienenoord-bridge to the Maasvlakte (right-side of the figure to the left-side) takes about three hours for an average barge. Barges normally enter the port via the Van Brienenoord-bridge or the Spijkenisse-bridge and use the marked waterways to move between the regions. The reason why we distinguish regions within the port is that sailing between regions takes relatively much time (about one to two hours), whereas sailing from one terminal to another terminal in the same region is less time consuming (about 20 minutes). For barges it is therefore less important in which sequence terminals are visited within a region, than the sequence in which the regions are visited. About 50% of the barges sailing between Antwerp and Rotterdam enter and leave the Port of Rotterdam via the Spijkenisse-bridge, all other barges mainly enter the port via the Van Brienenoord-bridge.

The municipality of Rotterdam writes in *Havenplan 2020* (Gemeente Rotterdam, 2004), a vision on the future of the Port of Rotterdam, that changes in the power balance in the container sector are expected. Increase in scale, concentration, and chain integration lead to the appearance of a small group of global players maintaining connections between own terminals with own ships. The municipality of Rotterdam expects that competition will be more among networks of global players. Rotterdam can possibly become a link in these networks. In addition, Slack et al. (1996) state that -in general port settings- the loyalty of port clients cannot be taken for granted. There is a constant risk

Port	Barge	Rail	Road
Rotterdam	36%	13%	51%
Antwerp	33%	8%	59%
Hamburg	1%	29%	70%
Le Havre	6%	8%	86%

Table 2.2: Modal split for container transport in 2007 for the ports of Rotterdam, Antwerp, Hamburg, and Le Havre. Source ESPO (2007)

that former clients leave the port, not because the provided infrastructure is insufficient, but because of new service networks and new partnerships of the carriers.

To stay part of the global network of liner shipping companies, it is important that the Port of Rotterdam distinguishes itself from competing ports in Europe (CBRB, 2003). This means that, among others, hinterland accessibility becomes increasingly important (Konings, 2007). Basic requirements for a competitive hinterland system are services that are cost-effective, reliable, and have short transit times (Visser et al., 2007). In this respect, Rotterdam has a competitive advantage over Hamburg and Antwerp. The port is very well connected to the main European waterways and has a good connection to the rail and road network. Hamburg, e.g., has limited access to the main European waterways and therefore has a low share of inland barge container transportation.

The importance of different transportation modes can be expressed by the modal split, i.e., the share of different transportation means in hinterland container transportation. In Table 2.2 we denoted the modal split in 2007 for the ports in the Hamburg-Le Havre range as reported by ESPO (2007). The modal split in the Port of Rotterdam has been relatively stable over the past five years, although in these years the number of shipped containers increased with 38% (www.portofrotterdam.com). This results in increasing congestion on the road and rail network. Increasing the capacity of a rail and road network is hard and takes several years. To increase rail capacity, the Dutch government recently developed a dedicated freight line connecting Rotterdam with the hinterland of Germany, called the *Betuwelijn*. This line is taken into operation in 2007. However, this increase in capacity is probably insufficient if the container flows continue to increase. Barge transportation can then be an attractive alternative, since it has lower freight rates, is less harmful to the environment, has no congested infrastructure, and the transport capacity can relatively easily be increased (Visser et al., 2007). The municipality of Rotterdam expects that the share of barge transportation will grow from 30 to 40% in 2020 (Gemeente Rotterdam, 2004). They consider barge transportation as an attractive alternative to relieve the congested roads.

If the vision of the municipality of Rotterdam comes true, then the barge sector will face a growth in the coming years resulting from i) a *modal shift* and ii) an increase in container flows. A modal shift means that cargo flows are shifted from road to rail or water. The question is whether this growth can be realized given the current situation. Deep-sea terminals have reached the limit of their transshipment capacity, which leads to significant waiting times for trucks and barges over the last years. For barges this results in uncontrolled sojourn times in the port. Press-releases in 2007 (cf. Figure 1.1) even report waiting times for trucks up to six hours and for barges varying from 24 to 72 hours (see, e.g., Nieuwsblad Transport, 2007a).

To stay competitive as Port of Rotterdam it is important that sustainable solutions will soon be developed for these problems.

2.5 Barge hinterland container transportation

We now focus on barge hinterland container transportation. We first discuss briefly the history, describe the barge sector and the market structure of inland container shipping, and we finish with discussing some recent developments.

2.5.1 History of barge container transportation

Barge container transportation has a relatively short history. After the introduction of the container in the late 1960s, barge container transport first started in the late 1970s. From then on barge container transportation has increased from 60,000 TEU in 1980 (CBRB, 2003) to more than 2 million in 2006 (www.portofrotterdam.com). This growth has come together with the development of inland terminals along the main European waterways. Interestingly, trucking companies have stimulated the development of inland terminals to offer more and cheaper transport services to their customers. The inland terminals are, besides transshipment, also used for the storage of both full and empty containers (CBRB, 2003).

The main Rotterdam related hinterland for barge container transportation can be geographically divided in four areas or markets (CBRB, 2003; Konings, 2007):

- *The Rhine river trade*

  The Rhine gives access to several industrial areas in Germany, France and Switzerland. Barge operators offer almost daily services to the hinterland. Three different trajectories are usually distinguished; the Lower Rhine river (round trip of a ship every 1.5-2 times a week), the Middle Rhine river (round trip of a ship is usually once a week) and the Upper Rhine river (round trip of a ship is usually once per two weeks). See also Figure 2.3.

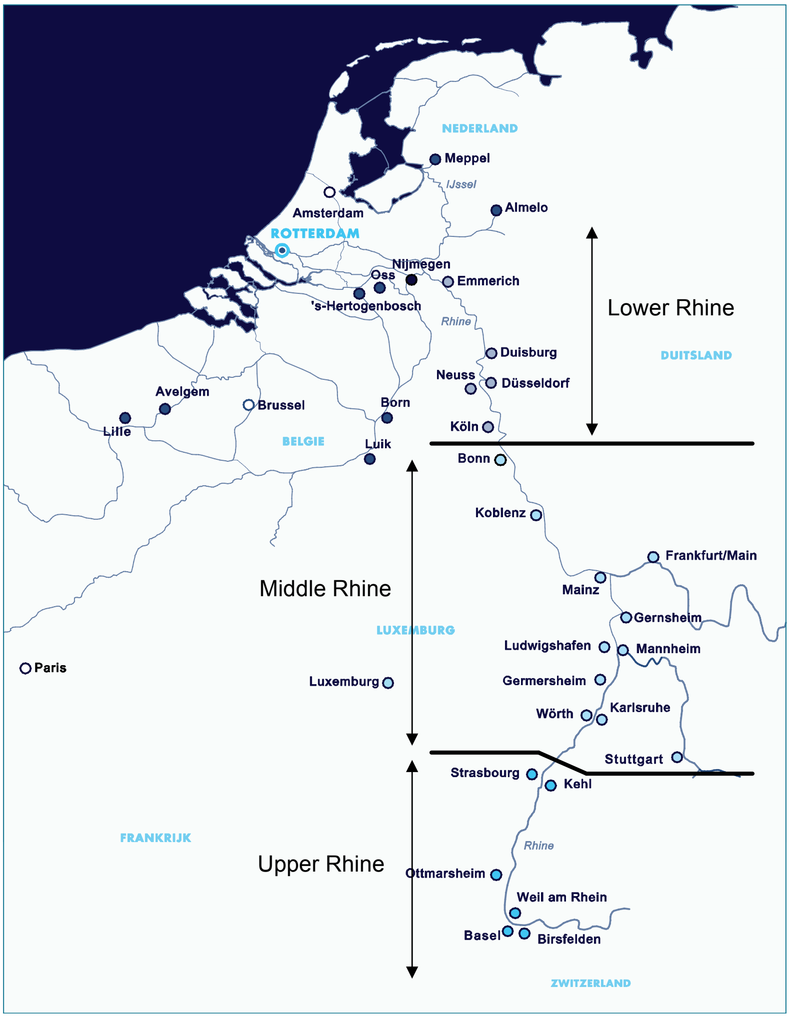


Figure 2.3: Different sub markets in the Rhine container transport. Picture adapted from Gemeente Rotterdam (2004)

- *Rotterdam - Antwerp trade*

  The transport of containers between Rotterdam and Antwerp is the result of a strategic decision of the carrier to visit either Antwerp or Rotterdam with a deep-sea vessel. This means that sometimes containers need to be transported over land to the other port. The barge transit time from Rotterdam to Antwerp (and vice versa) is about ten to twelve hours. Barges are relatively large and visit only a few terminals in the Port of Rotterdam (compared to the Rhine trade). Call sizes at terminals are therefore relatively high. Almost all trade is carrier haulage.

- *Domestic trade*

  For a long time, barge transportation of domestic containers was considered as unattractive. However, since the 1990s the domestic barge container transportation has grown as well as the number of inland container terminals. Most of the domestic trade is merchant haulage. Some of the terminal operators are barge operator as well.

- *Inter terminal transport*. A low share of the containers (about 2%) are transported between terminals in the port. This is partly done by regular barges and partly by the *waterbakfiets,* a dedicated barge for inter-terminal container transportation.

Rotterdam has traditionally a strong position in the Rhine trade (and the sub-markets), whereas Antwerp is mainly strong on the Upper Rhine market (Notteboom, 2002). In the Rhine river trade, barges visit usually about three to five terminals in the hinterland and about ten terminals in the Port of Rotterdam (Konings, 2007). The reason why a barge has to call at several port terminals, is that containers (collected in the hinterland) have to be shipped by specific sea vessels which are handled at specific terminals.

In contrast to the other modalities (road and rail) barges are handled at the same side of the terminal as sea-going vessels. Some terminals have dedicated barge quays and cranes, but most terminals use the same equipment for barges as they use for loading and unloading sea vessels. In practice this means that barges are handled in between the handling of sea vessels. When the modal split of barge container transport was low (at the start of the 1970's) this was not a problem. However, today about 36% of the containers are shipped by barge to the hinterland. This means that the processing of barges impacts the operations of the terminals significantly. Terminals, however, give priority to the handling of sea vessels, since these are paying for the terminal operations. Barge operators have no contractual relationship with the terminal and are processed with low(er) priority at the terminals. During several discussions with barge operators and barge companies we got the impression that they feel (still) not to be taken seriously by the main terminals. The result is uncertain waiting and handling times at the terminals.

CEMT class	#TEU	#Barges	Fraction of barges
II	24	776	20%
III	48	1027	26%
IV	120	1039	26%
Va	208	965	25%
VIb	470	136	3%

Table 2.3: Examples of the active barge fleet in different CEMT classes (per 1/1/2000). Derived from CBRB (2003)

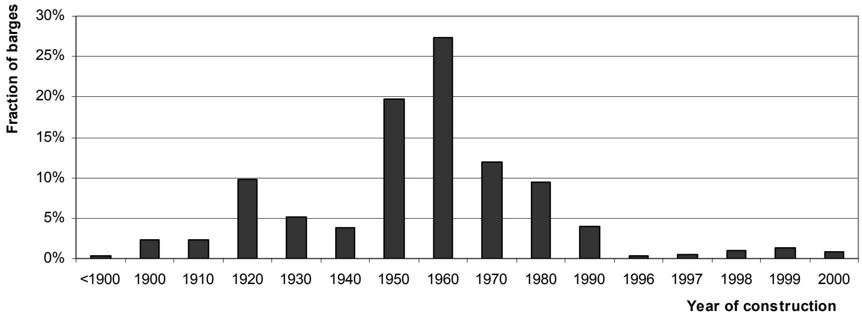


Figure 2.4: Year of construction of the active Dutch barge fleet in 2000. Source: CBS and CBRB (2003)

The active barge fleet is defined as all vessels that participate in national and/or international shipping on inland waterways in a given year (source: CBS). The active Dutch barge fleet consists of barges that can sail different classes of waterways, so-called, CEMT classes (CEMT, 1992). CEMT classes are a European standard of the waterways dimensions and the maximum sizes of ships (and push-tug combinations) that can navigate these waterways. In Table 2.3 we give examples of the capacity of barges that can navigate waterways of different CEMT classes. In addition, we give an indication of the active barge fleet per CEMT class. The Dutch government recently invested in the improvement of waterways to make them navigable for larger vessels.

Barges have a long life span as becomes clear from Figure 2.4. The figure presents the year of construction of the active barge fleet in 2000. The long life span impacts the speed of renewal of the fleet.

2.5.2 The market structure of inland container shipping

The market structure of inland container shipping is rather complicated. However, to understand the barge handling problem it is important to have some knowledge about it. In this section we describe different actors involved in the inland shipping of containers, their tasks, and their mutual contractual

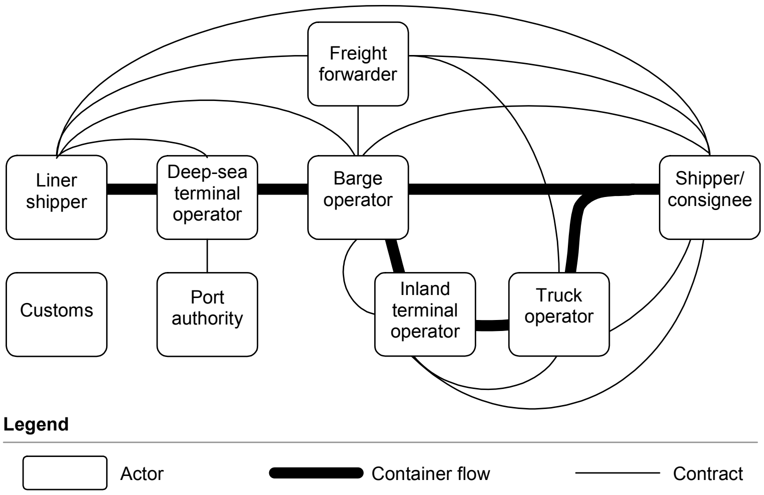


Figure 2.5: Actors involved in the hinterland container transportation and their mutual contractual relationships. Adapted from Van der Horst and De Langen (2008)

relationships.

In Figure 2.5 we depict the most important actors involved in the inland shipping of containers. The figure is adapted from Van der Horst and De Langen (2008). We distinguish the following actors. For every actor we give a brief description of his role or task:

- *Liner shipping company.* A company whose primary business is the ocean-going transportation of containers. See for a more extensive description Section 2.3.
- *Deep-sea terminal operator.* A company that operates a deep-sea terminal and offers services to transship and temporarily store containers. A deep-sea terminal is located in the port. When we refer to *terminal operator* in the remainder of this thesis we mean the deep-sea terminal operator. In practice, a terminal operator can operate more than one terminal. Without loss of generality, we assume in the sequel that every terminal operator operates one terminal and every terminal is operated by one terminal operator.
- *Barge operator.* A company that offers and organizes barge container transportation services to and from the hinterland. These companies usually do not own barges themselves, but contract barge companies,

which are companies that own and operate barges. In practice, barge operators can operate more than one barge. Without loss of generality, we assume in the sequel that every barge operator operates exactly one barge and every barge is operated by one barge operator.

- *Inland terminal operator*. A company that offers services to transship containers from barge to truck or rail (and vice versa) and to temporarily store containers. An inland terminal is located in the hinterland.
- *Truck operator*. A company that offers transportation of containers by truck.
- *Shipper/consignee*. The organization (or person) where the container is sent to, or the organization (or person) that owns the freight or initiates the container transportation.
- *Freight forwarder*. A company who arranges container transportation on behalf of a shipper.
- *Customs*. The organization responsible for inspecting freight that enters or leaves the country.
- *Port authority*. The organization that leases sites to port-related businesses and which is responsible for an efficient and safe handling of shipping traffic. The organization also takes care of the port infrastructure and other facilities in the port area.
- *Port InfoLink* (not in Figure 2.5). The organization that develops and exploits the IT infrastructure for the Port of Rotterdam.

Let us describe some of the actors in more detail. The role of the barge operator is to organize hinterland container transportation by barge. Most barge operators contract barge companies to sail fixed round trips between Rotterdam and the hinterland. Some barge operators own inland terminals, giving them a stronger position in the hinterland. Competition among barge operators is fierce and the profits are relatively low, due to the fact that little value is added (CBRB, 2003). To be competitive with other modalities it is important for a barge operator to offer frequent and reliable services with enough capacity and at low cost (CBRB, 2003). On the Rhine river and Rotterdam-Antwerp trade some barge operators cooperate by sharing barge capacity. In this way they can offer frequent services with enough capacity. These barge operators have their own central planning system and can operate their ships efficiently (CBRB, 2003). One of the tasks of the barge operator, which is relevant in our study, is the alignment of barge operations with inland and deep-sea terminals.

The (deep-sea or inland) terminal takes care of the transshipment of containers between different modes (Vis and De Koster, 2003). The transshipment of containers usually involves the temporary storage of containers to overcome timing and size differences between modes (Kumar and Dissel, 1996). Terminals can be accessed by water and road. In addition to storage, some terminals offer services to repair or clean containers (CBRB, 2003).

In Figure 2.5 we depict, besides the actors, also their mutual contractual relationships. The contracts that are necessary to ship a container depend on whether the shipper chooses for carrier or merchant haulage (see Section 2.3.1). Suppose a shipper chooses for merchant haulage, then he has to arrange the hinterland transport and the ocean transport of the container himself. The shipper therefore contracts directly a barge or truck operator, and the liner shipping company. Another option is to ask a freight forwarder to arrange the transportation. The freight forwarder then contracts a barge or truck operator, and the liner shipping company. In case of carrier haulage, the shipper contacts the liner shipping company, who arranges the hinterland and ocean transportation. Depending on who organizes which part of the transportation, several contractual relations are established.

2.5.3 Developments in the barge container sector

The barge container sector has been developing rapidly over the last years. In this section we describe some trends and some specific initiatives to reduce waiting time at terminals.

Trends and developments

The *Bureau Voorlichting Binnenvaart* (agency for innovation in the barge sector) mentions four major trends in a publication, entitled *The Power of Inland Navigation* (BvB, 2007):

- *Strategic alliances*. They expect that most barge companies in the future will remain small and medium-sized businesses operating about two to three barges. However, they also expect that barge companies will start strategic alliances with terminals, truck operators, or other barge companies, to gain new business and improve the quality of the offered services.

- *Increase in scale.* The increase in scale, especially vessel size, is expected to continue since the limits of the vessels and waterways are not reached yet.

- *Automation of transshipment and ship operations.* The *Bureau Voorlichting Binnenvaart* expects that the transshipment of containers will become fully automated in the coming years and that by the year 2020 also vessels will be navigated fully automatically.

- *Reduction of emissions.* The barge container sector continues to reduce the emissions. They claim that in 2006 a barge was on average six times cleaner than road and about two times cleaner than rail transport.

Other trends or developments concern the development of terminals serving as transshipment point of containers from barge to rail or road, the efforts put by carriers to increase the share of carrier haulage at the cost of merchant haulage, the increasing importance of barge hinterland transportation, and the increasing number of inland terminals (see, e.g., Notteboom, 2002; CBRB, 2003; Konings, 2007; Notteboom, 2007).

Another development we like to mention is the use of transponders in barges to do objective registrations of arrival, waiting, and handling times at terminals. Today there is not such an objective registration, resulting in discussions whether the times reported by the barge shipper or the terminal operator are true. In the barge sector there is a clear need for these objective registrations to make clear how barges are processed by terminals. By the end of 2008 a pilot project will be started concerning about 40 barges (source: private communication to the Port of Rotterdam). The expectation is that in the future all barges will be equipped with a transponder.

Initiatives to reduce the waiting times

We mention two specific ideas to reduce the waiting time at terminals and to deal with the increasing container volumes. Later in our study we consider them in more detail (see Chapter 8).

The first idea is the development of a large inland terminal close to the Port of Rotterdam (<50 km), a so-called *transferium*, where containers are transshipped from trucks to barges and vice versa. The idea is that barges bring containers from the transferium to the terminals in the port and vice versa, thus reducing the number of trucks that use the congested road infrastructure in the port (Konings, 2007; De Binnenvaartkrant, 2008). Realizing a transferium will lead to an increasing number of barge movements in the port.

The second idea is a large transferium far in the hinterland, e.g., Duisburg. Barges bring (unsorted) containers in large vessels (consisting of four boats, pushed by one barge) directly from a sea terminal to the inland transferium. At the transferium the containers are sorted and shipped to their hinterland destination.

Although the idea of a transferium (close to the port or far away) is not new, it never seemed to be economically and logistically interesting for truck and barge operators. However, recent publications show that these ideas are still viable and subject of research in science (Konings, 2007) and practice (De Binnenvaartkrant, 2008).

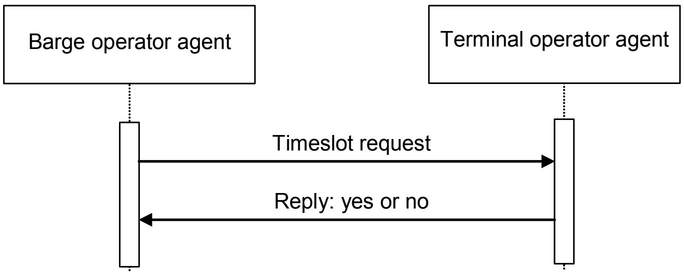


Figure 2.6: Sequence diagram of the communication between a single barge and terminal operator in APPROACH 1

2.6 Barge handling problem

In Chapter 1 we introduced the barge handling problem. In the previous sections we sketched the context of the problem and developments that will affect the barge container sector. In this section we describe in more detail some earlier or related studies to the problem. In Section 2.6.1 we take a closer look at the solution developed in APPROACH 1, the predecessor of our study. In Section 2.6.2 we describe the findings of a study of Van Groningen (2006) concerning key performance indicators of barge and terminal operators.

2.6.1 A more detailed look at Approach 1

In Section 1.1.4 we already mentioned the study, called APPROACH 1, which is an explorative study into the feasibility of a Multi-Agent based control for the barge handling problem (Connekt, 2003). The aim of the study is to create feasible (not necessarily optimal) plans for barges and terminals. They propose an off-line planning system, where plans are made once in every 24h for the next 24h and not updated during execution. In this section we describe the solution, proposed in Connekt (2003), in more detail and we focus especially on the way rotations and quays are planned.

The planning process is started at a fixed moment, e.g., 7 a.m., to align terminal and barge operations for the next 24 hours. At this moment, announced by a special timer agent, the barge operators, which have a barge that visits the port during the planning period, start a *rotation planning process*. The basic interaction protocol underlying this process is depicted in Figure 2.6. A barge operator determines the sequence of terminal visits (a rotation) and the corresponding (expected) arrival times. These arrival times are sent as request for handling to the terminals, which in turn reply with respective *yes* or *no* if a request for handling is ok for the terminal or not.

The rotation planning process is visualized in Figure 2.7. At the start (step 1) of the process the barge operator agent collects information about the barge

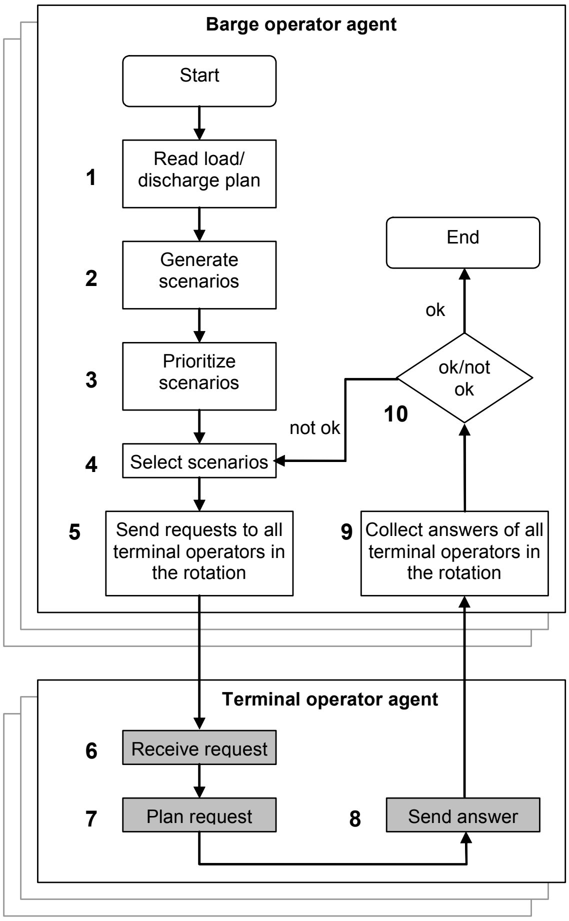


Figure 2.7: The rotation planning process, relating the different processes at the barge and terminal side. After Connekt (2003)

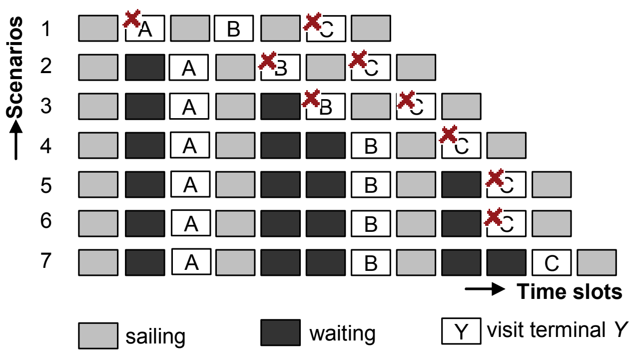


Figure 2.8: Possible sequence of scenarios a barge operator agent has tried to establish a rotation. Example copied from Connekt (2003)

it has to plan a rotation for. This information comprises the planned port arrival and departure times, the stowage plan, the terminals that have to be visited etcetera. In step 2 the agent starts to generate scenarios. A scenario is a possible rotation, determined by the sequence in which terminals are visited and the expected arrival times at the terminals. The agent generates different scenarios by varying the sequence of terminal visits or delaying arrivals at a terminal. Unfortunately, Connekt (2003) does not describe how scenarios are exactly generated. In step 3 the scenarios are sorted according to certain criteria, such as the total sailing distance. The best scenario is on the top of the list. Again, no clear description of the prioritization step is given in Connekt (2003). In step 4 the barge agent selects a scenario and prepares requests for time slots at the terminals. In step 5 the time slot requests are sent to the terminal operator agents. In step 6 the request is received by a terminal, together with requests of other barge operators. In step 7 the terminal operator agent tries to plan the barge at the requested time slot. If this is successful, a confirmation is prepared, if not, a refusal. In step 8 the answer (confirmation or refusal) is sent to the concerning barge operator. In step 9 the barge operator collects the answers of all terminal operator agents. In step 10 these answers are evaluated. If all requests are confirmed then the rotation planning process for this particular barge has come to an end. If one or more terminals have sent a refusal, the barge operator has to go to step 4 and select another scenario and send new requests to all terminals until all terminals have agreed with a specific time slot. It is not clear from the report Connekt (2003) what happens if the number of scenarios that are generated for a specific barge turn out not to be enough. They implicitly assume that this situation never occurs.

In Connekt (2003) an example is given of a sequence of possible rotations that a barge agent tries to establish successively (see Figure 2.8). The barge started with scenario 1, resulting in a refusal of terminal A and C. In response the barge operator retracted all requested slots and selected a new scenario (2) by adding waiting time in the rotation (before terminal A). However, now terminal B and C refused the proposed time slot. The barge continued to add slack and postpone terminal visits until a rotation was found that was confirmed by all terminals. In the example the agent did not resequence terminal visits, but this could be done as well. Surprisingly scenarios 5 and 6 are similar. From the report is does not become clear why.

In Connekt (2003) no experimental settings and results are presented. The application has been evaluated, however, in a workshop with practitioners (Moonen et al., 2007). The aim of the workshop was to make a comparison between the manual (current way) of planning and the Approach 1 prototype. Both used the same data set of barges and terminals. This data set comprised eight different container terminals. In total 22 barge rotations had to be planned for a period of 24h. For a more detailed game description we refer to Moonen et al. (2007). The manual planning exercise resulted in a lot of double bookings at terminals and late arrivals of barges. The Approach 1 prototype, on the contrary, was able to generate a feasible solution. The practitioners were, according to Moonen et al. (2007), shocked to see the magnitude of the planning problem caused by the manual planning process and they were eager to know more about the Approach 1 prototype and the possibility to implement it in practice. However, also some critical remarks were made. First, the outcome of the Approach 1 prototype contained illogical routes (routes with longer sailing times than needed), due to visiting terminals in a different order than a human planner would allow. Second, there were some critical remarks regarding the role of the barge planners after introduction of the system. The participants wondered whether these planners will disappear or that their task will change. Finally, the participants made remarks regarding the possibility to include the stowage plan, rules to give containers (of certain important customers) high priority at the terminals, and other business rules in the creation of rotations and quay schedules.

The workshop clearly illustrated the problems that arise in the current situation and the improvement that may be realized using a distributed planning system. Moonen et al. (2007) report that the practitioners were highly interested in the implementation of the distributed planning system in practice. The company Initi8 B.V. has developed a commercial application, called Synchron8, based on the illustrated prototype. However, the application has not been implemented yet, i.e., up to 2008.

The distributed planning system as proposed in Approach 1 has room for improvement. We elaborate on this in Section 3.4 from a Multi-Agent system point of view. For now, let us confine ourselves to some practical consequences. First, the distributed planning system is not able to cope with the dynamic nature of the problem. Information is not always known in advance for a period of 24h, which means that plans, unless they are very robust, need revision after some time. Second, the planning system focuses on feasible instead of optimal (or best) plans for each of the actors in the network. This is not desirable from a single actor's point of view, especially, when an actor discovers that he can make a better plan himself without using the system. Third, the orchestration of the agent communication can be crucial. If every barge operator agent can start talking to every terminal operator agent at the same time (as suggested in Connekt, 2003) then terminal operator agents have to answer several barge operator agents, more or less, at the same time. Based on these responses, all barges start to reconsider their rotations which leads to new requests. Since all barge operator agents do this at the same time, it might be that a time slot for which a negative response was given by a terminal at time t, is available again at time $t + \epsilon$ ($\epsilon > 0$) when another barge operator agent has cancelled his temporary reservation. This does not lead to satisfying results and creates a tremendous communicational burden.

2.6.2 (Key) performance indicators and the way actors can influence these

To design the agents in a Multi-Agent system we need to understand what the main objective is of every actor in the system. Van Groningen (2006) did an extensive study into the key performance indicators (KPIs) of the barge operators, terminal operators, and the Port of Rotterdam. He interviewed several barge and terminal operators, and employees of the Port of Rotterdam. The results are described in Van Groningen (2006). Based on these interviews he defined (key) performance indicators per actor (related to the barge handling problem) and showed how these (key) performance indicators are interrelated and enforce each other.

In our study we are mainly interested in the *key* performance indicators per actor. In future research we aim to differentiate the KPIs per barge and terminal operator. Let us describe the barge operator, the terminal operator, and the Port of Rotterdam successively based on the report of Van Groningen (2006).

Barge operator

The main objective of a barge operator is, according to Van Groningen (2006), to minimize possible delays in the sailing schedule. The sailing schedule determines the time at which a barge is planned to be in the port and at its hinterland destinations. The reasons why barge operators have difficulty to stick to their sailing schedule are several. We combine the findings of Van Groningen (2006), Moonen et al. (2007), and interviews we did ourselves:

- *Rotations are not feasible at the time they are constructed.* Infeasible means that, e.g., two terminal visits are planned at the same time, that terminals have double booked a time slot, or that terminal visits are planned too tight to each other. This effect is reinforced by uncertainties and disturbances during the execution of a rotation, which results in uncertain arrival and departure times of barges at terminals.

- *Restricted opening times of terminals.* Terminals sometimes have restricted opening times. For barges it is important to visit these terminals when they are opened. However, sojourn times at other terminals are uncertain, which means that it is hard to estimate whether a certain terminal can be visited during opening hours. If a barge arrives when the terminal is closed, it has to wait until the terminal opens again.

- *Processing time of a barge at the terminal.* The processing time of a barge depends on the speed of the terminal cranes. Terminal cranes can usually handle about 20 containers per hour. Barge shippers report, however, speeds varying between 2 and 20 moves per hour. Processing time is therefore an uncertain factor in the planning.

- *Uncertain waiting times at terminals.* Waiting times at terminals are uncertain. Barge operators do not know the start of the barge processing for sure, until the barge is really processed. This results in additional uncertainty in the planning of a rotation.

- *Capacity of the barge.* If a barge operator decides to visit terminals in a different order, it might happen that a barge has not unloaded enough containers to load the containers available at the next terminal. This means that a barge cannot load all containers and has to leave the remaining containers behind for a successive barge. This results in different call sizes than initially announced at a terminal.

- *Stowage plan of the barge.* The way containers are stacked on the barge, limits the sequence in which terminals can be visited.

- *Dominoes effect of disturbances.* Disturbances at one terminal easily propagate to other terminals and make the planned starting time of a barge at a terminal even more uncertain.

- *Disturbances.* Disturbances, such as crane breakdowns, lacking shipping documents, and containers not yet released by the customs, introduce another source of uncertainty.

Barge operators deal in several ways with the uncertainties in the barge handling process. We mention the following (derived from Van Groningen (2006) and interviews we did ourselves):

- *Increase sailing speed.* Barges can increase their sailing speed to recover lost time. However, increasing sailing speed hardly makes a difference on short trips (from Rotterdam to Antwerp for instance). Moreover, an increase of the sailing speed with about 4km can result in an increase of fuel consumption with more than 200% (Van Groningen, 2006). Increasing sailing speed is therefore not always beneficial.

- *Flexible stowage.* Barge operators (and shippers) try to stow their barge as flexibly as possible, i.e., they stack containers with the same terminal destination together and place them such that they get no problems with nautical aspects of the barge. Flexible stowage allows the barge operator to visit terminals in different sequences.

- *Shifters.* Barge operators can ask a terminal to move some containers on board of the barge, if the containers destined for this terminal are stacked below containers with another terminal destination. The associated moves are called *shifters.* Terminals charge barge operators for this service. The cost of a shifter is significant and barge operators try to avoid the use of it.

- *Extra slack in the sailing schedule.* When barge operators make their sailing schedules they include already a certain amount of slack to be able to recover from delays in the port and to get back on schedule again.

Clearly, the mentioned ways to deal with uncertainties in the barge handling process have a price for barge operators. First, the price of shifters. Second, the flexible stowage results in a lower utilization degree of the barge, which means that a barge operator has to employ more barges to transport the same number of containers. Third, the fact that barges have long waiting times in the port results in fewer round-trips to the hinterland which results in a loss of revenues for the barge operators.

To operationalize the main objective of the barge operator, Van Groningen (2006) defines four key performance indicators. All these indicators are measured over several rotations of the same barge. The indicators are:

1. The fraction of rotations for which a barge leaves the port in time, i.e., in accordance with its sailing schedule.

2. The average delay of a barge on arrival in the port, compared to the time set in the sailing schedule.

3. The maximum delay of a barge on arrival in the port, compared to the time set in the sailing schedule.

4. The operational costs of being delayed.

Note that some key performance indicators (mentioned by Van Groningen, 2006) need to be maximized and others to be minimized. In our opinion, especially the second and the third key performance indicator are not appropriate indicators to measure the extent to which a barge operator realizes his main objective (as formulated in the beginning in this section), namely to minimize possible delays in the sailing schedule. The second and third indicator are related to the external system (outside the port) and not to the operations in the port, which is the focus of our study.

Terminal operator

The main objective of a terminal operator is, according to Van Groningen (2006), to maximize the utilization of his resources, i.e., the berth, crane and crew productivity. Under-utilization of these resources can be caused by several factors (Van Groningen, 2006):

- *Load and unload list announced late.* The list with containers to be loaded and unloaded is sometimes provided late by barge operators. This means that containers are not always available at the time of call or the barge has to wait until the container has been stacked at the quay-side.
- *Deviating call sizes during operation.* Announced call sizes of barges regularly differ from the call sizes at the time the barge handling is started. This can have different causes, but requires flexibility of the terminal operator.
- *Disturbances.* Disturbances, such as crane breakdowns, lacking shipping documents, delayed arrivals of sea vessels, and containers not yet released by the customs introduce another source of uncertainty.

Terminals have different options to increase the utilization of the resources (Van Groningen, 2006):

- *Create a queue of barges.* Some terminal operators prefer to have a queue of barges in front of their terminal to prevent idle time of the quays and to maximize the terminal utilization.
- *Moor barges in parallel.* Barges (especially barges with small call sizes) are sometimes allowed to moor along side another barge yet moored. This is only possible if the crane can reach over the second barge. The benefit is that during mooring of the second barge the crane can continue to work on the first barge. Moreover, barges with small call sizes favor a better service of the terminal.
- *Increase the speed of the cranes.* Terminal operators can sometimes increase the speed of the cranes, thus increasing the utilization of the associated crew and berth.
- *Employ additional workforce.* Most terminals have their own crew and also have the possibility to hire additional workforce.

Van Groningen (2006) reports that if a terminal is flexible towards barge operators, in making or changing appointments for instance, it faces the risk that barge operators use this flexibility to capture uncertainties caused by less flexible terminals. Small terminals in the city feel that they are usually more flexible than the highly occupied terminals at the Maasvlakte.

Under-utilization of the terminal leads to opportunity losses for terminal operators. Direct costs are the hiring of additional labor. Terminals usually charge barges for shifters, since these shifters do not contribute to the number of productive moves of the crane.

To operationalize the objective of the terminal, Van Groningen (2006) defines the following four key performance indicators:

1. The amount of profit a terminal has made in a certain period.
2. The number of realized moves (excluding shifters).
3. The average time a barge has to wait at the terminal before being processed.
4. The maximum waiting time of barges at the terminal.

Note that some of the terminal key performance indicators (mentioned by Van Groningen, 2006) need to be maximized and others to be minimized. Surprisingly, the utilization degree of a terminal is not defined as key performance indicator, which seems to be a logical indicator for the main objective of the terminal operator, namely to maximize the utilization of his resources

Port of Rotterdam

The Port of Rotterdam has on a strategic level three main concerns, namely competition with other ports, environmental issues, and congestion of the infrastructure (Van Groningen, 2006). The Port of Rotterdam has several goals which are related to the three concerns mentioned. Van Groningen (2006) defines the main objective (related to the barge handling problem) as minimizing the congestion in the port. The key performance indicators are: the average level of congestion per hour and the maximum level of congestion per hour.

To maintain the competitive position, it is important that the Port of Rotterdam is attractive for liner shipping companies (carriers) to do their business. This comprises the organization of storage, the organization of hinterland transport, port safety, and costs. For the Port of Rotterdam it is therefore important that hinterland connections are such that containers can move quickly, at low costs, and reliably to their destination. Transportation via water is therefore seen as a promising alternative compared to road and rail transport, since the latter two have not enough capacity and get congested. Moreover, transportation via water is less harmful for the environment.

2.7 The key performance indicators used in our model

In our study we focus on the average performance of barges and terminals, i.e., we do not consider the performance of a single barge or terminal. In our study 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 can not be influenced by the terminal operator (we elaborate on this in Section 4.5.2). 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 scheduling literature and are related to the off-line benchmark (see Section 4.4). 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 barges leaving the port late.* Leaving the port late means that a barge leaves the port later than the time set in its sailing schedule. This indicator measures the fraction of barges that did not manage to leave the port in time.

2. *Average project tardiness.* The project tardiness of a barge is equal to the maximum of zero and the *project lateness* of the barge. Project lateness is: the actual time the barge leaves the port minus the time set in the sailing schedule. Project lateness can be negative.

3. *Average activity tardiness.* If an activity (the loading or discharging 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: the actual time an activity is completed minus the due date of that activity. Activity lateness can be negative.

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

In future research the performance indicators might become more configurable for different barge operators, taking into account the capacity of barge, the stowage restrictions, the unload/load balance at a terminal, etcetera.

2.8 Project details

Our research is performed within the BSIK program Transumo, which is a platform of companies, governments, and knowledge institutes who together develop knowledge to realize sustainable mobility. Transumo is an abbreviation of TRANsition to SUstainable MObility. Within Transumo, we performed our project within the theme Chain Integration, sub project Diploma, sub project Approach 2. Diploma stands for DIstributed PLanning Of freight transport networks using Multi-Agent technology. This project aims for development of Multi-Agent systems for real-time transportation planning with multiple actors. Approach 2 is named after a preceding project called Approach 1 (see Section 2.6.1) and has the aim to improve the handling of barges in the Port of Rotterdam by means of Multi-Agent technology.

2.8.1 Objectives

The mission of Transumo is *to accelerate/encourage the transition to sustainable mobility. This will be achieved by initiating, and establishing for the long term, a transition process that leads to the replacement of the current, supply driven, mono-disciplinary technology and knowledge infrastructure, with a demand driven, multidisciplinary and trans-disciplinary, participative knowledge infrastructure.* The transition to the new knowledge infrastructure leads to advances that help to strengthen the competitiveness of the Dutch transport sector (profit) and to preserve and improve spatial and ecological (planet), and social (people) aspects of mobility. The three goals profit, planet, people specify the concept sustainability. Additionally, Transumo aims for the transition of knowledge to practice and enhancing the cooperation between governments, companies and knowledge institutes.

Our project contributes to sustainability by increasing the utilization of resources (profit and planet), reducing unnecessary movements (profit, people, planet), reducing the necessary transport fleet (planet, profit), increasing the flexibility and reliability of transport (people, profit), and stimulating the cooperation between companies, universities and government (people).

During our research we have organized and participated in several symposia and workshops to present and explain our research and to start the discussion with practitioners. Also several interviews and articles have appeared in, e.g., CTIT year report 2005, Computable (February 2006), Supply Chain Magazine (March 2007), Tubantia (November 2007), and the Logistiek Krant (July 2008). We also presented results at international conferences: Odysseus (2006), Smart Business Network Initiative (2006), R3 seminar (2007), Tristan VI (2007), HICL (2008), and HMS (2008).

2.8.2 Parties involved

In our research the following parties participated: the University of Twente, the Delft University of Technology, the Erasmus University, Initi8 B.V., the Port of Rotterdam, and the Ministry of Transport, Public Works and Water Management. The tasks in the project have been divided over the participants. Our task was to develop (on an aggregated level) a Multi-Agent system for the barge handling problem, to develop algorithms in order to optimize the performance of the system, and to test the performance by means of simulation and comparison with an off-line benchmark (centralized optimization). Initially, the Delft University of Technology would perform a detailed simulation (including the current of the river, detailed maps of the port etcetera) to evaluate the performance of our Multi-Agent system in the Port of Rotterdam. The idea was to link the (detailed) simulation model to be developed by the Delft University of Technology with the (abstract) simulation model developed by the University of Twente. Unfortunately, the Delft University of Technology withdrew during the project. We continued with our study and have designed our simulation model such that in the future a link with the simulation model of the Delft University of Technology can still be established.

2.9 Summary

In this chapter we described the background of the barge handling problem. We focused on aspects and developments that affect or will affect our problem. We discussed the history and prospects of global container transportation. We described the intercontinental transport of containers by focusing successively on the role of liner shipping companies, the Port of Rotterdam, and the barge sector. We placed these roles in a historical context and described developments that are observed and expected in the container sector. Then, we described in detail an earlier study to our problem and we summarized the results of another study regarding the (key) performance indicators of barge operators, terminal operators, and the Port of Rotterdam. We concluded the chapter with a description of the performance indicators we use in our model and described the way our project is organized.

The growth of intercontinental container transportation affects all actors in the container transportation chain. Liner shipping companies are continuously developing and reorganizing their business. The Port of Rotterdam faces challenges to deal with the increasing container volumes in terms of transshipment capacity in the port and the capacity of hinterland transport modes. The barge container sector in turn is rapidly developing and becoming more attractive due to the increasing oil prices and the attention for sustainable transportation. These developments result in several problems, such as increasing waiting times of barges at terminals, and show the need for structural improvements in the container barge sector.

Chapter 3

Decentralized planning: analysis and design choices

3.1 Introduction

In this chapter we discuss the research question: *What is an efficient and effective Multi-Agent system for the barge handling problem?* To design an effective Multi-Agent system, both optimization and acceptance are important. Traditionally, the focus in the realm of Operations Research has been more on the realization of optimal solutions (efficiency). However, if the system is not acceptable for the players involved, a solution will never be implemented in practice and can certainly not be effective.

Optimization in a Multi-Agent system is not trivial, since each agent is only concerned with its own goals, and there is no global notion of utility (Zlotkin and Rosenschein, 1996). Global efficiency is the result of the actions of autonomous, rational agents and cannot be obtained directly. Acceptance of the users has to do with the extent to which participants feel that they are better off by participating in the system and that they can agree with the conditions to participate. Our research question in this chapter can therefore be formulated more specifically:

1. How to facilitate optimization by means of multiple interacting agents?
2. How to realize acceptance by the users of the system?

In this chapter we discuss several design issues that affect the efficiency and acceptance of our solution. In Section 3.2 we discuss the notion of decentralized control, agents and Multi-Agent systems. Moreover, we make some first choices regarding our Multi-Agent system. In Section 3.3 we discuss desired properties of a Multi-Agent system and an interaction protocol, and we discuss some specific difficulties we face in our problem. Section 3.4 analyzes the interaction protocol proposed in a previous study, called Approach 1 (see Section 2.6.1). Finally, in Section 3.5 we propose our interaction protocol and we reflect on its strengths and weaknesses.

3.2 Decentralized control

In this section we describe the concept of decentralized control and Multi-Agent systems, and we make some first choices regarding the design of our Multi-Agent system.

3.2.1 What is decentralized control?

With decentralized control we mean the coordination of activities among multiple self-interested players that make decisions independently and where the system performance is a result of the actions and decisions of individuals. Centralized control, on the other hand, is the coordination of activities by a single party that decides on and coordinates the actions of individuals, to maximize the system performance. Decentralized control is in the literature also referred to as *distributed rational decision making* (Sandholm, 1999) or just *Multi-Agent systems* (Jennings et al., 1998; Sandholm, 1999; Wooldridge, 2005).

Two important elements in the design of a Multi-Agent system are generally: i) the *interaction protocol* and ii) the *strategy of players* (Zlotkin and Rosenschein, 1996; Sandholm, 1999; Wooldridge, 2005). We define an interaction protocol as a protocol that prescribes the way agents communicate, the content of the communication, and the result of the communication. The strategy of a player is the way a player maps (historical) information of the system to actions. The interaction protocol and the strategy of players are strongly related. For example, if we assume that players are self-interested, we can expect them to choose a strategy that is in their best interest. However, not all strategies are desirable from a system (designer's) perspective, e.g., providing incomplete or incorrect data may not be beneficial. To design a robust non-manipulable Multi-Agent system, Sandholm (1999) claims that we need to adopt a *noncooperative, strategic* perspective, i.e., we have to think about the social outcomes that follow from a protocol which guarantees that opportunistic players will choose a desired strategy. In fact, the designer can set the 'rules of the game' by introducing a certain interaction protocol in such a way that players, although they are self-interested, will display desired behavior. The design of an interaction protocol is therefore important since it determines the acceptability and sustainability of the system. Interaction protocol design is also known in game theoretic literature as *mechanism design* (Zlotkin and Rosenschein, 1996).

3.2.2 What are agents and Multi-Agent systems?

Agent technology first emerged in the mid to late 1980's in the field of Artificial Intelligence. Since then, a lot of interest has been given to the agent concept by a variety of disciplines in computer science. According to Wooldridge and Jennings (1995), an agent is usually defined as a hardware or (more often) software based object with key properties *autonomy, social ability, reactivity,* and *pro-activeness*. This means that an agent can operate without direct intervention of humans or others (autonomy), can interact with other agents (social ability), can perceive its environment and respond to changes in it (reactiveness), and is able to exhibit goal-directed behavior by taking the initiative (pro-activeness). The purpose of an agent is to take over tasks from its principal, i.e., the human or organization it represents. To do so, agents can collect, store, and analyze data during operation, and make decisions and agreements on behalf of their principal.

A system with more than one agent is called a Multi-Agent system. A Multi-Agent system can be defined as a loosely coupled network of problem solvers that work together to solve problems that are beyond the individual capabilities or knowledge of each problem solver (Durfee and Lesser, 1989). Agents in a Multi-Agent system can be self-interested. The basic idea behind the application of a Multi-Agent system is that through the interaction of local agents coherent global behavior is achieved. Characteristics of a Multi-Agent system are that i) each agent has incomplete information, i.e., a limited viewpoint, ii) there is no global system control, iii) data is decentralized, and iv) computation is asynchronous (Jennings et al., 1998).

Multi-Agent systems are not applicable to all kinds of problems. Instead they are very suitable for problems that have the following properties (Van Dyke Parunak, 1999):

- *Modular.* The problem can be partitioned in entities that have a well-defined set of state variables that are distinct from those of the environment. A subset of the entity's state variables is coupled with the environment's state variables to provide input and output.
- *Decentralized.* The problem can be decomposed into stand-alone processes, each capable of doing things autonomously, without direction by other processes.
- *Changeable.* The structure of the problem may change quickly and frequently.
- *Ill-structured.* Not all the necessary structural information of the problem is available when a solution is designed.
- *Complex.* The problem has a significant combinatorial complexity.

These problems usually require a more robust and adaptable solution, which can be realized using Multi-Agent systems. Research into Multi-Agent systems in the field of logistics has increased over the last years. Several researchers have attempted to apply agent technology to manufacturing systems (see, e.g., Arbib and Rossi, 2000; Dewan and Joshi, 2002), supply chain management (see, e.g., Ertogral and Wu, 2000; Qinghe et al., 2001), and transportation networks (see, e.g., Bürckert et al., 2000; Figliozzi et al., 2006; 't Hoen and La Poutré, 2006; Mes, Van der Heijden and Schuur, 2008; Mes, 2008).

3.2.3 The design of our Multi-Agent system

To define the agents we apply a *physical decomposition* of the problem, which means that agents represent entities in the physical world (Shen and Norrie, 1999). To simplify our design, we model only two types of agents: barge operator agents and terminal operator agents. There are different methodologies to design a Multi-Agent system (see for an introduction, e.g., Wooldridge, 2005). These methodologies are particularly useful when one has a large number of different agents and where there is a lot of freedom in the assignment of roles and responsibilities to agents. In our problem the definition of the agents and their roles and responsibilities is more or less imposed by the business problem.

With respect to the strategy of players, we assume that all agents are *opportunistic* and make decisions in the best interest of their principal. With opportunistic we mean that every agent exploits opportunities for its principal's benefit, regardless of the other agents. We assume that agents from the same type (like barge operator agents) have identical intelligence.

The Multi-Agent system, consisting of all the barge and terminal operator agents, has the same characteristics as the current situation in the Port of Rotterdam; it is open, ill-structured, and loosely coupled. We assume that communication in our Multi-Agent system only takes place between barge and terminal operators, not between agents of the same type. All agents use the same interaction protocol and can make decisions on behalf of their principal. In the Multi-Agent system we imitate more or less the way the market is currently structured to make the solution acceptable for the players involved. However, a Multi-Agent system can, compared to the current situation, contribute in two respects:

- *Communication.* The communication between players can be done faster, more efficiently, and more reliably with the use of agents. Moreover, the automation of the communication relieves the task of the barge and terminal operators.
- *Decision making.* Agents can partly overcome the bounded rationality of the barge and terminal operators, since they can use, e.g., techniques

from the field of Operations Research to search quickly through a large number of solutions.

In our model we rationalize the decision making of barge and terminal operator agents. This means that we exclude all kinds of personal influences in the decision making, which may be considered as important. In future research these issues may be included in the decision making of the agents.

In the next sections we concentrate on the design of the interaction protocol for our Multi-Agent system.

3.3 Interaction protocol design

In this section we discuss the design of the interaction protocol. The goal is to design an interaction protocol for self-interested agents such that, if agents follow this protocol, the overall system behavior will be acceptable (Jennings et al., 1998). In Section 3.3.1 we discuss desired properties of a Multi-Agent system and an interaction protocol. To realize these desired properties we face several difficulties. We discuss these in Section 3.3.2. We attempt to make clear that these difficulties need to be considered carefully to establish an acceptable and sustainable Multi-Agent system.

3.3.1 Desired properties of a Multi-Agent system and the interaction protocol

To design a Multi-Agent system that is acceptable for players and facilitates optimization of their respective objectives, we made a list of desired properties of the Multi-Agent system and the interaction protocol. This list of properties is based on Zlotkin and Rosenschein (1996), Sandholm (1999), and Wooldridge (2005). We only mention the properties that we consider as relevant in our problem:

- *Individual rationality*: It must be individually rational to participate, i.e., participating in the Multi-Agent system must lead to a higher pay-off than not participating.
- *Stability*: The protocol must be stable (non-manipulable), i.e., it should motivate each agent to behave in a desired manner.
- *Guaranteed success*: The protocol has to have a guaranteed outcome, i.e., finally an agreement has to be reached with certainty.
- *Quick success*: An agreement has to be reached in ‘reasonable’ time.

- *Distribution and communication efficiency*: Distributed protocols are preferred to prevent a performance bottleneck or the collapse of the Multi-Agent system due to failure of a single node. Moreover, minimizing the required communication to realize a solution is an attractive property.
- *Computational efficiency*: To improve the speed of the coordination it is desirable to design an interaction protocol that needs as little computational effort as possible, especially in a real-time fashion where decisions have to be made quickly.
- *Simplicity / understandability*: It is desirable to make the strategy of an agent 'obvious'. The principal of the agent can then easily (tractably) understand the agent's strategy.
- *Implementable / sustainable*: The Multi-Agent system should give a sustainable solution to the problem. This concerns both the logistical performance (no obstruction) and the organizational or administrative procedures. For example, the sharing of benefits and losses should be designed fairly. If players perceive the division of gains and losses as unfair, although the system performs well from a logistical perspective, players still might decide to withdraw if they feel that their competitive position is, or could be, threatened.

The design of the interaction protocol plays a key role in the realization of these properties (Zlotkin and Rosenschein, 1996). The design choices with respect to the interaction protocol therefore need careful consideration, since they impact the ease of keeping the system up and running, i.e., to make it attractive for participants to continue to be part of the system. In the next section we elaborate on several complicated design choices we faced.

3.3.2 Specific difficulties in the realization of the desired properties

To realize the desired properties listed in Section 3.3.1, we are confronted with specific design choices. These choices in the protocol design affect the ease of keeping the system up and running. Our experience is that the complexity of the choices is not always recognized at first sight. We therefore start with a small example to illustrate a possible conflict in the realization of two properties and the way this conflict can be resolved. In the remainder of this section we mention more general design issues and explain the impact on the functioning of our system.

Example to illustrate a difficulty: how to realize a robust system?

To illustrate the importance of the interaction protocol we give an example of a desired property of the system, namely sustainability, which seems to be

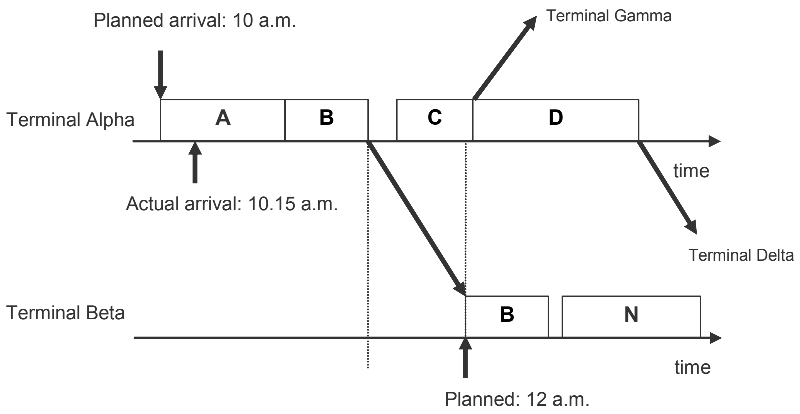


Figure 3.1: Example of the inter-dependency of different terminals through the actions of barge operators

in conflict with another property, namely individual rationality. Consider the following situation. We have terminal Alpha which has planned the barges A, B, C, and D successively in time (see Figure 3.1). After terminal Alpha, the barges B, C, and D visit the terminals Beta, Gamma, and Delta, respectively. If we look at barge B, we assume that the planned arrival time at terminal Beta minus the planned departure time at terminal Alpha is exactly the sailing time from terminal Alpha to Beta. This means that a delayed departure at terminal Alpha immediately causes a delay at terminal Beta. This is what happens in the example. Suppose that barge A arrives 15 minutes late and is started processing at 10.15 a.m. This delayed processing immediately causes a delay at the terminal Beta, Gamma and Delta, since the barge B, C, and D will be affected.

It will be clear that if barges plan terminal visits right after each other (no slack between terminal visits), one disruption can propagate quickly through the whole system and affects the operations of many barge and terminal operators. This makes the system vulnerable for even small disruptions. To reduce the dominoes effect of disruptions, it would be better for the system as a whole (that is for all barge and terminal operators) if barges would plan slack between terminal visits, so that disruptions can fade out and affect only a few local activities. However, why would a barge add slack in between terminal visits if its main concern is to leave the port as soon as possible or at least within its time window. Of course, if all barges plan terminal visits tight to each other, a single barge will surely be worse off. However, there is an individual incentive to deviate in the short term, i.e., to behave opportunistically. Here we have the conflict between sustainability and individual rationality.

In Section 3.5.1 we explain our solution to this example. In the following sections we first discuss some other design choices.

Regulation of the behavior of agents

To get a stable system, it is necessary that agents are not able to manipulate each other, e.g., by providing incorrect or incomplete information. Manipulation or strategic behavior happens frequently nowadays as we described in Section 1.1.1, leading to a lot of uncertainty for terminals and barges. To make manipulation or strategic behavior unattractive, we need to regulate the behavior of the agents. We have two options to do so, namely, external regulation and self-regulation. External regulation is realized by an external party (a kind of police officer) that registers the behavior of individual players and can punish the players if they display undesired behavior. Self-regulation means that players in the system correct each other without the help of an external party.

The question is whether external regulation leads to a sustainable solution, since an external regulator implies that there is an individual benefit to deviate, possibly at the cost of other players. The incentive to deviate will depend on the probability that one will be punished and the punishment itself. Moreover, the establishment of a police officer might conflict with the required autonomy of actors and the reluctance to share information. We therefore prefer the use of self-regulation.

Benefit/loss sharing

The sharing of benefits and losses is a complex issue and strongly related to the field of game theory. If it is possible to avoid the benefit and loss sharing issue in the design of a Multi-Agent system, it will have many benefits in the implementation and operation of the system. First, a certain division of gains and losses always results in discussions, especially when many players are involved and everyone has a clear interest in the result. Second, it will be hard to make a division that is acceptable by everyone and can be explained as the fairest division. What is defined as fair in game theory is not always perceived as fair (cf. Schotanus, 2007). Third, the registration of activities, tracing the causes of delays, and quantifying the impact will take much time and effort, and results in direct costs for operating the system.

Consider our problem. Suppose we can quantify the benefits and losses of individual players, then we have to find a mechanism that distributes these gains and losses fairly and acceptably among all the players. One can imagine that this is not easy. For instance, how to trace a delay to a specific barge or terminal if the dominoes effects are significant. Note that delays can cumulate which makes it less transparent what the actual cause of a delay is and what the contribution of this cause is to the delay. Another example is the contribution of an individual player to the performance of the others. Can the coalition of players decide to refuse a terminal or barge in the port, since this barge or terminal has a negative effect on the operations of the rest?

Negotiation versus contract net protocols

The coordination of activities among self-interested agents can be done either via direct communication (negotiation), or indirectly by means of a contract net protocol (Wooldridge, 2005). Direct communication means that agents contact each other directly to negotiate. A contract net protocol is according to (Smith, 1980; Wooldridge, 2005) the allocation of tasks by means of an auction. This means that a kind of (virtual) currency has to be introduced which is exchanged between parties.

The latter option is strange in the current situation in the Port of Rotterdam, since there are no contractual relationships between terminals and barges. Moreover, a virtual currency also introduces the problem how gains and losses have to be interpreted, how virtual money can be obtained, and how it can be exchanged in, e.g., real money. Direct communication, on the other hand, is the way the manual system works at the moment. It is more likely that this is an acceptable way of communication for barge and terminal operators.

Extent of autonomy of agents

Another design issue is the extent of autonomy of agents. If agents only preprocess tasks, i.e., they collect information and propose a decision which has to be confirmed by someone in the real world (the principal of the agent), one can imagine that delays in the decision process are likely to happen. A fully autonomous agent, however, makes decisions for its owner, where the owner can only afterwards comment on the decision of the agent.

In our problem a non-autonomous agent might have the following consequences. Suppose a barge operator agent has contacted all terminal operator agents about convenient handling times and proposes a rotation to its principal, i.e., its barge operator. The barge operator agrees with the proposed rotation about fifteen minutes after his agent did the proposal. Depending on the way terminals deal with requests for quay capacity, it can happen that a time slot is already given to another barge. This means that the barge operator agent has to make a new proposal, which has to be confirmed again by his barge operator. This process is time consuming and can result in several iterations before a feasible rotation is found. This can be annoying for the barge operator. A fully autonomous agent might then be more desirable. Disadvantage is that a barge operator is possibly not always satisfied with the decision of his agent and wants to overrule its decision.

Trust and the intelligence of the agent

When an agent makes decisions for its principal, then the principal needs to trust its agent. Somehow, the principal has to be convinced that the agent decides in its best interest and that it could not have done better. To gain this trust it may be important that the agent can explain why it made its decision and why this decision was the best for the principal (Wooldridge, 2005). If this is important, and we think it is in our problem, this pleads for the application of techniques that lead to results that can be explained. To give an illustration, the application of a reinforcement learning method may lead to good results, but it is much harder to explain why a certain decision was made.

Real-time decisions of the agents

In a real-time Multi-Agent system it is important that decisions can be made quickly, but how quick? If a barge operator agent in our problem needs to determine an optimal rotation, it has to consider a large number of options, namely $n!$, with n the number of terminals it has to visit. This can be computationally demanding and take some time before the agent makes a decision. In the meanwhile the state of the system could have changed, e.g., certain time slots at terminals might not be valid anymore because another operator agent applied for it. The faster a barge operator agent can make a decision, the more likely it is that the state of the system has not changed in the meanwhile. The longer it takes, the higher the probability that the decision of the operator agent appears to be infeasible and a new decision process has to be started. This affects the communication efficiency of the system.

Degrees of freedom for a principal to design his agent

The last difficulty we mention may become important on (or some time after) implementation of the Multi-Agent system and has to be taken into account during the design of the system. Suppose the Multi-Agent system is implemented in practice and after some time players indicate that they want to have more control over their agent. This raises the question (regarding the design of the Multi-Agent system) how many degrees of freedom do we give to players to design their own agent. Do we allow a barge or terminal operator only to configure their agent or to even design the full functionality of their agent themselves? The answer to this question may affect the functioning of the Multi-Agent system.

To give an illustration, suppose we allow terminal operators to design their own agent, then it is necessary that this agent acts according to the 'rules of the game'. If the terminal operator agent makes decisions which lead, e.g., to infeasible appointments with barges, then it must be possible to force the terminal operator to adjust his agent. If a terminal is not willing to do so and cannot be forced to do so either, then the Multi-Agent platform operators is left no possibility than to observe the diminishing quality of the Multi-Agent system.

3.4 Analysis of the interaction protocol proposed in Approach 1

Before we propose our interaction protocol, we analyze the interaction protocol used in the Approach 1 project to make clear that this protocol has room for improvement. This regards both the efficiency and the acceptability of the protocol. Let us give a brief summary of the protocol (for a detailed description we refer to Section 2.6.1). In Approach 1, every day at a fixed time (e.g., 7 a.m.) the rotations of all barge operators are planned for the next 24h. The barge operators therefore prepare several scenarios which vary in the sequence in which terminals are visited or the time at which terminals are visited. At the fixed time all barge operators start to plan their rotation. The strategy of every barge operator is to settle a rotation by considering the successive scenarios. The barge operator starts off communicating the arrival times at terminals determined in the first scenario. If all terminals agree with the proposed arrival time, the rotation is settled. If not, the barge operator communicates the arrival times determined in the second scenario to the respective terminals, etcetera, until a scenario is confirmed by all terminals.

In our opinion this protocol and the resulting Multi-Agent system lack some of the desired properties mentioned in Section 3.3.1. Let us clarify this for each of the properties that can be improved.

- *Individual rationality*: It is likely that participating in the Multi-Agent system is not individually rational. The reason why is that barge operator agents concurrently try to settle rotations. During this process, terminal operator agent Alpha might reply to barge operator agent A, that a time slot is already occupied, e.g., by barge operator agent B. However, barge operator agent B might discover that another terminal did not agree with a proposed arrival time and therefore cancels the appointment already made with terminal Alpha in order to go for another scenario. This means that the time slot has become available for barge operator agent A, but this barge operator agent already started considering another scenario. Finally, barge operator agent A might end up with a rotation that could have been planned more efficiently. When barge operator A (the principal of the agent) discovers that he could have done better than his agent, he might start to distrust his agent and is maybe less willing to participate in the system.
- *Stability*: It is not clear how the designers of the Approach 1 Multi-Agent system deal with strategic behavior. Based on the description there

is no self-regulation or external regulation implemented, which means that lying about processing times can still happen.

- *Guaranteed success*: It is not clear how a successful result for all barge operators can be guaranteed. Every barge operator generates a large number of scenarios, but that is not a guarantee for success, since there may be simply too much work to be planned in the planning horizon. Specific requirements with respect to the generation of the scenarios are not mentioned. For instance, what if the number of scenarios generated by a barge operator agent turns out not be enough to settle a rotation? Moreover, the limited information barge operator agents have at their disposal does not yield many options for determining smart scenarios.
- *Quick success*: Especially when the utilization degree of the network increases, it will be hard to find a feasible rotation (if that is even possible). Barge operator agents might need to communicate with the terminal operator agents for a considerable time to settle a rotation.
- *Distribution and communication efficiency*: The system is distribution efficient, since all agents can run on different platforms. However, it is not communication efficient, since a lot of communication is necessary to come to an agreement.
- *Simplicity / understandability*: Although the way of communication can be explained easily, it will be difficult for an agent to explain why it comes up with a certain rotation. It could hardly explain that the lesser quality of its rotation was caused by the actions of other agents and the temporary reservations they made at the terminals.
- *Implementable / sustainable*: The aim of APPROACH 1 was to support the construction of rotations for the barges concerned. However, the way barges can plan their rotations may lead to a less robust system, as barges can plan terminal visits tight to each other. Moreover, it is likely that barges that have to call at a lot of terminals, have some trouble to find a feasible rotation compared to barges that have to call at only a few terminals. If players feel that they have a natural disadvantage over other barges, just because they have to visit a large number of terminals, they might be less willing to accept the Multi-Agent system.

It is clear that the interaction protocol proposed in APPROACH 1 has some drawbacks, besides that the system is not able to respond to updated information. In the next section we propose another interaction protocol, applicable in a dynamic setting (where information is revealed over time) and we show the extent to which desired properties are realized.

3.5 Proposed interaction protocol

In this section we propose our interaction protocol and we reflect on the extent to which the desired properties of an interaction protocol are realized.

3.5.1 The interaction protocol

The starting point for our interaction protocol design is that we aim to create a real-time planning system. This means that the protocol must be suitable for a dynamic implementation, i.e., where barges arrive over time and make decisions successively. Moreover, it has to meet the desired properties discussed in Section 3.3.1, to deal with the difficulties as mentioned in Section 3.3.2, and to meet the specific business constraints introduced in Section 1.1.3, such as the limited information exchange.

An important assumption we make is that barge and terminal operators want to make reliable appointments for the transshipment of containers. Appointments have several advantages for barge and terminal operators. They give barge operators certainty in their operations. This gives them, e.g., the possibility to optimize their container services and to issue reliable transit times to customers. Terminal operators, on the other hand, can use appointments to influence the handling times of barges and to optimize the operations that are related to the container transshipment, such as the yard planning. To see what the effects are of making appointments, we evaluate in Chapter 7 a totally different situation in which no appointments are made.

The interaction protocol we propose consists of three parts, namely, the way of communication, the information that is exchanged, and the result of the communication. In the next two sections we describe these parts successively, starting with the result of the communication.

The result of the communication

We assume that the result (or the aim) of the communication between barge and terminal operator agents, is to make an agreement about convenient handling times. We call this agreement an *appointment.* We propose the following definition of an appointment. An appointment made by a barge and a terminal is an agreement from two sides. The barge promises the terminal to be present at the terminal at a certain time, i.e., the latest arrival time. The terminal in turn *guarantees* the barge a latest starting (or departure) time, if the barge keeps its promise. If the barge does not keep its promise and arrives later than the announced time, it has to make a new appointment. By making appointments, the barge uses the guaranteed latest starting (or departure) times at preceding terminals to calculate the arrival time at the next terminal.

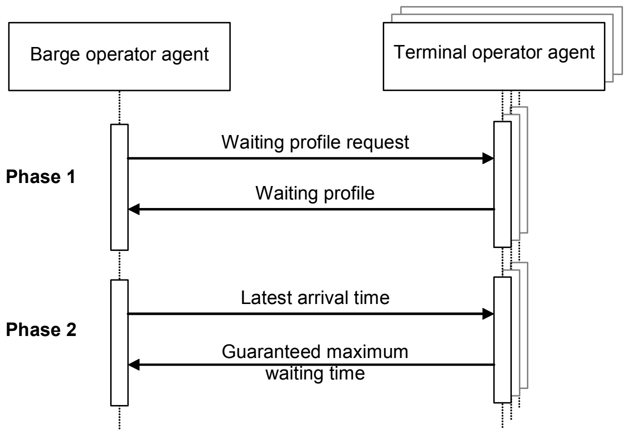


Figure 3.2: A sequence diagram of the communication between a single barge operator agent and multiple terminal operators in our model

An appointment, as result of the communication, is important in our protocol. We elaborate on this in Section 3.5.2. We first discuss how barge and terminal operator agents can make these appointments.

The way of communication and the information exchanged

In our model we choose for direct communication between the barge and terminal operator agents. We assume that the barge operator agent initiates the communication with the terminals. In Figure 3.2 we illustrate the way barge and terminal operator agents communicate in our protocol.

The communication between the barge operator and terminal operators can be divided in two, so-called, phases. These phases correspond with two phases in the decision process of the barge operator agent. In the first phase of the communication, the barge operator agent tries to obtain information about the occupation of the terminals. Based on this information it determines a convenient rotation. In the second phase the barge operator agent makes appointments with the terminal operator agents, based on the rotation determined in the first phase. Let us consider this in detail and also focus on the information that is exchanged.

For the first phase of the communication we introduce the concept *waiting profile*. A waiting profile, issued by a terminal for a particular barge, denotes the *maximum waiting time* for a barge until the start of processing, for every possible arrival time during a certain time period. The maximum waiting times determining the waiting profile are *guaranteed* maximum waiting times. We can define a *waiting profile* of terminal j, more formally, as a $t-$parameter family of pairs (t, w_{jt}), where w_{jt} is the maximum waiting time when the barge arrives at time t at terminal j, for all t during a certain time period $[0, T]$. A waiting profile is generated by a terminal after the request of a barge and is barge specific. We explain the concept of waiting profiles in detail in Section 5.4.3.

At the start of the communication, the barge operator agent sends a request to all terminal operators to issue a waiting profile. In this request the agent announces the total number of containers to load and unload, i.e., the call size. After receiving the waiting profiles, the barge operator agent can determine a rotation that minimizes its sojourn time in the port. It therefore assumes that it has to wait the guaranteed maximum waiting time at a terminal. When the barge operator agent has determined the best rotation, i.e., the rotation minimizing its port sojourn time, it proceeds to phase 2.

In phase 2 of the communication, the barge operator agent announces its latest arrival time to all the terminal operator agents. The terminal operator agents reply that, if the barge arrives not later than its latest arrival time, it has a certain guaranteed maximum waiting time. We assume that the whole communication (phase 1 and 2) takes place in real-time. The reason why we focus on real-time decision making is that the faster a barge operator agent decides on its best rotation, i.e., the faster phase 2 follows on phase 1, the more likely it is that the state of the system has not changed significantly and that waiting profiles issued by terminal operator agents in the first phase are still valid.

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

3.5.2 Analysis of the interaction protocol

In this section we assess the interaction protocol we propose. We successively analyze the extent to which our protocol has the desired properties of an interaction protocol (as introduced in Section 3.3.1) and mention some additional strengths and weaknesses.

Reflection on the protocol properties

To motivate the choice for the protocol we reflect briefly on the desired properties mentioned in Section 3.3.1 and the specific complexities of Section 3.3.2. Recall that we assume opportunistic agents and that agents act in the best interest of their principal. The objectives of the barge and terminal operators are introduced in Section 2.7.

- *Individual rationality*: The fact that agents are opportunistic and make the best decisions -relative to their frame of reference- for the barge and terminal operators, makes the system individually rational as long as the system performance as a whole is acceptable for the players concerned. The latter cannot be guaranteed, but will be evaluated in the successive the Chapters 5 and 6.
- *Stability*: The system can be manipulated if agents can provide incorrect or incomplete information without negative consequences for them. By introducing a kind of self-regulation we expect that cheating becomes less attractive. Self-regulation can be established if terminals keep track of the 'reputation' of a barge. In case a barge arrives late frequently, or provides wrong information, the terminal can penalize this barge by providing time slots that result in a lot of waiting time or by giving the barge low priority. Due to the dominant position terminals have compared to barges, it will be hard to regulate the behavior of terminals from within the system. Our model does not provide a solution to that.
- *Guaranteed success*: A successful outcome of the communication can be guaranteed if the waiting profiles issued by the terminals are sufficiently long, i.e., at least as long as the time a barge needs to complete its rotation. Since terminals do not know how much time a barge needs, we suggest that a terminal operator agent issues a waiting profile over a time period, which ends after the last planned activity.
- *Quick success*: By exchanging waiting profiles, a quick success is likely, since the barge operator agent can choose a rotation that minimizes the sum of the sailing, waiting, and handling times and communicate the resulting arrival times to terminals. There is a combinatorial complexity in finding the rotation with the smallest sojourn time. However, with the use of heuristics this problem can be tackled.
- *Distribution and communication efficiency*: Barge and terminal operator agents communicate directly, there is no bottleneck in the communication. Moreover, the communication is efficient, since it requires only a limited number of message exchanges.
- *Simplicity / understandability*: The rotation proposed by a barge operator agent minimizes the sojourn time in the port. This can be easily explained to the barge operator.

- *Implementable / sustainable*: With respect to implementability and sustainability several issues play a role. The first issue is robustness as introduced in the example of Section 3.3.2. By making the barge responsible for a timely arrival at the terminal, we make it the barge's responsibility to cover uncertainties in the processing and sailing times at terminals. In this way, although the barge operator is opportunistic and aims for minimization of the sojourn time in the port, it is likely that the barge operator agent will include slack in between terminal visits. We thus uncouple the operations of terminals such that the propagation of disruptions is likely to fade out. The second issue is the fact that we use a negotiation protocol so that we can stay away from benefit/loss sharing issues. These two issues make it more likely that the system is implementable and sustainable.

We suggest to give the agent a high level of autonomy to make quick decisions possible. We think that especially barge operators are likely to trust their agent, since the agent aims to make decisions in the best interest of the barge operator and can easily explain why it made its decision.

Strengths and weaknesses

Besides reflecting on the desired protocol properties, we also mention some specific strengths and weaknesses of our interaction protocol regarding the communication, the information that is exchanged, and the result of the communication.

In our opinion, the choice for direct communication is a strength, since direct communication (or negotiation) is suitable for coordination among competitive or self-interested agents (see also, Jennings et al., 1998; Huhns and Stephens, 1999), mirrors daily practice, is probably easier accepted by barge and terminal operators, and enables us to stay away from benefit and loss sharing issues, which would make the implementation of the system unnecessarily complex.

The exchange of waiting profiles is a service of terminals to barges to accurately estimate the arrival time at the next terminal. This allows barges to plan a rotation and to agree on a latest arrival time at a terminal. Terminals, on the other hand, can use waiting profiles to balance their workload during the day. They can, e.g., indicate preferred handling times by varying the waiting times during the day or by increasing the maximum waiting time during periods that are expected to be busy. This makes waiting profiles different from time windows, since the latter only indicate whether the processing of a barge can be started or not. Moreover, a terminal can augment the maximum waiting time with some slack to create flexibility in its quay schedule. In this way, a terminal may be able to increase its utilization, since it has more possibilities to schedule the barges with which appointments are already made.

The fact that terminal and barge operator agents can make reliable appointments contributes to a more reliable and robust system, since barges are more or less forced to include slack in between terminal visits. If a terminals really 'punishes' a barge by giving it less favorable waiting times if it does not stick to the agreement made, then we establish a kind of self-regulation to regulate the behavior of barges. This makes the system more robust than when we apply some kind of external regulation. In the latter case it remains somehow beneficial to show undesired behavior, whereas with self-regulation this is no longer true. On the other hand, a self-regulation mechanism to regulate the behavior of terminals is much more difficult to establish, since terminals have naturally a dominant position in the port. To let terminals display desired behavior it might be necessary to apply other mechanisms or means, e.g., contracts, service level agreements with the Port Authority, or law.

The appointment structure with the corresponding agreements which contribute to a reliable and robust system, is also the weak part in our solution. If terminals do not penalize barges that display undesired behavior, then they create an incentive for barges to continue with this behavior. This can result in a situation that barges reduce the slack in between terminal visits, since it is not very important to arrive at the announced time. This means that the system becomes less reliable and robust, and tends to slide off to the current situation. It is therefore important that all players (and especially terminals) agree with the 'rules of the game' and apply these rules consistently. Several people in practice are skeptical whether this is realistic, i.e., that barges and especially terminals will stick to the 'rules of the game'. We think that if people are not willing to make reliable agreements, planning makes not much sense. We then had better looked for a solution in which no agreements are made. In Chapter 7 we consider such a system.

3.6 Alternative 'levels of information exchange'

Recall that terminal and barge operators in our problem are reluctant to share information to prevent deterioration of their competitive position. To evaluate the value of sharing information, we also consider alternative *levels of information exchange* in our interaction protocol. We define a level of information exchange as the extent to which a terminal gives insight to a barge in the occupation of the terminal during a certain period. 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. A first level of information exchange is waiting profiles, introduced in Section 3.5.1. In our study we consider two other levels of information exchange:

1. *No information.* Terminals reveal no information about their occupation. The barge operator therefore determines his rotation by finding the shortest path along all the terminals concerned.

2. *Yes/No.* A barge operator can ask terminals repeatedly whether a certain arrival time is convenient and the terminal replies only yes or no. To find a convenient rotation the barge operator can delay terminal visits or visit terminals in a different order. The basic idea is that barges can retrieve information about the occupation of terminals, but the information is very limited. The way we implemented this is described in Section 5.2.3. This level of information exchange resembles to a certain extent the interaction protocol implemented in Approach 1 (see Section 2.6.1).

We stress that the levels of information exchange relate to the first phase of the interaction protocol (see Section 3.5.1). The second phase of the interaction protocol, where barge and terminal operator agents make appointments, is the same, regardless of the level of information exchange in the first phase.

3.7 Summary

In this chapter we aimed to design an efficient and effective Multi-Agent system for the barge handling problem. The design of a Multi-Agent system is generally about the strategy of players and the interaction protocol. Both are important to gain *acceptance* and to facilitate the *optimization* of the operations of the actors concerned. We assumed that the strategy of players is to use opportunistic agents that make decisions in the best interest of their principal. We elaborated on the design of the interaction protocol, through which we can determine to a large extent the 'rules of the game'. We mentioned several desired properties of a Multi-Agent system and the associated interaction protocol and we discussed specific difficulties in the realization of these desired properties. We analyzed the interaction protocol proposed in Approach 1 and showed that this protocol has room for improvement. We therefore proposed our own interaction protocol and we reflected on its strengths and weaknesses. In Chapter 5 we specify our conceptual Multi-Agent system and we analyze its performance by means of simulation.

Chapter 4

Performance evaluation

4.1 Introduction

Before we specify the conceptual Multi-Agent system proposed in Chapter 3 and evaluate its performance in Chapter 5, we first describe in this chapter the *approach* we take to evaluate the performance of our Multi-Agent system. We describe successively the research approach we adopt (Section 4.2), our conceptual simulation model (4.3), the off-line benchmark model and solution techniques we apply (Section 4.4), the way we model a scenario (Section 4.5), and the structure in the data sets and the data we use (Section 4.6). We conclude the chapter with a summary in Section 4.7.

4.2 Research approach

To evaluate the performance of the Multi-Agent system proposed in Chapter 3, we adopt the research approach as given by Figure 4.1. Starting from the problem description we develop two models. First, we create alternative Multi-Agent based controls and evaluate the performance by means of simulation. Second, we develop an off-line model to benchmark the simulation results. We call this model the *off-line benchmark*. This off-line benchmark differs from the simulations 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 simulation information is revealed over time (dynamic). Second, it optimizes the problem for a single objective function, whereas in the simulation the performance is the result of interacting self-interested agents.

To facilitate a sound comparison of the two models, we use the same data sets for both the simulations and the off-line benchmark. A data set is a set of scenarios which are stored in a database. The database contains a complete set of all the relevant parameters describing a scenario. Here, on may think of

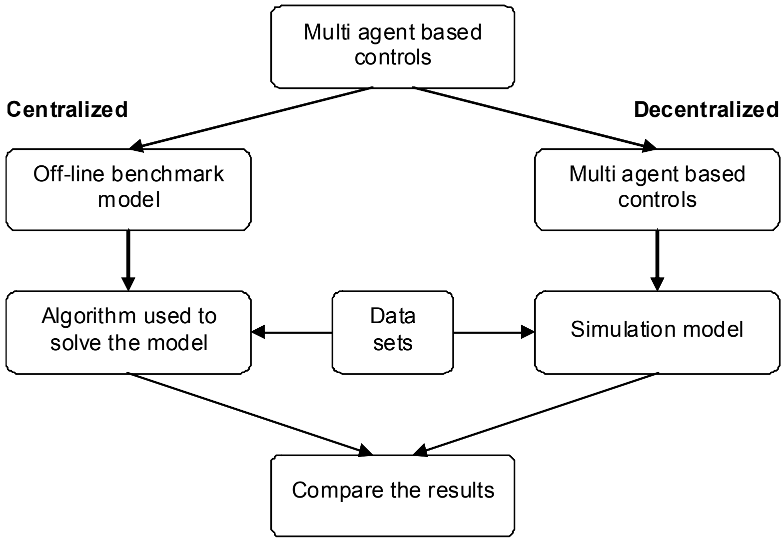


Figure 4.1: Our research approach with respect to the evaluation of the performance of the Multi-Agent system

the capacity of terminals, the arrival times of barges in the port, the number of terminals a barge has to visit, etcetera. A scenario is generated by means of a data set generator, based on certain distributions etcetera. From a single scenario we can have multiple replications, which are different instances of a scenario.

The fact that the off-line benchmark has all information over the whole planning horizon in advance influences the comparison with the simulations. What we mean is that part of the difference between the off-line benchmark and the simulations can be explained by lack of knowledge about the future in the simulations. The magnitude of this impact is hard to estimate, but needs to be taken into account considering the results.

4.3 Conceptual simulation model

In this section we describe the conceptual simulation model. We first describe the basic architecture and motivate our choices. Then we describe the general simulation structure and finally we present the state transition diagrams of both the terminals and barges.

4.3.1 Basic architecture

The initial plan in the research project was to build two simulation models. One simulation model would be built by the University of Twente (made in EmPlant) to investigate the way different decentralized control methods perform. Another simulation model would be developed by the Delft University of Technology (made in DSOL) to do a detailed simulation of the Port of Rotterdam, including the waterways in the port and the current in the rivers. DSOL is an abbreviation for Distributed Simulation Object Library (Jacobs, 2005). The idea was to link the DSOL simulation model to the EmPlant model, which would be invoked during the DSOL simulation to perform specific computations. However, for organizational reasons this has not been established. The basic architecture of the EmPlant model, however, is still based on the idea that the model has to run both stand-alone as well as in cooperation with the DSOL simulation model.

The simulation model we have made is a discrete event simulation. This means that the course of the simulation is event based. An event is an instantaneous occurrence that may change the state of the system (Law and Kelton, 2000). Event based means that after an event the barge or terminal can perform an action resulting in a state transition, after which the state of the system remains unchanged until the next event. The state of the system is described by the state of all the barges and terminals in the system. The simulation model is object oriented, which means that the simulation consists of objects that interact with each other as the system evolves through time. An object consists of data (describing the state of the object at a particular point in time) and methods (describing the actions the object is capable to perform). Data of an object can only be changed by its own methods. For further reading we refer to Law and Kelton (2000).

The simulations we perform are *non-terminating*. This means that there is no 'natural' event that specifies the length of each simulation run (Law and Kelton, 2000). We choose to apply the replication/deletion method, which means that we consider several finite simulation runs. Every simulation run starts and ends with an empty system. This implies that the system needs some time to reach steady state (the warm-up period) and at the end of the simulation leaves steady-state (the cool-down period). Both periods (warm-up and cool-down) are omitted from the simulation data during the analysis.

In the next section we describe the general structure of the simulation model.

4.3.2 General structure of the simulation model

Figure 4.2 presents the parts of our simulation model and the way they are implemented in different software components. We briefly explain the parts of the simulation model. The part 'EmPlant model' consists of four objects,

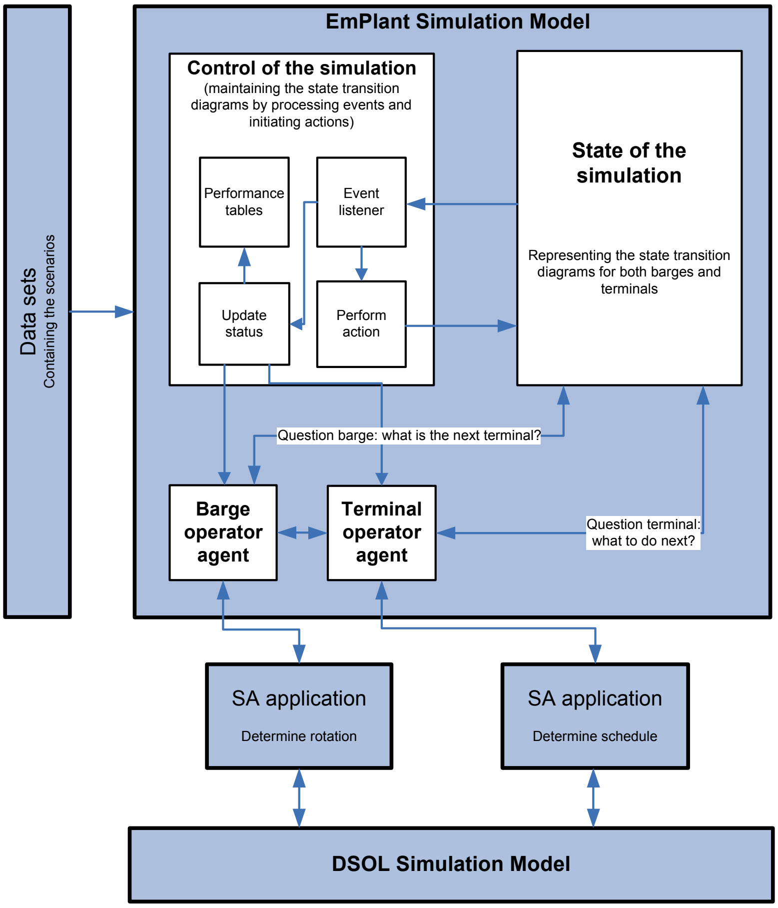


Figure 4.2: General structure of the simulation model and the way the parts are implemented in different software components

namely the state of the simulation, the control of the simulation, the barge operator agent, and the terminal operator agent. In addition, we have the DSOL simulation model, two stand-alone (SA) programs, and the data sets.

We describe the parts (and the objects within the parts) successively.

- *State of the simulation*: The state of the system represents the state of barges and terminals at a particular point in time, and processes the events and actions initiated by the simulation control. The way the state transitions are managed is described in Section 4.3.3, containing the state transition diagrams for both barges and terminals.
- *Control of the simulation:* To be able to communicate with the DSOL

simulation model, we separated the state representation of the simulation from the control of the simulation. This means that all events in the simulation are processed by the event listener, which determines whether only a status update is required or also a state transition. The method 'update status' informs the barge and terminal operator agents about the current status of the simulation model and invokes a method to update the performance tables. The method 'perform action' initiates the required state transitions of the terminals and barges.

- *Barge operator agent:* The barge operator agent object determines for every new arriving barge the best sequence in which it can visit the terminals in its rotation. During the simulation, the barge operator agent is invoked regularly when a barge has finished its handling at a terminal and needs to know the next terminal it has to head to. In our model, the barge operator agent maintains the rotation of the barge and makes updates when necessary. The barge is informed only if it has to change its current activity or start with a new activity. To determine a rotation the barge operator agent can communicate with the concerning terminal operator agents to get insight in the state of the terminals. This information can be passed on to a stand-alone program which returns a rotation that matches best with the objectives of the barge operator agent.

- *Terminal operator agent:* The terminal operator agent can communicate with barge operator agents to make appointments. If a terminal operator agent wants to reconsider its quay schedules, it can invoke a stand-alone application to make a schedule. In addition, the terminal operator agent is invoked regularly during the simulation when a terminal has finished an activity and needs to know what to do next.

- *Stand-alone programs:* The two stand-alone programs (in the form of dynamic link libraries) determine respectively the rotation of a barge and the schedule of a terminal. The reason we implemented these routines in a separate program instead of EmPlant is that executing these algorithms in EmPlant takes too much time. In the stand-alone applications the algorithms can be executed very fast.

- *DSOL*: DSOL is the other simulation model used for the case study of the Port of Rotterdam. DSOL uses the same stand-alone programs. Other routines necessary for computations are transferred from EmPlant to DSOL.

- *Data sets*. On initialization of the simulation all information regarding a specific scenario is taken from a data set. The data sets are used by both the off-line benchmark and the simulation models. The structure of the data sets is described in Section 4.5.

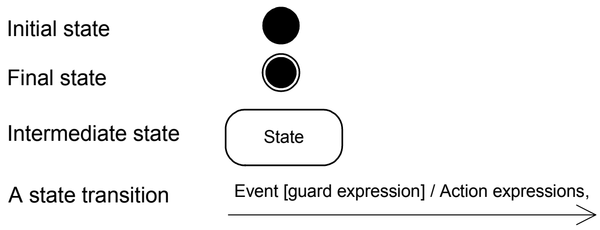


Figure 4.3: Explanation of symbols in the state transition diagram

In the next section we describe in more detail the state transition diagrams representing the state of the system.

4.3.3 State transition diagrams

The state transition diagrams act as basis for the simulation model and show the possible states and state transitions of terminals and barges. A state transition is the transition from one state to another as the result of specific events or actions. In the diagrams we denote possible state transitions by means of an arrow (see Figure 4.3 for a legend of the state transition diagrams).

The initial state in a state transition diagram is represented by a black dot and the final state as a black dot in a white circle. A state is a characterization of the activities that are performed at a specific moment in time. We include the following states in the simulation: a terminal can be either closed, handling a sea vessel, handling a barge, or idle. A barge can be sailing, waiting, mooring, or handling. A transition from one state to another is triggered by an event. On every event the terminal or barge operator agent can decide which action to take, resulting in a specific state transition. Sometimes a guard expression determines which response is required. For example, on the event *finish barge handling* there are many state transitions possible. However, when a sea vessel arrives at the terminal, the terminal will go to the state *handling sea vessel*.

The state transition diagram of the terminal is depicted in Figure 4.4 and the state transition diagram of the barge in Figure 4.5. The diagrams denote the states, events, and actions that are simulated.

4.4 Centralized off-line benchmark

To get an impression of the solution quality of our Multi-Agent system, we compare it with the situation in which we know all information in advance, i.e., we know a priori all the terminals in the port system, the arrival times of barges, the terminals each barge has to visit, the number of containers a barge has to (un)load, the arrival times of sea vessels and their processing time, and

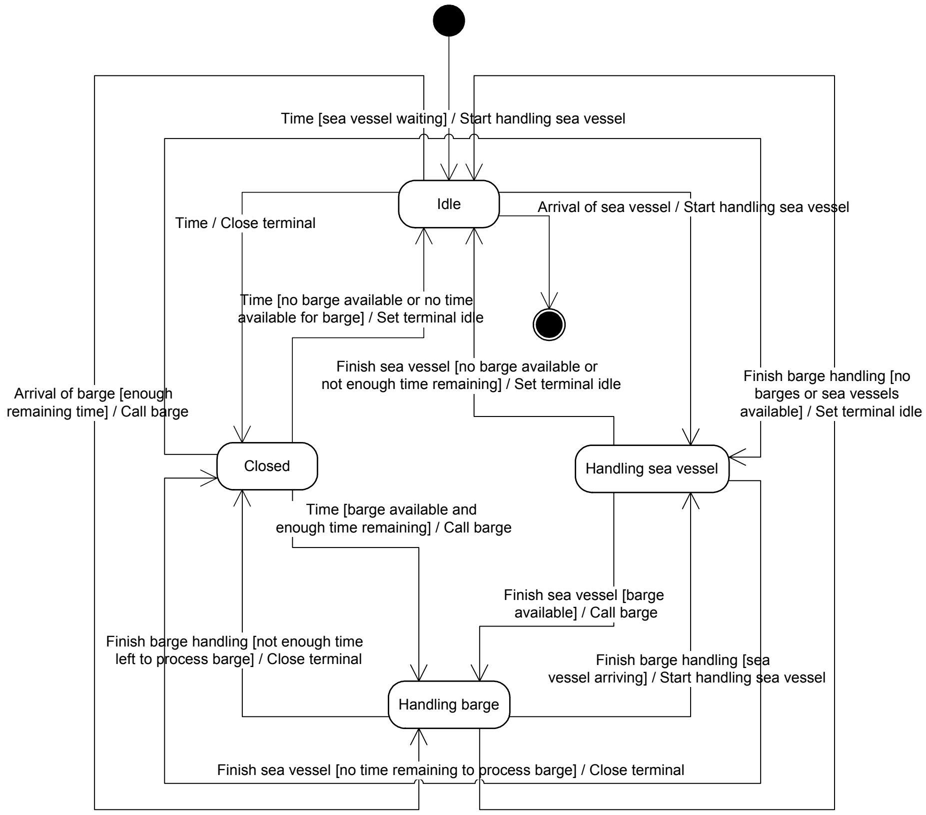


Figure 4.4: State transition diagram of a terminal

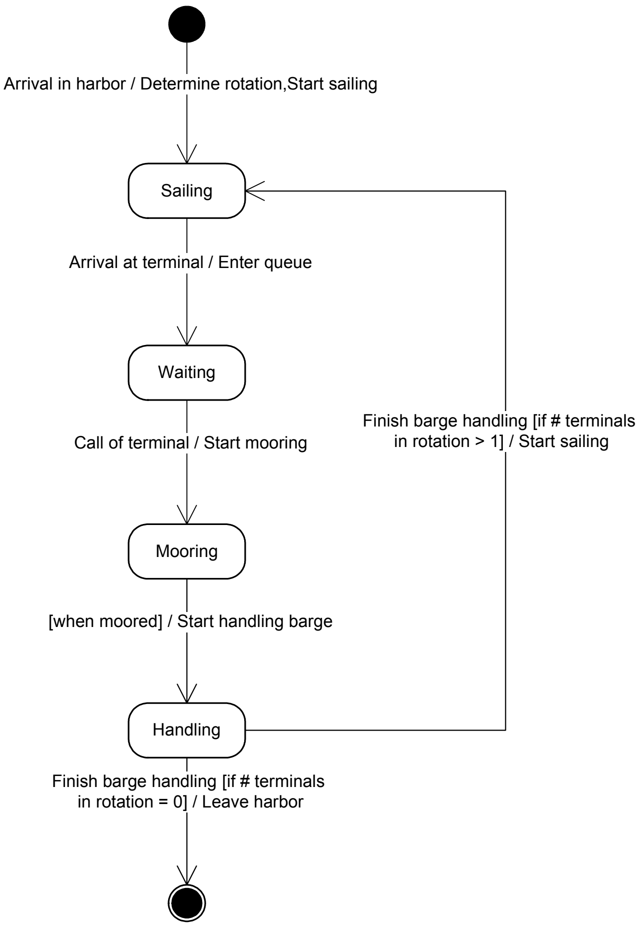


Figure 4.5: State transition diagram of the barge

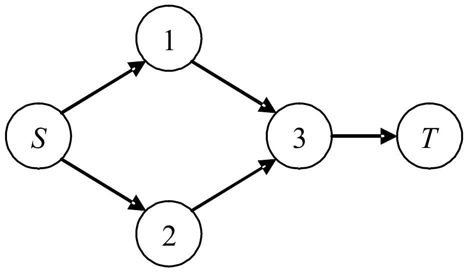


Figure 4.6: Example of an activity-on-node network

the sailing times in the network. In this section we describe the model and the solution methods we apply.

4.4.1 Model

We describe the basic model we use and the way specific characteristics of the problem are modeled.

Basic model

We model the off-line benchmark as a Resource Constrained Project Scheduling Problem (RCPSP; see, for example, Demeulemeester and Herroelen, 1992), as a base model. In the classical RCPSP, a single project has to be scheduled, which consists of a number of activities. These activities have to be processed on a number (not necessarily all) of the available resources, consisting of a number of parallel processors. If an activity is processed at a resource, it can use one or more processors during its processing (for example, 2 processors from resource 1 and 5 processors of resource 2). Between activities precedence relations exist. Precedence relations determine the sequence in which activities have to be processed, i.e., an activity cannot start before all its predecessors have been finished. The RCPSP is $\mathcal{NP}$-hard in the strong sense (see Demeulemeester and Herroelen, 2002).

For solving the RCPSP a graph representation is widely used in the literature. This graph representation is known as an 'activity-on-node network'. See Figure 4.6 for the activity-on-node network for a simple project, consisting of three activities, which all have a node in the graph. There are precedence relations between activity 1 and activity 3, and between activity 2 and activity 3. This means that activity 3 can only start when both activity 1 and activity 2 are finished. In addition to the nodes representing an activity, there are two dummy nodes S and T. Dummy node S represents the project start and dummy node

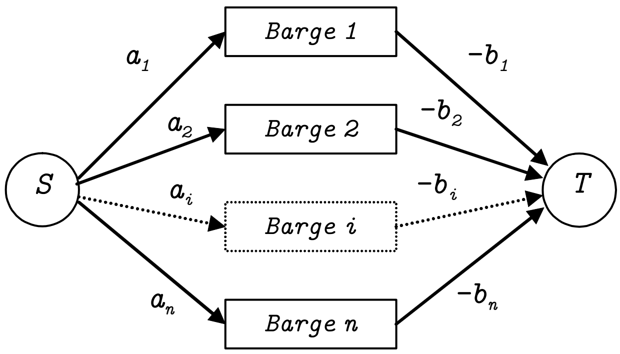


Figure 4.7: Barges are denoted as parallel projects in one super-project that start with dummy node S and end in dummy node T

T represents the project end. If we give each node a weight that is equal to the required processing time, then the length of a longest path from S to a node i is equal to the earliest possible starting time of activity i. The length of a longest path from S to T is equal to the earliest possible completion time for the project.

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

Barges are represented as independent projects within one super-project (see Figure 4.7). The super-project start node precedes all activities of each barge and the super-project end node succeeds all activities of each barge. From the start node an arc goes to each barge (each project) with an associated weight a_i, denoting the arrival time of barge i in the port. From every project an arc goes to the super-project end node with an associated weight $-b_i$. The weight b_i is the due date of barge i, i.e., the planned port departure time denoted in the sailing schedule of the barge. By modeling it this way, the length of a path from S to T is equal to the lateness of the corresponding path. This enables us to compare the time a barge actually leaves the port to the time the barge has to leave the port according to its sailing schedule. We can use this to solve the problem by, e.g., *minimizing the maximum (project) lateness* or *minimizing the total (project) tardiness* of all paths from S to T, i.e., for all barges (projects). See for a definition of project lateness and project tardiness Section 2.7.

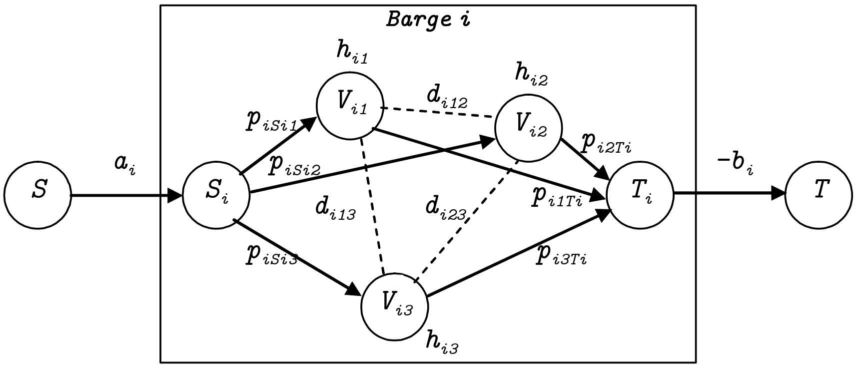


Figure 4.8: A barge is represented as a project consisting of different activities, starting in dummy node S_i and ending in dummy node T_i

If we focus on one barge (one project) we get a picture such as Figure 4.8. Every barge project i starts in dummy node S_i, which precedes all activities of barge i. An activity is denoted by V_{ij}, i.e., the processing of barge i at terminal j. The associated handling time of activity V_{ij} is h_{ij}. From dummy node S_i an arc goes to all activities of barge i, with weight p_{iS_ij} denoting the sailing time from the start node S_i (denoting the entrance point of the port) to terminal j. Similar arcs go from all activities to the dummy end node T_i (denoting the port exit point of the barge). The weights on these arcs are p_{ijT_i}, i.e., the sailing time from terminal j to the port exit point T_i. Between activities a dashed lined is depicted, with associated weights $d_{ij_1j_2}$, denoting the sailing time for barge i to go from terminal j_1 to j_2. We assume that the sailing time from $j_1 \rightarrow j_2$ is equal to the sailing time from $j_2 \rightarrow j_1$. During the scheduling of activities the dashed lines have to be given a direction to indicate the sequence in which the activities are performed. The reason why, is that the sequence in which activities are performed is not determined beforehand, but is part of the decision. To model this, we introduce for each barge a dummy resource that has capacity one and is required by each activity associated with this barge.

We represent opening times of terminals (resources) by so-called resource profiles. Since we assume that the arrival and processing times of sea vessels are deterministic, we choose to model the sea vessels in the resource profiles as well. To deal with closing times of containers we have to register both the activity tardiness and the project tardiness.

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

Time	Capacity
6	2
16	-1

Table 4.1: Example of a resource profile for a terminal with two parallel processors. The terminal opens both processors at time t=6 and closes both processors at time t=16

Time	Capacity
6	1
10	0
14	1
16	-1

Table 4.2: Example of a quay resource profile

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 Section 6.4.2).

Opening times

Opening times of terminals (resources) are represented by so-called *resource profiles*. A resource profile denotes the times at which the capacity of the resource changes as a result of opening or closing of one (or more) parallel processors. Table 4.1 gives an illustration for a terminal with two parallel processors. The times are denoted in hours.

In the example of Table 4.1 the terminal opens two parallel processors at time $t = 6$ and closes them again at time $t = 16$. The '-1' in the column 'capacity' of Table 4.1 denotes closing of the terminal, i.e., all quays are closed due to restricted opening times. Every number in the column 'capacity' greater than or equal to zero denotes the number of processors available for processing.

Sea vessels

In our model we assume deterministic arrival times of sea vessels. We also assume the quay(s) at which a sea vessel is processed to be given. We therefore choose to represent sea vessels also in the resource profiles. This has the advantage that sea vessels are not seen as activities and therefore do not need to be planned. To do so it is necessary to translate terminal resource profiles to quay resource profiles. Quay resource profiles describe the availability of a single quay.

To give an illustration of quay resource profiles, consider Table 4.2. At time $t = 6$ it is opened for barge processing. At time $t = 10$ the quay is occupied by a sea vessel until time $t = 14$. The quay is closed again at time $t = 16$.

Closing times of containers

A container that is shipped to a terminal in the port can have a closing time. This means that the container has to be at the terminal at a specific time. To deal with closing times of containers in the off-line benchmark we have to register both the activity tardiness and the project tardiness (see Section 2.7 for a definition). We indicate the importance of meeting the closing of containers by varying the project tardiness penalty γ. We consider this in detail in Chapter 6.

4.4.2 Solution method and implementation

To solve the resulting problems we use a randomized construction algorithm that is based on the adaptive search algorithm by Kolisch and Drexl (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. We describe first the adaptive search algorithm and then the way we implement the algorithm.

Description of the algorithm

A priority rule scheduling heuristic is a construction heuristic, which constructs a schedule based on a certain (randomized) priority rule. A priority rule scheduling heuristic consists, according to Kolisch (1996), of two components, namely a schedule generation scheme and a priority rule. In general two different schedule generation schemes are distinguished, namely a serial and a parallel method. The *serial method* consists of $n = 1, ..., J$ stages, with J the number of activities to be scheduled. In each stage one activity is selected based on a priority rule and scheduled successively. In every stage, we can distinguish two disjoint sets of activities, namely a *scheduled set* and a *decision set*. The *scheduled set* contains all the activities which are already scheduled. The *decision set* D_n is the set of unscheduled activities with every predecessor being in the scheduled set. In every stage one activity is selected from the decision set, based on a priority rule, and scheduled at its earliest start time feasible with respect to the precedence and resource constraints. Then the algorithm continues to the next stage. The *parallel method* works as follows. It consists of at most J stages, and in every stage one or more activities are scheduled. We can again distinguish two disjoint sets of activities in each stage, namely a *scheduled set* (similar to the serial method) and a *decision set*. In the parallel method the decision set D_n is different from the one in the serial method. Let T_n be the smallest possible start time of all unscheduled activities which are available for scheduling with respect to precedence and resource constraints.

The set D_n then consists of all unscheduled activities which are available for scheduling with respect to precedence and resource constraints and which can start at time T_n. In a stage the set might consist of one or more activities ready for scheduling. Based on a priority rule one activity of the set D_n is selected and scheduled. After scheduling the activity the set D_n is updated, i.e., the scheduled activity is removed as well as all activities which cannot be started anymore at time T_n. If the set D_n is empty after this update, then the algorithm proceeds to the next stage. If the set D_n is not empty after this update, then again one activity of the set is selected and scheduled. This is repeated until all activities in D_n are scheduled and D_n has become empty.

A priority rule, which is independent of a serial or parallel method, is a mapping $v : j \in D_n \rightarrow R_{\geq 0}$, which assigns to each activity j in the decision set D_n a priority value $v(j)$ (Kolisch, 1996). The regret factor ρ_j is determined by comparing the priority value $v(j)$ of an activity with the $\max_{i \in D_n} v(i)$ as follows:

$$\rho_j = \max_{i \in D_n} v(i) - v(j)$$

The probability of activity j to be selected as the next activity is based on the regret factor and determined by:

$$P(j) := \frac{(\rho_j + 1)^\alpha}{\sum\limits_{i \in D_n} (\rho_i + 1)^\alpha}$$

The parameter α controls the extent to which the selection of an activity is biased. If α is high, the selection of an activity is done almost deterministically. When α is low, the selection of an activity is more random.

A construction of a single schedule is called a single-pass or iteration. We apply a random sampling approach, which means that we perform a number of iterations successively, based on the same schedule generation scheme and priority rule. However, after a number of iterations we change the bias in the priority rule by adjusting the value of α. For further reading we refer to Kolisch (1996) and Kolisch and Drexl (1996).

Experiments revealed that our adaptive search solution method appears not to be satisfactory anymore if we deal with non-preemptive downtime, e.g., caused by sea vessels. We have tried several priority rules, but were not able to find a rule that weighs in a proper way the scheduling of an activity in the current stage, and the possible consequences of scheduling the activity in a later stage. The effect of a poor decision can be quite significant, especially when containers have a closing time. We therefore apply, after running the adaptive search algorithm, the following neighborhood searches (in the given order) to improve the schedule:

1. Move an activity on a resource to an earlier time on the same resource. The sequence of activities within a project stays the same.
2. Swap two activities on a resource. The sequence of activities within a project stays the same.
3. Swap two consecutive activities within a project.
4. Swap two non-consecutive activities within a project.

In every step we only consider activities that contribute to the objective function, or activities that are part of a project with project tardiness greater than zero. If an improvement is found, we start again with step 1 (moving activities on a resource). We stop if no improvements can be found after the four mentioned steps.

Implementation of the solution method

We have implemented the algorithm as follows. We use a parallel method to generate a schedule. The reason is that the serial method considers all activities of a barge as candidate activity for scheduling, regardless of the sailing time required to sail from the current position to the next terminal. The parallel method, on the contrary, tries to schedule first the activities at terminals close to the current location. This leads to better results. We use different priority rules for scenarios with and without closing times of containers.

First, we describe the priority rule we use for scenarios without closing times of containers. We define the adjusted activity due date of an activity as the due date of the associated project minus the sailing time from the terminal (associated with the activity) to the port exit point. The contribution of an activity is then equal to the adjusted activity due date augmented with the sailing time from the previous terminal to the terminal associated with the activity and diminished by the sum of the processing times of all unscheduled activities in the associated project. The higher the contribution, the lower the priority of the activity.

Second, we describe the priority rule we use scenarios with closing times of containers. We define the activity tardiness of an activity as the maximum of i) zero and ii) the (planned) completion time of the activity minus its due date. The contribution of an activity is equal to the activity tardiness, augmented with the following term: the project tardiness penalty times the increase in project tardiness of the associated project. The higher the contribution, the higher the priority of the activity.

To determine the decision set D_n in each stage n of the algorithm, we update the release date of all unscheduled activities in each stage of the algorithm. The

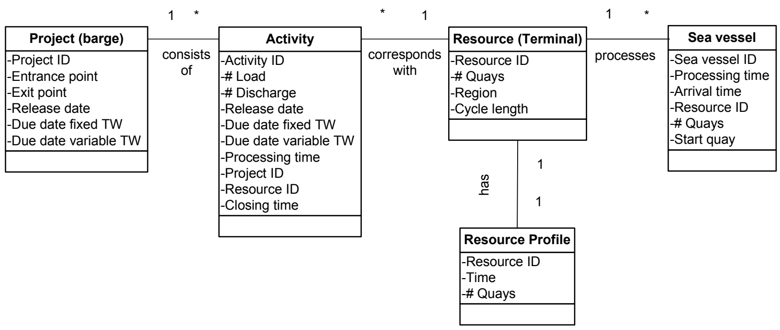


Figure 4.9: An entity relationship diagram to describe a scenario. This diagram also serves as basis for the data sets in which the scenarios are stored

release date of an (unscheduled) activity j_2 in stage n is equal to the completion time of the last scheduled activity j_1 in the same project plus the sailing time from j_1 to j_2.

A difference between opening times (denoted in the regular resource profiles) and sea vessels, is that preemption is allowed in the former but not in the latter case. We explain this in more detail in Section 6.2. In the quay resource profile we can recognize the closing of a terminal by the fact that capacity becomes -1, whereas in case of a sea vessel capacity is always greater than or equal to zero.

In the random sampling procedure we construct 34,000 schedules (iterations). After every 2,000 iterations we decrease the value of parameter α. The construction of a schedule is stopped if the value of the objective function of the (partial) schedule is not better than the best objective value found so far. We use $\alpha \in \{20, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0\}$. We start the random sampling procedure with the highest α.

4.5 Modeling a scenario

To describe the scenarios in our model we use an Entity Relationship Diagram (ERD). This ERD serves also as basis for the database in which we store a scenario. A database can be read on initialization of the simulations or the off-line benchmark. In this section we describe a scenario, the way the database is constructed, and how we generate a scenario.

4.5.1 Description of a scenario

We can describe a scenario by considering the different entities in our model. We distinguish the following entities: resources (terminals), projects (barges), activities, sea vessels, and resource profiles (see Figure 4.9). A *barge* (project) has several attributes, namely an ID, a point where it enters and leaves the port, an arrival time (release date), and a time window in which it has to finish its activities (a due date). We define two due dates, namely for fixed and variable time windows (TW). We elaborate on this in Section 4.6.2.

A *terminal* (resource) has four attributes: a resource ID, a capacity (defined in number of quays), a region where it is located, and a cycle length. Every resource has one resource profile, defining the times at which the terminal is opened or closed. A resource profile has as attributes a resource ID (denoting the resource to which the profile belongs), a time, and a corresponding capacity. For every resource multiple entries for a combination of a time and corresponding capacity may exist. See Section 4.4.1 for a more detailed description of resource profiles. If terminals have cyclic opening times, e.g., every day or every week, then the cycle length denotes the length of one cycle. This reduces the data stored in the database, since we can repeat the resource profile as many times as we need for our planning horizon.

An *activity* (the processing of a barge at a terminal) has several attributes: an activity ID, number of containers to load and unload, a release date (a time after which the activity can be performed, usually the release date of the project), a due date for fixed and variable time windows, a processing time, a project ID (with which barge the activity corresponds), a resource ID (with which terminal the activity corresponds), and a boolean (called closing time) to indicate if an activity has a closing time.

A *sea vessel* has the following attributes: a sea vessel ID, a processing time, an arrival time, a resource ID (to denote at which terminal the vessel has to be processed), the number of quays required for processing, and the (start) quay. The start quay denotes the first quay where the sea vessel processing is planned, e.g., quay 5. If a sea vessel requires more than one quay, the quays following on the start quay are assigned, e.g., quays 6, 7, 8 etcetera.

The ERD in Figure 4.9 forms the basis of the database in which we store a scenario. A necessary element in our model, which is not depicted in the ERD, are the travel times between terminals. These are also stored in the database.

4.5.2 Model to create a scenario

In the previous section (Section 4.5.1) we described a scenario by discussing the different entities and their attributes. In this section we describe the *model* we use to create a scenario. To create a scenario we assume that several parameters (related to terminals, barges, and sea vessels) are given. In Table 4.3 we denote these parameters in *italics*. In addition, we determine several parameters when creating a scenario. These parameters are denoted *not italics* in Table 4.3.

Terminals	**Barges**	**Sea vessels**
# Quays	*Avg and st.dev. of # calls in rotation*	*Avg and st.dev. of the processing times*
Opening times	*Mooring time at a terminal*	Total number
Utilization degree	Total number	The terminal to visit
Avg and st.dev. of the call size	Which hinterland market a barge serves	Start quay
Location	Time window of the barge	Arrival time at the terminal
Sailing time to other terminals	Avg interarrival times of barges	Avg interarrival time of sea vessels
Time to move a container		

Table 4.3: Different parameters related to terminals, barges, and sea vessels, used to create a scenario

Let us describe the model we use to create a scenario. Note that some of the parameters coincide with attributes of entities in Figure 4.9.

One of the starting points to create a scenario is the utilization degree of the terminals. We experienced that it is not always evident why we fix the utilization degree and take it as one of starting points to create a scenario. Let us first define the concept itself. The *utilization degree* of a terminal is equal to the following ratio: the processing times (in minutes) of all the activities performed at the terminal during a certain period of time, divided by the total capacity of the terminal (in minutes) during the same period of time. The capacity of a terminal is equal to the sum of the capacity of its quays. Note that if the terminal is closed during the night for barges, then the period of closing does not contribute to the capacity of the terminal. Recall that we assume the capacity of terminals to be fixed (see Section 1.3). This implies that the utilization degree of a terminal is the *result* of the number of activities a terminal has to perform in a certain period. A terminal cannot influence its utilization degree by increasing the terminal capacity or refusing barges. The latter two options are out of the scope of our model.

The reason why we take the utilization degree of terminals as starting point, is that we can control it very well in this way. This is desirable, since we think that the utilization degree of terminals determines to a large extent the performance of the system. An alternative would be to create a number of barges, assign them to a number of terminals, and see what the resulting utilization degree is. So, in general we have two options. Or we create a random number of barges and take the resulting utilization degree of terminals for granted, or we adjust the number of barges to get a certain desired utilization degree. In both cases we fix one parameter to determine another.

We take the following approach to create the barges in a scenario:

1. We calculate for each terminal j in the port ($j \in \mathcal{N}$) the total number of activities ψ_j that terminal j can perform during a certain period. We do this by calculating the terminal capacity, i.e., # quays · the time (in minutes) the terminal is opened every day · the length of the period (in days) we consider, and multiply this with the desired utilization degree. This results in the number of minutes M_j the terminal has to be busy to realize the desired utilization degree. We then divide M_j by the average processing time of an activity at terminal j, i.e., the average call size · the time to move a container + 2 · mooring time.

2. Then we calculate the total number of activities that can be performed in the port during the specified period of time: $\Psi = \sum_{j \in \mathcal{N}} \psi_j$.

3. We divide Ψ by average number of terminals barges visit during their rotation. This results in the number of barges we have to generate during the specified period of time to obtain the desired utilization degree of the network.

From the total number of barges per day we derive the mean interarrival time. We assign barges proportionally to terminals, such that each terminal realizes the desired utilization degree. Other attributes of barges, terminals, and activities (as presented in Figure 4.9) are determined (randomly) during the scenario creation.

Let us give an illustration. Suppose we have a port with two terminals A and B. Both terminals have one quay and are opened eight hours a day. The utilization degree we want to realize for these terminals is 75% over a period of 10 days. We assume that the average call size is 10 containers. The time to move a container is 3 minutes and the mooring time on arrival and departure is 10 minutes. We proceed as follows. First, we calculate $M_A = M_B = 1$ (quay) · 8 (hours) · 60 (minutes) · 10 (days) · 0.75 (utilization degree) = 3600 minutes. Then $\psi_A = \psi_B = 3600/(10 \cdot 3 + 2 \cdot 10) = 72$ activities. This means that terminal A, as well as terminal B, has to perform 72 activities during 10 days to realize a utilization degree of 75%. The total number of required activities in the port $\Psi = \psi_A + \psi_B = 144$. Suppose every barge visits on average 1.5 terminals, then we need 144 / 1.5 = 96 barges to realize our desired utilization degree of terminal A and B of 75%. The mean interarrival time of the barges is 10 (days) / 96 (barges) = 150 minutes.

For sea vessels we follow a similar computation as for the barges, taking the desired utilization degree of terminals and the capacity available for sea vessels as starting point, and deriving from that the required number of sea vessels.

Data set	Description	Chapter
0	Basic data set	5
1	Increased terminal capacity	5
2	Restricted opening times of terminals	6
3	Unbalanced networks	6
4	Unbalanced networks, opening times, sea vessels, and closing times of containers	6
5	Realistic data set of the Port of Rotterdam	8

Table 4.4: Overview of the structure of the data sets

4.6 Data and data sets

In this section we describe the structure of the data sets we study and the way we determine the data for these data sets.

4.6.1 Structure of the data sets

To evaluate the performance of our models we create different data sets. A data set is a group of scenarios. The successive data sets become increasingly complex. In this way we can study the effect of increasing complexity on the performance of the Multi-Agent system.

Table 4.4 provides an overview of the data sets we consider and in which chapter each data set is studied. Data set 0 is the basic set and forms the basis for data sets 1 to 4. Data set 5 is a separate data set based on realistic data of the Port of Rotterdam and is studied in Chapter 8. In Section 4.6.2 we give a description of the basic data set and in Section 4.6.3 we describe the data sets 1 to 5.

4.6.2 The basic data set

The basic data set serves as basis for the other data sets. In the basic data set we vary the following dimensions (see Table 4.5): the network layout, the number of terminals per region, the utilization degree, and the time windows of the barges. This results in 36 scenarios.

Variable	Value
Network layout	three variants
Number of terminals per region	4 and 9
Utilization degree	50, 75, 90%
Times windows of the barges	fixed and variable

Table 4.5: Dimensions varied in the experiments for the basic data set

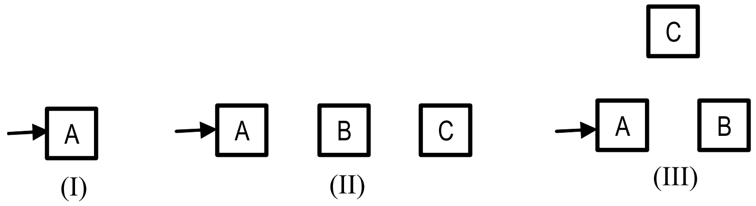


Figure 4.10: Three network layouts: one region (I), three regions in line (II), and three regions in a triangle (III). The arrows represent the port entrance and exit point

In the next sections we describe the network layouts, some parameter settings and distributions, and the time window of the barge.

Network layouts

We consider three different network layouts (see Figure 4.10), which are inspired by the geographical structure of large ports around the world (Rotterdam (layout II), Antwerp (layout III), Hamburg (layout III), Singapore (layout II), and Shanghai (layout II)). We do not claim that our network layouts fit these ports exactly, but they are reasonable approximations. Layout I is added to evaluate the effect of regions on the performance of the system. The arrows in the pictures determine the port entrance and exit points, i.e., the point through which barges enter and leave the port.

We vary the number of terminals per region (either four or nine terminals). A terminal in this basic data set has one quay and is opened 24h a day. All terminals are identical. The sailing time between two terminals only depends on the regions each of the terminals belongs to, not on the Euclidian distance. In the port it is not possible to sail straight from one terminal to another, since there are only a few connecting waterways. We therefore assume that sailing time within a region is fixed. We choose it to be 20 minutes. Sailing through a region (without visiting a terminal) takes 40 minutes. Sailing times between terminals are given by Table 4.6 on a regional level. So, from Table 4.6 we can see that traveling from a terminal in region A to a terminal in region C takes 240 minutes in a line network configuration.

Parameter settings and distributions

The number of barges that visit the port within the planning horizon is derived from the number of terminals in the network, the number of quays per terminal, the average utilization degree, and the average number of terminal visits in a rotation. The interarrival time between barges is exponentially distributed.

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

Table 4.6: Sailing times (in minutes) between terminals belonging to specific regions, specified for different network layouts

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

Table 4.7: Parameter settings

The call size (sum of the containers to load and unload) at a terminal is drawn from a normal distribution with mean 30 containers and a standard deviation of 10 containers. We discretize the normal distribution by rounding to the nearest integer with a minimum value of one. The number of terminal visits (calls) in a rotation is triangularly distributed with a minimum a, maximum b, and mode c. The mode denotes the most frequent value in the distribution. The minimum a is equal to one. The maximum b is equal to minimum of i) the maximum number of terminal visits per rotation and ii) the number of terminals in the network. Mode c is equal to $(a + b)/2$. Other parameters are given in Table 4.7. The distributions used for the call size and the number of terminals in a rotation are chosen in the absence of real data.

Time windows of the barges

Most barges, sailing the river Rhine to Rotterdam, sail according to sailing schedules that are determined once a year. This means that the time a barge is supposed to be in the port is fixed, irrespective of the number of calls in the port, i.e., the time windows of all barges have equal length. With respect to the time windows of barges in our study we choose to consider two extremes. One extreme is a so-called *fixed time window*, i.e., all barges have the same time window irrespective of their *rotation length*. The rotation length denotes the number of terminal visits in a rotation. The other extreme is a so-called *variable time window*, i.e., each barge has an individual time window depending on its rotation length.

Fixed time windows are determined as follows. We assume an average number of calls and an average call size per call. We assume that an average barge visits all regions in the network. Based on that, we can calculate the expected handling time (including mooring time) and sailing time. This is an estimate of the minimum time an average barge needs in the port to finish all its activities.

To add slack for waiting at terminals, we multiply the sum of the handling and sailing time with some slack factor which varies per experimental setting. The slack factor does not depend on the chosen average utilization degree.

The *variable time windows* are calculated as follows. For every barge, we calculate the sum of the handling time (including mooring time) and the expected sailing time. The result is increased with some fixed percentage of slack and a variable percentage depending on the number of terminals in the rotation. The slack per terminal (x) and the slack per rotation (y) depend on the utilization of the network and differs per experimental setting we consider. To give an illustration, assume that we have a network with a utilization degree of 50% and corresponding slack factors $x = 4\%$ and $y = 10\%$. The time window of a barge with, e.g., eight terminals is then equal to $(1 + 10\% + 8 \cdot 4\%)$ times the total handling and sailing time in the rotation.

In the basic data set we use the following slack factors. For the fixed time windows the slack factor is equal to 180%. For the variable time windows the variable slack per terminal (x) is equal to 4%, and the fixed slack per rotation (y) is equal to 10, 50, and 100% if terminals have a utilization degree of 50, 75, and 90% respectively. These factors are determined experimentally, such that a reasonable number of barges can leave the port within their time window.

4.6.3 Data sets 1 to 5

The data sets 1 to 5 are extensions of the basic data set (data set 0). Let us give a brief description of the data sets.

- *Data set 1: Increasing terminal capacity*

  Data set 1 is similar to the basic data set, however, now all terminals have two quays instead of one. Data set 1 consists of 36 scenarios.

- *Data set 2: Restricted opening times of terminals*

  Data set 2 is similar to the basic data set, but now 50% of the terminals in a region have restricted opening times. In Chapter 6.6.2 we introduce this data set in more detail. Data set 2 consists of 36 scenarios.

- *Data set 3: Unbalanced networks*

  Data set 3 is similar to the basic data set, but now we have unbalanced networks. Unbalanced means that terminals differ in capacity (the number of quays), the utilization degree, and the mean and standard deviation of the call size. In addition, we can have busy and quiet regions in the port. In Chapter 6.6.3 we introduce this data set in more detail. Data set 3 consists of 6 scenarios.

- *Data set 4: Unbalanced networks, restricted opening times, sea vessels, and closing times of containers*

  Data set 4 is similar to data set 3, but now the small terminals (1 quay) have restricted opening times and the bigger terminals (>1 quay) handle sea vessels. The small terminals do not handle sea vessels and the larger terminals are opened 24h a day. Part of the containers with a sea terminal destination have a closing time. In Chapter 6.6.4 we introduce this data set in more detail. Data set 4 consists of 4 scenarios.

- *Data set 5: Case Port of Rotterdam*

  Data set 5 is a case study to the Port of Rotterdam and subject of study in Chapter 8. Data set 5 consists of 3 scenarios.

The data sets 0 to 4 are based on fictitious data, inspired by the Port of Rotterdam. Data set 5 is based on realistic data of the Port of Rotterdam.

4.6.4 Data sources

In the data sets 0 to 5 we use certain values for the mooring time, the time to move container, the average number of terminal visits in a rotation, the sailing times, etcetera. These values have been determined based on interviews with experts in the field (especially with a project partner, namely Initi8 B.V.), earlier studies to the barge handling problem, recent studies performed within Transumo, and public domain sources. To our surprise, it was difficult to get realistic and reliable data about the Port of Rotterdam. Registrations of activities in the port turned out to be limited, ambiguous, incomplete, distributed, and sometimes even contradicting. Moreover, actors in the port seem to be reluctant to share information for competitive reasons. It took us considerable effort to obtain realistic data and to create a realistic model of the Port of Rotterdam (see also our discussion in Chapter 8).

4.7 Summary

In this chapter we described the approach we take to evaluate the performance of our Multi-Agent system. We propose to evaluate the Multi-Agent system by means of simulation and to compare the results with an off-line benchmark. The off-line benchmark is a central optimization algorithm, which solves the problem for a single objective and has a priori all information about the planning horizon. We discussed successively our research approach, the conceptual simulation model, the off-line benchmark, the way we model a scenario, the structure in the data sets we study, the data sets we constructed, and data sources we used.

Chapter 5

Waiting profiles

5.1 Introduction

In the previous chapters we introduced our conceptual Multi-Agent system (Chapter 3) and proposed a way to evaluate its performance (Chapter 4). In this chapter[1] we specify the conceptual Multi-Agent system of Chapter 3 and we evaluate its performance in the way we described in Chapter 4. We consider a simplified setting of the problem. In Chapter 6 we extend the model we develop in this chapter with several practical issues. Let us describe the contribution of this chapter, our assumptions, and the outline of the chapter.

5.1.1 Contribution

The aim of this chapter is to gain insight in the functioning of our Multi-Agent system, what the effect is of different levels of information exchanged, and how well the system performs compared with more traditional algorithms. For this purpose, we model our Multi-Agent system of Chapter 3 in more detail, i.e., we model the barge and terminal operator agent, and we describe in detail the concept of waiting profiles. In addition, we evaluate the performance of the Multi-Agent system by means of simulations and comparison with the off-line benchmark. In our Multi-Agent system we consider different levels of information exchange (as introduced in Section 3.6). In this chapter we focus on the data sets 0 and 1 (see Section 4.6).

[1] This chapter is based on the paper: A.M. Douma, J.M.J. Schutten, and P.C. Schuur (2008). Waiting profiles: an efficient protocol for enabling distributed planning of container barge rotations along terminals in the Port of Rotterdam. Accepted for publication in *Transportation Research Part C*. DOI: 10.1016/j.trc.2008.06.003.

5.1.2 Assumptions

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

5.1.3 Outline of the chapter

The outline of the chapter is as follows. In Section 5.2 we model the barge operator agent and in Section 5.4 the terminal operator agent. In Section 5.5 we describe our experimental settings. Section 5.6 describes the results of the experiments and in Section 5.7 we draw our conclusions based on the results.

5.2 The barge operator agent

In our model, the task of the barge operator agent is to plan a rotation for a barge. In this section we describe the mathematical model we use and propose a way the barge operator agent can make decisions based on the available information.

5.2.1 Model and notation

The barge operator agent has to make two decisions, namely i) in which sequence the barge visits the terminals and ii) the specific time each terminal is visited. We assume that every barge operator agent makes these decisions at the moment the concerning barge enters the port. We assume that the information of the terminals is reliable and does not change during the time the barge operator agent needs to make its decisions. It is important that barges make decisions in real time.

Let us consider one specific barge entering the port. The barge operator agent of this barge is assumed to have the following information. First, it knows the set $\mathcal{N}$ of terminals that have to be visited. This set is a subset of all the terminals in the port. For every $j \in \mathcal{N}$ the agent knows how much handling time h_j is needed at the terminal. The handling time consist of the time needed for loading and unloading of containers, as well as the mooring time on arrival and departure. In addition, the agent knows the sailing times $s_{j_1 j_2}$ between every pair of terminals (j_1, j_2), where $j_1, j_2 \in \mathcal{N}$. We assume that both the sailing and the handling time are deterministic. Every barge enters and leaves the port via the *port entrance and exit point,* which is a specific point barges have to pass to access the port.

The aim of the barge is to finish all the activities within a certain time window and to leave the port before the end of this time window. In this study, we have defined the objective of the barge to finish all activities in the port as soon as possible, although in practice there is usually a trade-off between fuel costs and waiting time. We have therefore defined the secondary objective as minimize the total sailing time. This means that if two solutions are equal in terms of total sojourn time in the port, the solution with the least sailing time is preferred. The extent to which a barge operator agent can realize this objective depends both on the actual state of the port and the information it has about this. Assume that the barge operator agent knows the maximum waiting times w_{jt}, for every terminal $j \in \mathcal{N}$ and every arrival moment t. The barge operator assumes that it has to wait the maximum waiting time at a terminal. We can now model the barge operator agent by a *time dependent traveling salesman problem* (TDTSP). We define the time dependent travel time $\tau_{j_1 j_2}(d_{j_1})$, as the sum of the sailing, handling, and maximum waiting time going from terminal j_1 to j_2, where the barge departs at time d_{j_1} at terminal j_1. Let the arrival time at terminal j_2 be denoted by $a_{j_2}(d_{j_1})$. Clearly, $a_{j_2}(d_{j_1}) = d_{j_1} + s_{j_1 j_2}$. Furthermore, the time dependent travel time $\tau_{j_1 j_2}(d_{j_1})$ is given by:

$$\tau_{j_1 j_2}(d_{j_1}) = s_{j_1 j_2} + w_{j_2, a_{j_2}(d_{j_1})} + h_{j_2}$$

Consequently, $\tau_{j_1 j_2}(d_{j_1})$ gives the maximum amount of time between leaving j_1 and leaving j_2. Aim of the barge is to find a rotation along all terminals it has to visit, such that the sum of the time dependent travel times is minimized. We assume that a barge visits a terminal only once and that the handling process is never interrupted after the handling has started.

5.2.2 Algorithms applied to solve the model

In the last decades, several studies have been devoted to the time dependent traveling salesman problem (TDTSP) and the time dependent vehicle routing problem (TDVRP). We mention especially the contributions of Malandraki and Dial (1996), Fleischmann et al. (2004), and Ichoua et al. (2006). For a more extensive literature review, we refer to these papers as well. The three mentioned papers each follow a different approach to solve the TDTSP or TDVRP. Malandraki and Dial (1996) present a heuristic based on Dynamic Programming. This heuristic solves relatively small instances near optimal, under the assumption that the *non-passing property* holds with respect to travel times. Non-passing means that earlier departure from the origin node guarantees earlier arrival at the destination node (Ahn and Shin, 1991). Fleischmann et al. (2004) present an extension of the Clark-and-Wright algorithm using a modified savings formula (based on the work of Paessens, 1988). They repeat their construction algorithms for different combinations of parameters in the savings formula and use a 2-opt procedure after each construction to improve the route. Finally, Ichoua et al. (2006) present (among others) a parallel tabu search algorithm and show the performance of this algorithm on several test instances.

An important constraint in our model is that the TDTSP needs to be solved in real time. For convenience, we have defined this (on a Pentium IV 3.4 GHz dual core computer) as less than 200ms, although this is by no means restrictive. For that reason, we did not consider the algorithm of Ichoua et al. (2006), but first implemented a savings algorithm with the following savings formula (the zero denotes the depot):

$$\phi_{j_1 j_2} = \tau_{j_1 0}\left(d_{j_1}\right) + \tau_{0 j_2}\left(d_0\right) - g \cdot \tau_{j_1 j_2}\left(d_{j_1}\right) + f \cdot \left|\tau_{j_1 0}\left(d_{j_1}\right) - \tau_{0 j_2}\left(d_0\right)\right|$$

and we constructed several routes for $g \in [0,3]$ and $f \in [0,1]$ (see Paessens, 1988). Every construction is followed by a 2-opt procedure with 1000 iterations. To compare the performance of this heuristic we also implemented a depth-first branch-and-bound algorithm (B&B). However, from preliminary tests we did two observations. First, it turned out that we could solve the TDTSP optimally using B&B within 100ms for instances with up to about seven to eight terminals. Second, the savings based algorithm resulted in solutions with an optimality gap varying from 10-40%, which is not satisfying.

In the Port of Rotterdam, barges visit at most twenty terminals, which means that we need an algorithm that is able to solve instances with up to about twenty terminals. We therefore implemented the DP-heuristic of Malandraki and Dial. The DP-heuristic of Malandraki and Dial (1996) constructs rotations by adding to each partial tour in the previous stage a (yet unvisited) node, which becomes the new end node of the partial tour. A state is a set of terminals that are in the partial tour, and an end node. The value of the state is the minimum time to travel from the depot, along all the terminals in the partial tour and ending in the end node. To limit the growth of the state space and to keep the computation time of the DP tractable, Malandraki and Dial (1996) retain in every stage the best H states, i.e., the states that most likely result in the optimal rotation. We compared the algorithm with an exact branch-and-bound algorithm for different experimental settings (for more details, see Appendix A). In the experiments, we took H=1000. From the results we conclude the following:

- The DP-heuristic of Malandraki and Dial yields a solution within 1% of the optimum within 200ms for all experimental settings we considered.
- For instances up to seven to eight terminals, depth-first branch-and-bound is faster than the DP-heuristic of Malandraki and Dial.

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

5.2.3 Dealing with different levels of information exchange

In our experiments we also consider the two alternative levels of information exchange introduced in Section 3.6. We implement these levels of information exchange as follows:

1. *No information.* We assume that the barge operator agent has no information about the state of the network and plans a rotation only based on the sailing time between terminals. We still apply the same algorithms as described before (Section 5.2.2), but now the waiting time w_{jt} is equal to zero at every terminal $j \in \mathcal{N}$ and departure time t.
2. *Yes/No.* As in option 1, the barge operator agent has no information about the state of the network. Every time a barge arrives in the port, the corresponding barge operator agent determines the ten rotations with the shortest sailing time. These rotations are ranked in descending order of total sailing time. Then the barge operator agent starts with the first ranked rotation and tries to find out what the first possible start time is at the first terminal in the rotation. It therefore asks whether a certain arrival is ok and the terminal operator agent replies yes or no. If the terminal operator agent replies no, the barge operator agent proposes a new time and the terminal operator agent answers again (yes/no). Once the terminal operator agent has replied yes, the barge operator agent makes an appointment with this terminal operator agent and proceeds to the second terminal in the rotation and tries to find out the first possible start time at this terminal by proposing specific arrival times repeatedly. Once all terminal operator agents in the rotation have agreed

on a specific arrival time, then the barge operator agent calculates the total sojourn time in the port. If this time exceeds the time window of the barge, it cancels all appointments with the terminal operator agents and it considers the second ranked rotation in the same way as the first rotation. If a rotation can be completed within the time window, then the barge operator agent accepts the appointments made for this rotation. If the barge operator agent has arrived at the tenth ranked rotation and still did not find a feasible rotation, it stops considering other rotations and accepts the appointments made for the tenth terminal. This approach is similar to the model used in a previous study (Connekt, 2003) and is inspired by practice, where barge operators improve their rotation by contacting terminals iteratively, until they are satisfied with a certain rotation or they simply stop putting effort in finding improvements if it takes too much time.

Recall that these two levels of information exchange relate to the first phase of the barge operator agents' decision process when the agent determines the best rotation for its principal (see Section 3.5.1).

5.3 In between: simplified terminology

The reader may agree that it is more convenient to use the words *terminal* or *barge*, when it is perfectly clear that we either refer to the *operator* or the *operator agent* of the respective barge or terminal. Therefore, in the sequel we use the words terminal, terminal operator, and terminal operator agent interchangeably, when there is no confusion about which of the three we actually refer to. The same holds for the words barge, barge operator, and barge operator agent.

5.4 The terminal operator agent

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

5.4.1 Model and notation

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

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

An appointment is an agreement between the barge and the terminal operator (see Section 3.5.1). By making an appointment, the barge promises the terminal to be present at a certain time, namely the *latest arrival time (LAT)*. Based on the waiting profile, issued in the first phase of the barge's decision process, the terminal derives the *maximum waiting time (MWT)* for the LAT and guarantees the barge a *latest starting time (LST)*. It holds that LST=LAT+MWT. After the appointment is made, the terminal schedules the barge tentatively such that the appointment with the barge is not violated, nor the appointments made with other barges. In the schedule (which is hidden for the barge) every barge has a *planned starting time (PST)* and an *expected departure time (EDT)*. The EDT is equal to the PST and the *processing time (PT)* of the barge. The PT is equal to the handling time (h_j) of the barge, as introduced in Section 5.2.1, and is revealed by the barge in the first phase of its decision process.

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

Table 5.1: Example of a quay schedule and the corresponding agreements

To give an illustration, consider the agreements made with barge B1 in Table 5.1. Barge B1 has promised to arrive not later than $t = 10$ and it has a guarantee that its processing will be started not later than time $t = 20$ (if it arrives in time). Hidden for the barge is the schedule of the terminal, denoting the PST, PT, and EDT, respectively 10, 10, and 20 for this barge. Barge B2 has made similar appointments as shown in Table 5.1.

5.4.2 Keeping appointments

The keeping appointments task is the scheduling of *expected barges* on the quays such that no appointment is violated. Expected barges are barges that already made an appointment with the terminal and still need to be processed (see Table 5.1 for an example). From time to time, it might be beneficial for a terminal to reschedule barges on the quays, e.g., when a barge arrives earlier than its latest arrival time. This is possible when a barge does not need to wait the maximum waiting time at previous terminal(s).

To schedule the expected barges, the terminal has to decide how the barges are assigned to the available quays, such that all appointments are met, i.e., the handling of no barge is started later than its LST. Starting the handling of a barge later than its LST is not permitted in our model.

To assign the expected barges to the available quays of the terminal, we apply list-scheduling in combination with depth-first branch-and-bound (see Schutten, 1996). We define the *terminal lateness* of a barge as the planned starting time of this barge minus its latest starting time. Our primary scheduling objective is to minimize the maximum terminal lateness (which needs to be less than or equal to zero) and the secondary objective is minimizing the average terminal lateness of all expected barges. A good initial solution to the branch-and-bound algorithm is usually to sort barges based on latest starting time (LST), which results in a strong bound and makes the algorithm fast.

5.4.3 Making appointments: how to construct the terminal waiting profile

When a barge asks for waiting times, the terminal operator makes a waiting profile expressing the *maximum* waiting time until the handling of the barge is started, for every arrival moment during a certain planning horizon. The determination of the waiting profile consists of several steps. We first make a waiting profile for each quay and then combine the quay waiting profiles to a terminal waiting profile.

Waiting profile for one quay

Let us first consider the generation of a waiting profile for a single quay. Assume that we have a schedule and that we have to make a waiting profile for barge b with expected processing time PT. To explain our approach, we use the example data in Table 5.1. In our example, barge b has a PT equal to 15.

To determine the quay waiting profile, we first determine all *start intervals* for barge b in the current quay schedule. A start interval is defined as a time interval in which the handling of the concerning barge could be started immediately, without violating any appointments made with other barges. The start intervals are determined by considering all possible insertion points i in the schedule. Insertion point i means insertion after the i^{th} barge in the schedule ($i = 0$ means insertion before all scheduled barges). At every insertion point, all barges before the insertion point are replanned as early as possible (i.e., without violating any appointment) and barges after the insertion point as late as possible. Remark that we do not re-sequence the barges. Let m_i and n_i be the respective start and end of start interval i. Then m_i becomes equal to the EDT of barge i (the last scheduled barge before insertion point i), and n_i is equal to the $PST_{i+1} - PT$ (recall that a gap represents possible starting times of the barge). If $i = 0$, then m_i is equal to the actual time. If i is equal to the number of planned barges, then n_i is set to infinity. If $n_i < m_i$ then the start interval is empty and we do not consider it anymore. If $m_i \leq n_i$ then we remain the start interval and add it to the list of start intervals. For every start interval, we also remember the corresponding insertion point in the schedule, to be able to quickly insert the barge after it has announced its arrival time.

Repeating the steps for all possible insertion points, results in a list with start intervals ordered on start time. Table 5.2 presents the start intervals for the example schedule of Table 5.1.

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

Table 5.2: The start intervals in the example

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

To determine the waiting profile we evaluate, starting from the current time t_0, for all possible arrival times t in the time interval $[t_0, \infty)$ the corresponding maximum waiting time. We do this as follows. If arrival time t is in one of the start intervals, the corresponding waiting time $w_t = 0$. If arrival time t is not an element of one of the start intervals, the corresponding maximum waiting time w_t is equal to the time until the start time of the first start interval after t. The latter means that maximum waiting times are linear functions in time, which means that we only need to evaluate the times $t \in [t_0, \infty)$ at which the maximum waiting times change. Changes correspond directly with the start times of all start intervals, including the actual time t_0. Considering our example we denote the resulting waiting profile in Table 5.3, where $t_0 = 0$.

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

Table 5.3: Data representation of a waiting profile

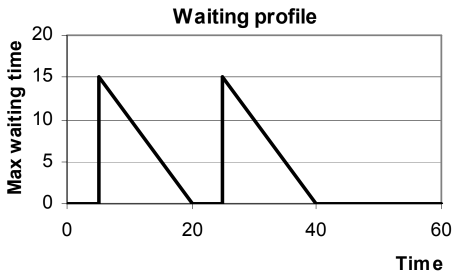


Figure 5.1: Example waiting profile for a single quay

The maximum waiting times are given by linear functions with slope -1 and a minimum of zero. See Figure 5.1 for a visual representation.

Waiting profile for the terminal

To determine the waiting profile for the terminal we take for every time $t \in [t_0, \infty)$ the minimum of the maximum waiting times at each of the quays. To do so we only need to evaluate the times t at which the maximum waiting time changes at one of the quays and to calculate the corresponding maximum waiting time at the other quays. Consider the following example with two quays and two different waiting profiles (see Table 5.4).

Quay 1

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

Quay 2

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

Table 5.4: Example of a waiting profile of two quays

In the example it is clear that the maximum waiting time changes at time $t = 0, 5, 10, 25$. Since a waiting profile can be expressed by linear functions, the resulting terminal profile can easily be obtained (see Table 5.5 and Figure 5.2).

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

Table 5.5: Data representation of the terminal waiting profile

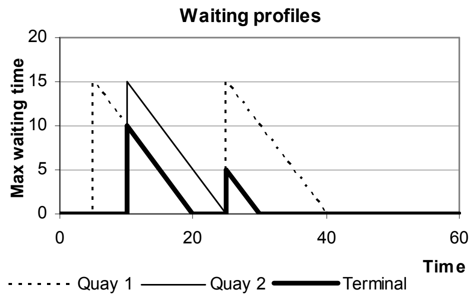


Figure 5.2: The terminal waiting profile derived from the quay waiting profiles

The minimum maximum waiting time in the waiting profile is now equal to zero. This means that a barge can get a maximum waiting time of zero after its arrival time. The result is that the terminal operator has no flexibility in its schedule anymore, since this barge must be started at the agreed time. To increase flexibility, we can add some slack s to the maximum waiting time, which means that we augment the maximum waiting time at every time t with an amount s. Note that $s = 0$ resembles the issuing of time windows. In the experiments, we evaluate different values of s to see the impact on the performance of both the terminals and the barges (see Section 5.5).

5.5 Experimental settings

To evaluate our Multi-Agent system we consider different scenarios and compare the results with the off-line scheduling algorithm (see Section 4.4). We also consider, besides waiting profiles, two alternative levels of information exchange (see Section 3.6). In our experiments we consider data sets 0 and 1, introduced in Section 4.6. This results in 36 scenarios per data set, varying along the dimensions: network layout, number of terminals per region, utilization degree of terminals, and the time window of the barge.

Every scenario is evaluated using the off-line benchmark and simulations of the Multi-Agent system. For the Multi-Agent simulations, we consider waiting profiles, no information, and yes/no information exchange. In case of waiting profiles we also vary the value of s (the additional flexibility in the waiting profile) for $s \in \{0, 30, 60\}$, with s in minutes.

All scenarios have a run length of 100 days. We apply a warm-up period of ten days (which proves to be sufficiently long) and a cool-down period of three days. The length of these periods are determined by a graphical procedure (the method of Welch (1983)). We evaluated the development of the average project lateness by taking the moving average over the successive barges until the average project lateness becomes more or less stable. The length of the warm-up period is especially important for higher utilized networks. In that case it takes some time before the system is fully loaded. To calculate the results of the off-line benchmark, we take only nineteen days, with the same warm-up and cool-down period as in the simulations. The reason why we consider nineteen days is that this is still computationally viable to compute with the off-line benchmark, whereas the number of days that can be used for the analysis, after subtracting the warm-up and cool-down period, is still acceptable. The primary objective of the off-line benchmark is minimizing the sum of the project tardiness, and the secondary objective to minimize the fraction of late barges. Since it is possible that the total project tardiness for specific scenarios is zero, we also implemented the standard RCPSP objective to the problem, i.e., the primary objective is to minimize the maximum project lateness and the secondary objective is to minimize the total project lateness. In fact we have two benchmarks; one based on total project tardiness and one on maximum project lateness.

Recall that all handling and sailing times are deterministic. As time unit we use minutes in our experiments. To speed up the simulation we let the terminals reconsider their quay schedules not on every arriving barge, but after every ten arriving barges.

5.6 Results

In this section we describe the results of our experiments. In Section 5.6.1 we compare the results of the different models we consider, namely the Multi-Agent system (including the different levels of information exchange) and the off-line benchmark. Then we take a closer look at the waiting and sailing time in the rotation in Section 5.6.2 to gain insight in the way our Multi-Agent system performs.

5.6.1 Comparing the different models

If we consider the performance of different levels of information exchange in the Multi-Agent system, we find major differences. In Figure 5.3 and 5.4 we show the differences in average project tardiness. We averaged the results over all considered scenarios, specified for the number of quays (one and two), and the utilization degree of the terminal (50% and 90%).

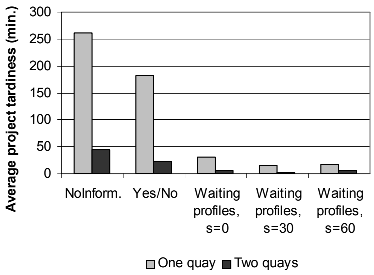


Figure 5.3: The average project tardiness per barge averaged over all considered scenarios, specified for the number of quays per terminal and a utilization degree of 50%

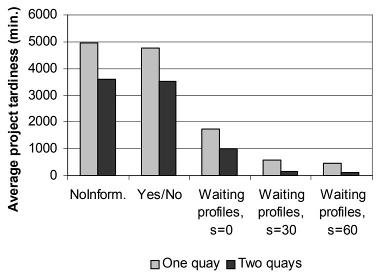


Figure 5.4: The average project tardiness per barge averaged over all considered scenarios, specified for the number of quays per terminal and a utilization degree of 90%

The aim of Figure 5.3 and 5.4 is not to give an estimate of the average project tardiness of the barges, but to depict patterns that arise in each of the scenarios. Looking at Figure 5.3 and 5.4, we make the following observations. First, the average project tardiness correlates with the utilization degree of the terminals. Second, the average project tardiness reduces significantly when all terminals increase their capacity (from one to two quays) for the same utilization degree. This effect is in accordance with queuing theory, especially theory on queuing systems consisting of multiple servers (see, e.g., Gross and Harris, 1998). Third, the effect of adding slack s to the waiting profiles (see Section 5.4.3) depends on the average utilization degree of the terminals.

We depict the results for the average project tardiness, but similar results are found considering the fraction of late barges and the average project lateness. If we take a closer look at the results, we conclude that issuing waiting profiles performs significantly better for all scenarios than yes/no or no information exchange. Yes/no information exchange in turn performs significantly better than no information exchange in terms of fraction of late barges, but not necessarily in terms of average project lateness or average project tardiness.

For network layout II and III (see Figure 4.10), our experiments show that the relative performance of the different levels of information exchange are similar, although the average performance in the 'line network' is slightly better than in the 'triangle network'. This is because barges visit region B only once in the latter network and twice in the 'line network'. Therefore, in the 'line network' barges have the possibility to visit the terminals in region B partly before and partly after visiting region C.

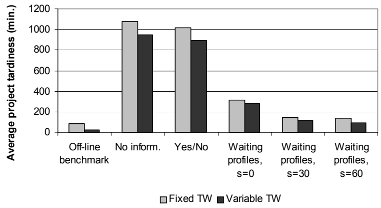


Figure 5.5: Average project tardiness per barge averaged over all scenarios and compared with the off-line benchmark

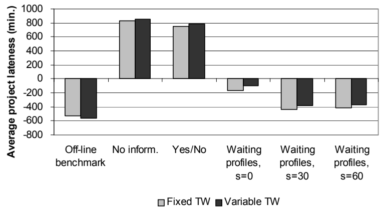


Figure 5.6: Average project lateness per barge averaged over all scenarios and compared with the off-line benchmark

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

Table 5.6: The best amount of slack in minutes (considering the total project tardiness). We considered all 72 scenarios and specified the results for fixed and variable time windows, and the different utilization degrees considered

To get an impression of the quality of the Multi-Agent approaches, we make a comparison with the off-line benchmark based on a subset of all scenarios. We consider only the scenarios with nine terminals per region, since they are the most complex and realistic ones. From these scenarios, we consider only the ones with network layout I and II, since the results of network layout II and III are comparable. We analyzed the simulation results for the same set of days as we consider for the off-line benchmark. Figure 5.5 depicts the average project tardiness and Figure 5.6 the average project lateness of every simulated level of information exchange, compared to the off-line benchmark. Taking into account that the off-line benchmark has all information in advance and weighs the interests of different barges against each other, we think that the decentralized control based on issuing waiting profiles is very promising in the port. Not only because of the acceptability, but also considering the increase in performance that could be achieved.

If we take a closer look at the waiting profiles, we observe that the benefit of a certain amount of slack added to the waiting profile clearly depends on the utilization degree of the terminals in the network and whether barges use fixed or variable time windows. The best amount of slack for different utilization degrees, and fixed or variable time windows is denoted in Table 5.6. The reason why more slack is needed in case of variable compared to fixed time windows is explained in Section 7.2.5.

With respect to a Multi-Agent system using waiting profiles, we see that both terminals and barges benefit from adding slack to the waiting profile. The increased flexibility gives the terminal more opportunities to compactify its quay schedules, resulting in a higher quay utilization and a higher throughput.

5.6.2 Analysis of waiting and sailing times

Let us now take a closer look at the waiting and sailing times of barges. First, we explain how we can check the consistency of our simulation model considering the waiting time in our model. Second, we take a closer look at the total waiting time in a rotation related to the length of the rotation. Third, we analyze the average waiting time per terminal and relate this to the performance of barges and terminals. Finally, we analyze the relation between the processing time and the corresponding waiting time at a terminal.

Checking the consistency of the simulation model

A way to check the consistency of our simulation model is to look at the waiting time announced by the terminals at the time the barge plans its rotation and the waiting time realized by the barge during execution. If we use the waiting profiles without slack, the announced and the realized waiting time must be exactly the same. The reason is that every barge will be processed at the agreed time, since no terminal can process a barge earlier or it would have announced a shorter waiting time. In all simulations we did this was indeed true.

In case we add slack to the waiting profile, it might happen that a barge has to wait less than its maximum waiting time and arrives earlier at the next terminal. This terminal can then decide to process the barge earlier if no other appointments are violated. A difference between the announced and the realized waiting time is to be expected in this case. Barges can possibly benefit from this by estimating the waiting time instead of using the maximum waiting time during the planning of the rotation. However, it might then happen that a barge makes a wrong estimate of the waiting time and has to replan its rotation if it cannot arrive timely at a terminal. This is a topic for future research.

Total waiting time related to the rotation length

In our simulations we found an interesting relation between the *average total waiting time in a rotation* and the *number of calls (terminal visits) in a rotation*, especially for utilization degrees of 75% and higher. In Figure 5.7 we depict this relation for a scenario with network layout II, 9 terminals per region, 90% utilization degree, all terminals having one quay. From the picture we conclude the following. First, the average total waiting time reduces significantly if more slack is added to the waiting profile. Second, the average total waiting time in a rotation reduces if a barge visits more terminals in its rotation. More specifically, the sum of the average total waiting and sailing time seems to be more or less constant in the number of terminals a barge visits (see, Figure 5.8). This may seem counter intuitive. The explanation for this result is that

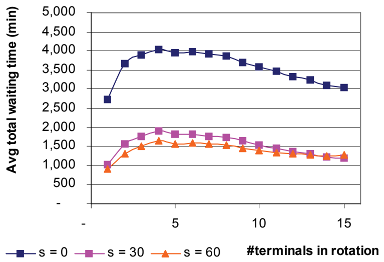


Figure 5.7: Average total waiting time for different levels of slack and different rotation lengths

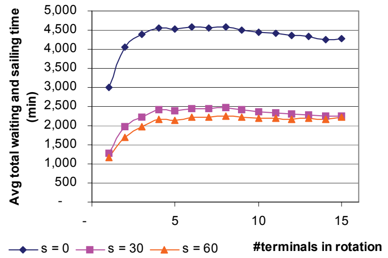


Figure 5.8: Average total waiting and sailing time for different levels of slack and different rotation lengths

barges, if they visit more terminals, use the waiting time at terminal A to visit another terminal B. Waiting time is then used partly for sailing, handling, and waiting at terminal B. Especially when a barge visits more terminals, it has more options to minimize its total waiting time in a rotation.

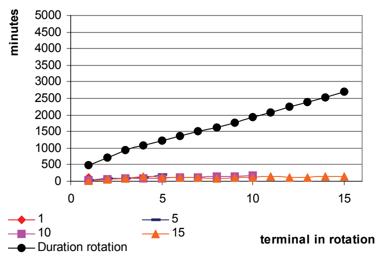


Figure 5.9: The avg waiting time at a terminal measured at time t_0 as function of the position of the terminal in a rotation with a specific length (1, 5, 10, and 15). Results for a utilization degree of terminals of 50%

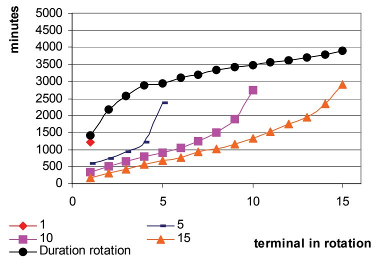


Figure 5.10: The avg waiting time at a terminal measured at time t_0 as function of the position of the terminal in a rotation with a specific length (1, 5, 10, and 15). Results for a utilization degree of terminals of 90%

Let us take a closer look at the waiting times at terminals when barges plan their rotation. We are interested in the correlation between the waiting times at terminals and the order in which a barge visits the terminals concerned. Consider a barge with a specific rotation length. In order to plan a rotation, the barge requests waiting profiles at the terminals that have to be visited. At the time these waiting profiles are issued to the barge, we (as researchers) collect these waiting profiles and extract from these the waiting time at each terminal at that very moment. Then we consider the rotation a barge has planned based on these waiting profiles and we link the observed waiting time at each terminal to the position of that terminal in the rotation. We average the observed waiting time at the n^{th} terminal in the rotation of all barges with the same rotation length. In the Figures 5.9 and 5.10 we depict the average waiting time at the n^{th} terminal in the rotation, at the time a barge plans its rotation. To create the figures we use the results for waiting profile communication with 30 minutes slack and we specified the results for different utilization degrees. For comparison, we also depict the average duration of a rotation for different rotation lengths. Note that the x-axis for the average duration denotes the rotation length, whereas for the other lines the x-axis denotes the n^{th} terminal visited in the rotation. The results indicate that terminals with long waiting times -as perceived at the very moment of planning- are usually visited at the end of the rotation. In general, however, most terminals do not have long waiting times at the time a barge plans its rotation, which means that they can be visited rather easily in the short run.

Considering Figure 5.9 and especially Figure 5.10 (results for a utilization degree of 90%) we gain the following insights. First, if the utilization of the network increases, then the waiting time at a terminal -as perceived at the very moment of planning- determines to a large extent the position of this terminal in the rotation. The longer the waiting time the later the terminal will be visited in the rotation. Second, the waiting times at terminals determine to a large extent the duration of the rotation, especially the waiting time at the last terminal. Third, barges with short rotations face relatively much waiting time at a terminal compared to barges with long rotations. In a network with a low utilization degree there seems to be no significant correlation between the waiting time at terminal -as perceived at the very moment of planning- and the position of the terminal in the rotation.

Recall that barges use the maximum waiting time given by the waiting profiles to plan their rotation. This means that if terminals add more slack to their waiting profile, then the rotations of barges will be spread over a longer period of time. The result is that terminals have less 'dense' quay schedules and therefore, there are more opportunities for future barges to visit a terminal in the short run. We can see this effect if we look at the waiting time at terminals as perceived at the time a barge plans its rotation. If terminals apply 60 instead of 0 minutes slack, then the average waiting at a terminal, for a barge with rotation length one, drops from 2,500 to 1,000 minutes.

Based on Figure 5.10 it is clear that usually the terminal with the longest waiting time is put at the end of the rotation. We find a similar result if we look at the average waiting time a barge faces at successive terminals. In Figures 5.11 and 5.12 we depict the waiting time of barges with different rotation lengths experienced at successive terminals during execution of a rotation. The results are depicted for the same scenario as we used for Figure 5.7 and 5.8, using waiting profile communication with 30 minutes slack. The reason why we also depict the 90% utilization degree (Figure 5.12) is that in this setting

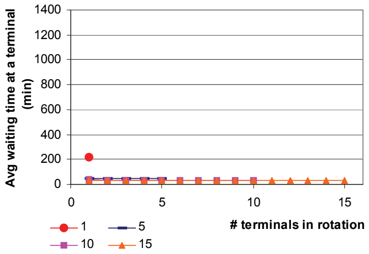


Figure 5.11: Waiting time barges face on average at the n^{th} terminal in their rotation during execution of their rotation. Results for 50% utilization degree and different rotation lengths (1, 5, 10, and 15)

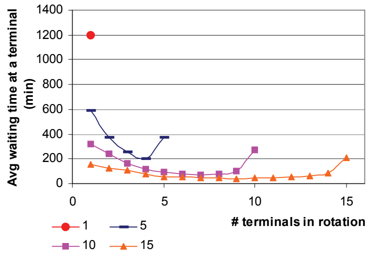


Figure 5.12: Waiting time barges face on average at the n^{th} terminal in their rotation during execution of their rotation. Results for 90% utilization degree and different rotation lengths (1, 5, 10, and 15)

the impact of waiting times on the duration of barges becomes most apparent. Typical in Figure 5.12 is the bend at the end of each line, which means that barges wait relatively more at the last terminal than at previous terminals. From the picture it becomes clear again that barges with long rotations face on average less waiting time per terminal.

Based on the figures we just discussed we can conclude that if the terminals are highly utilized, it seems not to really matter for barges how many terminals they visit with respect to average total waiting and sailing time. For terminals, however, it does matter since longer rotations imply smaller call sizes at terminals which implies in turn more barge visits per time unit, and every visit results in mooring time during which the quay cranes are not productive.

Analysis of the average waiting time per terminal

If we consider the average waiting time per terminal and relate this to the performance of barges and terminals, then we find the following pattern. The best amount of slack in a waiting profile corresponds with the lowest average waiting time at terminals. Based on our results we cannot claim that this holds in general, but if it is true, then terminals might have an incentive to choose that amount of slack that minimizes the average waiting time at the terminal. This in turn is in the interest of barges. In Figure 5.13 we show which amount of slack leads to the lowest average waiting time per terminal for different utilization degrees of the terminals. The results were obtained from scenarios with network layout II, 9 terminals per region where terminals have one or two quays.

Considering the average waiting time that successive barges face at terminals, we see great fluctuations if the utilization degree of terminals increases. This

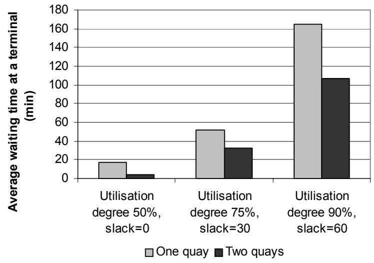


Figure 5.13: The amount of slack that leads to the lowest average waiting time per terminal, specified for different terminal utilization degrees. Results are presented for scenarios with network layout II, 9 terminals per region where terminals have one or two quays

is to be expected considering theory about the relation between utilization degree, the lead time, and the work-in-progress level of a system (Hopp and Spearman, 2000). If the utilization degree of the system increases, the lead time and/or the work-in-progress level will grow as well. Combined with the variability in the system, the system becomes more vulnerable, which results in fluctuations in both lead times and work-in-progress levels. This means that the total amount of waiting time as part of the total sojourn time in the port increases when the utilization degree in the network grows. This is of importance in practice when barge operators set time windows for barges.

Relation between processing time and waiting time

In practice nowadays, barges with small call sizes have a higher chance of being handled at a terminal in the short run, than barges that have large call sizes. The reason why is that the former barges can more easily be squeezed in the schedule of the terminal. Barges with large call sizes see this as a major disadvantage.

Also in our simulations we find a relationship between the call size (the processing time) and the waiting time at the terminal. In Figure 5.14 we approximate this relationship for the same scenario as used in Figures 5.7 and 5.8 in case we use waiting profiles with a slack of 30 minutes. Note that processing times are drawn from a normal distribution (see Section 4.6.2), which means that the results for small and large processing times are based on a limited number of observations. This explains the fluctuations in the graph. The depicted relationship between the processing time and the average waiting time at a terminal is only found for higher utilization degrees (>75%). For lower utilization degrees the relationship is not significant. In Figure 5.15 we combine the insights obtained from Figures 5.10 and 5.14, by depicting the relation between the average processing time at a terminal and the position of the terminal in a

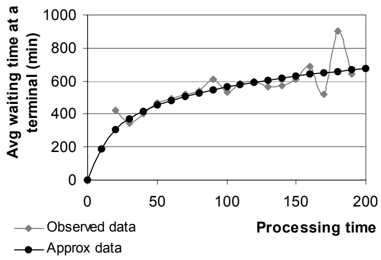


Figure 5.14: Relation between processing time and waiting time. Results for a 90% utilization degree

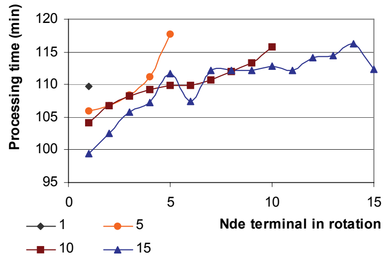


Figure 5.15: Processing time related to the sequence in which the terminals are visited, depicted for different rotation lengths

rotation.

We remark that the average waiting time at a terminal is not only determined by the processing time at the terminal concerned, but also on the number of terminals a barge has to visit in its rotation as we have explained based on Figures 5.9 and 5.10.

5.7 Conclusions

In this chapter we evaluated the performance of our Multi-Agent system. We modeled the barge and the terminal operator and described in detail how waiting profiles can be constructed. We evaluated our Multi-Agent system by means of simulation and compared the results with alternative levels of information exchange and the off-line benchmark. We analyzed the results and provided several insights in the functioning of a distributed control in our problem.

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

Chapter 6

Service-time profiles

6.1 Introduction

In this chapter[1] we make some practical extensions to the model we developed in Chapter 5. These extensions concern restricted opening times of terminals, unbalanced networks, sea vessels, and closing times of containers. To deal with these realistic issues, we introduce the concept of *service-time profiles*, which enables us to deal with time-dependent service-times. This allows barges to plan a rotation along terminals, taking into account the availability of terminals and the effect on the service time. We perform several simulations to analyze the effect of different factors on the performance of barges and terminals, such as restricted opening times and unbalanced networks. We compare the results with the off-line benchmark. The model in this chapter forms a basis for implementation in practice. Let us describe the contribution of this chapter, our assumptions, and the outline of the chapter.

6.1.1 Contribution

In this chapter we extend the model proposed in Chapter 5 by introducing the following practical extensions, which make the model more realistic:

- *Restricted opening times of terminals.* Terminals can be closed during a certain period of the day.
- *Unbalanced networks.* Terminals can have a different capacity and utilization degree, and in a port we can have busy and quiet regions.
- *Sea vessels.* Terminals process, besides barges, also sea vessels.

[1] This chapter is based on the paper: A.M. Douma, P.C. Schuur, and J.M.J. Schutten (2008). Using service-time profiles for distributed planning of container barge rotations along terminals. Beta-working paper WP-247. Submitted for publication.

- *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.

The aim of this chapter is i) to develop and extend our Multi-Agent system to be able to deal with these more complicated settings and ii) to get insight in the way the system functions. We extend the model proposed in Chapter 5 by introducing service-time profiles, which are an extension of the waiting profiles with respect to time dependent service times. We consider different experimental settings to evaluate (by means of simulation) the effect of increasing degrees of complexity on the performance of the Multi-Agent based controls and to compare the results (obtained by simulation) with an off-line benchmark, which is a central algorithm based on the assumption that information over the whole planning horizon is known a priori. This information concerns the arrival times of barges and sea vessels, the terminals each barge or sea vessel has to visit, the processing time of each barge and sea vessel, the terminals and their capacity (including opening times), the closing times of containers, and the sailing times in the network. In the off-line benchmark this information is known prior to the related planning horizon, whereas in the Multi-Agent based control, this information is revealed gradually during the planning horizon (except for opening times of terminals, and the arrival and processing times of sea vessels).

6.1.2 Assumptions

We make the following assumptions. Barges arrive over time with independent stochastic interarrival times. On arrival in the port the barge operator decides which rotation a barge is going to execute. Decisions of both barges and terminals have to be made in real-time and we assume that two barge operators do not plan rotations simultaneously, but one after another. With respect to a terminal, we assume that it handles both barges and sea vessels. With respect to barges, terminals only have information about barges that arrived in the port. The service of a barge is not preempted for the service of another barge. Terminals have fixed capacity and can have restricted opening times, during which barges can be processed. Opening times of terminals are hard, i.e., no work is done in overtime. The time to handle a container, the mooring time, and the sailing time between terminals are deterministic. Sea vessels arrive with stochastic interarrival times at terminals and processing takes a stochastic amount of time. Arrival times and handling times of sea vessels are known to the terminal prior to the planning horizon. Sea vessels have absolute priority over barges.

With respect to a barge we assume that, on arrival in the port, it needs to make a decision about the sequence in which it visits the terminals concerned. On arrival in the port, the barge has information about the terminals it has to visit, the number of container it has to (un)load at each terminal, and the mooring time at and the sailing time between terminals. It has no information about the state of the network, such as waiting times at terminals. We consider no capacity or stowage constraints for the barge. Barges visit terminals only once. We define closing of a terminal as *preemptive downtime*, and the processing of a sea vessel as *non-preemptive downtime*. Preemptive downtime means that the handling of a barge may start before the downtime and finish after it (Schutten, 1998). Non-preemptive downtime means that the handling of a barge might not be interrupted by a downtime. We do not consider disturbances and their effect on the operations of barges and terminals. These assumptions allow barges and terminal operators to make reliable appointments, since no unexpected delays occur. Introducing disturbances requires the introduction of a kind of penalty, e.g., penalty costs, if appointments are not met by one of the parties. This is part of future research.

6.1.3 Outline of the chapter

The outline of the chapter is as follows. In Section 6.2 we introduce the practical extensions we study in this chapter. In Section 6.3 we argue why waiting profiles need to be extended to service-time profiles. Sections 6.4 and 6.5 describe the barge operator agent and the terminal operator agent, respectively. Section 6.6 describes the experimental settings and Section 6.7 the results of the simulations and the off-line benchmark. Finally, we draw some conclusion in Section 6.8.

6.2 Practical extensions

In addition to Chapter 5 we make our experimental settings incrementally complex to study the effects of a specific factors on the performance of barges. We consider the following practical extensions successively: restricted opening times of terminals, unbalanced networks, sea vessels, and closing times of containers.

Restricted opening times of terminals mean that terminals are only opened during a certain time window every day or week. In the Port of Rotterdam a lot of terminals are closed during the night, especially the smaller terminals near the city. This means that the processing of a barge can be interrupted by closing of the terminal, which impacts the sojourn time at the terminal significantly. We assume that opening times of terminals are hard, i.e., even when a barge only needs to load or unload one container at the moment the terminal closes, it has to wait until the terminal opens again. This is not realistic for all terminals in the port, but in general it is true that workers at the terminal hardly work overtime. In our model we consider closing of a terminal as preemptive downtime (see Section 6.1.2).

With unbalanced networks we mean that i) terminals are not identical, but have a different capacity and utilization degree and ii) can be distributed unequally over the port, resulting in busy and quiet regions. This is a realistic situation. In the Port of Rotterdam, e.g., about 80% of all container transshipment takes place in one region.

Sea vessels are handled in the Port of Rotterdam only at a few terminals that are able to receive these large vessels. For all these terminals it holds that sea vessels have absolute priority over barges, which means that the handling of a barge is stopped at the time a sea vessel arrives. We consider sea vessels as non-preemptive downtime (see Section 6.1.2). This choice is in accordance with the current practice in the port, although some terminals sometimes process barges in parallel. Sea vessels have long processing times compared to barges. Sea vessels therefore significantly impact the capacity available for barges. To determine the interarrival and processing time distributions of sea vessels, we analyzed data of sea vessels that arrived from 2005 until 2007 at the major terminals in the Port of Rotterdam. See also Kuo et al. (2006) for a case study into arrival patterns of container vessels in international ports in Taiwan.

Closing times of containers represent specific times at which containers need to be at a terminal. The main reason for a closing time is the fact that the container has to be shipped with an ocean going vessel, which departs at a specific time (Sinclair and Van Dyk, 1987). We therefore assume that only those containers that have to be transshipped at a terminal where also sea vessels are processed, can have a closing time.

6.3 From waiting profiles to service-time profiles

In Chapter 5 a waiting profile, issued by a terminal for a particular barge, denotes the maximum waiting time for a barge until the start of processing, for every possible arrival time during a certain time horizon. The basic assumption in generating waiting profiles is that the duration of the processing of a barge at the terminal is independent of the time at which the handling is started. This assumption holds for situations where terminals are never closed and only deal with barges. However, if the processing of a barge is interrupted by, e.g., the closing of the terminal, the processing time increases with the time the terminal is closed, since closing of the terminal is considered as preemptive downtime. This means that we are now confronted with *time dependent service-times*.

If the service time of a barge is time dependent, it is clear that waiting profiles alone do not offer enough information to barge operators. To be able to optimize a rotation, additional information about the service time needs to be revealed by the terminal. To keep the information exchange between terminals

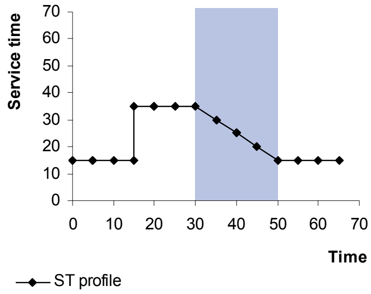


Figure 6.1: Example of a service-time profile for an incoming barge that needs 15 units of processing time. The terminals has planned no activity and is closed from t=30 to t=50

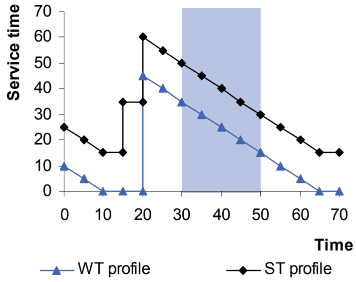


Figure 6.2: Example of a service-time profile for an incoming barge that needs 15 units of processing time. The terminals has planned already two barges (from t=0 to t=10 and from t=55 to t=65) and is closed from t=30 to t=50

and barges as simple as possible, we propose to integrate both processing and waiting times in one profile, called a *service-time profile*. A service-time profile represents the guaranteed maximum time necessary to serve the barge, given a certain start time. A service-time profile or service-time function is also introduced by (Christiansen and Fagerholt, 2002) and 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 more complex. We illustrate this with two examples.

In Figure 6.1 we give an example of a service-time profile for an incoming barge that needs 15 minutes of processing time. The terminal has planned no other activity and is closed from $t = 30$ to $t = 50$. In Figure 6.2 we give an example, which is a bit more complex. It shows a service-time profile for an incoming barge that needs 15 minutes of processing time. The terminal has scheduled already two other barges, from $t = 0$ to $t = 10$ and from $t = 55$ to $t = 65$, and the terminal is closed from $t = 30$ to $t = 50$. Note that the service of the incoming barge cannot be preempted for the service of a yet scheduled barge. That is why the waiting time increases at $t = 20$.

6.4 The barge operator agent

In our model, the barge operator decides in which sequence a barge is going to visit the terminals. The algorithms introduced in Chapter 5 have to be adapted slightly as a result of the change from waiting to service-time profiles and the introduction of closing times on containers. The objective of the barge operator agent in our experiments is a combination of activity tardiness and project tardiness. If we omit closing times on containers in our experiments (then the total activity tardiness becomes zero), the primary objective of the barge is solely to minimize the project tardiness, i.e., the barge tries to leave the port as soon as possible.

6.4.1 Model and notation

The barge operator agent has to make two decisions, namely i) in which sequence the barge visits the terminals and ii) the specific time each terminal is visited. We assume that every barge operator agent makes these decisions at the moment the concerning barge enters the port. We assume that the information of the terminals is reliable and does not change during the time the barge operator agent needs to make its decisions. It is important that barges make decisions in real time.

Let us consider one specific barge entering the port. The barge operator agent of this barge is assumed to have the following information. First, it knows the set $\mathcal{N}$ of terminals that have to be visited. This set is a subset of all the terminals in the port. We assume that the agent knows the maximum service time θ_{jt}, for every terminal $j \in \mathcal{N}$ and every arrival moment t. The maximum service time can be derived from the service-time profile and consists of waiting and handling time at the terminal. The handling time h_j consist of the time needed for loading and unloading of containers, as well as the mooring time on arrival and departure. In addition, the agent knows the sailing times $s_{j_1 j_2}$ between every pair of terminals (j_1, j_2), where $j_1, j_2 \in \mathcal{N}$. We assume that both the sailing and the handling time are deterministic. Let w be the planned port departure time set in the sailing schedule and c_j the earliest closing time of all containers that have to be (un)loaded at terminal j. Without loss of generality we assume $c_j \leq w$. Every barge enters and leaves the port via a *port entrance and exit point.*

In case containers have a closing time, then the barge operator has to weigh 'sticking to the sailing schedule' and 'meeting the closing times of containers', i.e., it has to weigh *project* tardiness against *activity* tardiness. The extent to which a barge operator agent can realize this objective depends both on the actual state of the port and the information it has about this. Optimizing its rotation, the barge operator assumes that the service time at the terminal will be equal to the time expressed in the service-time profile (which is actually a *maximum* service time). We can now model the barge operator agent by a *time dependent traveling salesman problem* (TDTSP). We define the time dependent travel time $\tau_{j_1 j_2}(d_{j_1}) = s_{j_1 j_2} + \theta_{j_2, a_{j_2}(d_{j_1})}$, as the sum of the sailing and service time going from terminal j_1 to j_2, where the barge departs at time d_{j_1} at terminal j_1. The aim of the barge is to find a rotation along all terminals it has to visit, such that the sum of the time dependent travel times is minimized. We assume that a barge visits a terminal only once.

6.4.2 Algorithms to solve the problem

To solve the TDTSP 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 (see also Chapter 5). We adapt the objectives of both algorithms to be able to deal with activity and project tardiness. The change in objectives also requires some changes in the DP-heuristic. We discuss the two topics in Section 6.4.2 and Section 6.4.2 respectively.

Objective

To deal with closing times of containers we have to adapt the objective function of the algorithms we use. Let ω_j^A denote the activity lateness at terminal j. If one or more containers at terminal j has a closing time, then $\omega_j^A = d_j - c_j$ and is zero otherwise. 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 w. 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

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

So, we consider all possible scenarios and select the rotation that minimizes the weighed 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.

Selection criterion in the DP-heuristic

To deal with container closing times, we have to adjust the DP-heuristic of Malandraki and Dial (1996), introduced in Section 5.2.2, 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 Section 6.4.2 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 different selection criteria 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 container that has to be transshipped at terminal j has a closing time, and is zero otherwise. The selection criterion we use is then given by:

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

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.

6.5 The terminal operator agent

In our model, the terminal operator decides about convenient times to process barges. In Chapter 5 we argue that a terminal has two tasks, namely, making appointments and keeping appointments. However, the introduction of restricted opening times and sea vessels has a major impact on the way the terminal performs these tasks. In this section we briefly introduce the mathematical model and discuss adaptations in the algorithms used to plan the terminal quays and the construction of the service-time profiles.

6.5.1 Model and notation

Every terminal in the port has a terminal operator agent which has to negotiate with barges about convenient handling times. Every terminal has a set of, so-called, quays Q. A quay is a combination of resources that are all necessary to handle a barge, namely a berth, crane(s), and a team. A barge is assigned to one quay $q \in Q$ and every $q \in Q$ processes at most one barge at a time. Sea vessels, however, can be assigned to more than one quay. If a sea vessel needs more than one quay, the berths of the respective quays need to be adjacent. Without loss of generality, we assume that all quays are positioned in a line and are numbered in ascending order, starting from zero. This means that quay q_0 is only adjacent to q_1, and quay $q_{|Q|}$ is only adjacent to $q_{|Q|-1}$. In general quay q_i is neighbored by q_{i-1} and q_{i+1}, where $0 < i < |Q|$. A terminal can be closed during certain periods of the day, which means that all quays $q \in Q$ are closed.

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

An appointment is an agreement between the barge and the terminal operator (see Section 3.5). By making an appointment, the barge promises the terminal to be present before a certain time, namely the *latest arrival time (LAT)*. Based on the service-time profile, issued in the first phase of the barge's decision process, the terminal derives the *maximum service time (MST)* for the LAT and guarantees the barge a *latest departure time (LDT)*. It holds that LDT=LAT+MST. After the appointment is made, the terminal schedules the barge tentatively such that the appointment with the barge is not violated, nor the appointments made with other barges. In the schedule (which is hidden for the barge) every barge has a *planned starting time (PST)* and an *expected departure time (EDT)*. The EDT is calculated based on the planned starting time, the processing time (PT), and the opening times of the terminal. The PT is equal to the handling time (h_j) of the barge, as introduced in Section 6.4.1 and is revealed by the barge in the first phase of its decision process.

	Agreements with the barges		*Schedule (hidden for the barges)*		
Barge	Latest arrival time (LAT)	Latest departure time (LDT)	Planned starting time (PST)	Processing time (PT)	Expected departure time (EDT)
B1	5	25	5	15	20
B2	55	75	50	10	60

Table 6.1: Example of a quay schedule and the corresponding agreements

To give an illustration, consider the agreements made with barge B1 in Table 6.1. Barge B1 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). The schedule of the terminal is hidden for the barge and denotes the PST, PT and EDT, respectively 5, 15 and 20 for this barge. Barge B2 has made similar appointments as shown in Table 6.1. In the example, the slack in the appointments is 5 to 10 minutes.

6.5.2 Keeping appointments

The keeping appointments task is the scheduling of *expected barges* on the quays such that no appointment is violated. Expected barges are barges that already made an appointment with the terminal and still need to be processed (see Table 6.1 for an example). From time to time, it might be beneficial for a terminal to reschedule barges on the quays, e.g., when a barge arrives earlier than its latest arrival time. This is possible when a barge does not need to wait the maximum waiting time at previous terminal(s).

To schedule the expected barges, the terminal has to decide how the barges are assigned to the available quays, such that all appointments are met, i.e., the

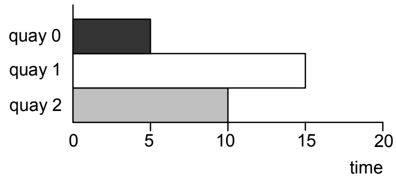


Figure 6.3: Example of a quay schedule of a terminal. The bars represent the scheduled time for sea vessels at each quay

handling of no barge is finished later than its LDT. Finishing the handling of a barge later than its LDT is not permitted in our model. By scheduling the barges, the terminal has to take into account the *sea vessels* that arrive in the near future. For every sea vessel the terminal knows the number of quays it needs and the amount of containers it needs to load and unload. We assume that sea vessels are known a priori to the planning horizon, or at least so early that no conflict arises with yet scheduled barges. We assume that sea vessels are scheduled in the order of arrival.

We schedule sea vessels in the following way. Let τ_j be the departure time of the last scheduled sea vessel on quay q_j, k the number of quays required for the sea vessel that is to be scheduled, and t^a the arrival time of the sea vessel. A sea vessel is then assigned to a group of quays $q_i, ..., q_{i+k-1}$ for which the most time has elapsed after the last scheduled sea vessel, i.e., $\max_{i=0,...,|Q|-k} \min_{j=i,...,i+k-1} [t^a - \tau_j]$. Suppose we have to schedule a sea vessel arriving at time $t^a = 20$, requiring two quays, and that we have already a schedule as given in Figure 6.3. The sea vessel is now scheduled on quays q_0 and q_1.

The primary objective for the terminal for the scheduling of 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 of a barge as the expected departure time of this barge minus its latest departure time. Remark that the terminal lateness always needs to be less than or equal to zero, since all appointments with barges need to be met. The algorithm we apply to obtain a schedule is list-scheduling in combination with depth-first branch-and-bound (see Schutten, 1996). The algorithm is adapted to deal with sea vessels and restricted opening times, since these events impact the start and service time of the barge. The basic idea of the list scheduling algorithm is to make an ordered list of barges and to assign these barges successively to the first available resource. The time the resource (quay-crane-team combinations q) is available is determined iteratively starting from the completion time t^c_q of the last scheduled barge. We use Algorithm 6.1, where h denotes the processing time of the barge that is currently scheduled. A good initial solution to the branch-and-bound algorithm is usually to sort barges based on latest departure time (LDT), which results in a strong bound and makes the algorithm fast.

```
ok ← false
t ← t_q^c

while not ok do
  h' ← h

  //First check the start and processing time with respect to closing of the
  terminal
  if t is during closing of the terminal then
    increase t to the time the terminal is open again
  else if t + h' overlaps with closing of the terminal then
    increase h' with the time the terminal is closed

  //Then check the start and processing time with respect to planned sea
  vessels
  if t + h' overlaps with the processing of a sea vessel then
    increase t to the time the processing of the sea vessel is completed

  //Is the start time ok with respect to the closing of the terminal and
  planned sea vessels?
  if t = t_q^c then
    ok ← true
  else
    t_q^c ← t
```

Algorithm 6.1: Calculating the first possible start time of an activity on a resource

6.5.3 Making appointments: how to construct the service-time profile

To construct a service-time profile we need to determine i) the possible start times of the barge and ii) the service-time given a certain start time. For the former we use the concept of waiting profiles as introduced in Chapter 5, and for the latter we create a service-time function. The waiting profile and the service-time function are then merged into a service-time profile. We explain the construction of the waiting profile, the service-time function, and the service-time profile in this section.

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 have to extend the procedure in Chapter 5 to construct a waiting profile that is compatible with sea vessels and restricted opening times of terminals. We proceed as follows. First we determine 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 such that no appointment with other barges or sea vessels is violated when the processing of the barge is completed. Based on the start intervals we create a quay waiting profile analogous to the procedure in Chapter 5, and based on all the quay waiting profiles we construct a terminal waiting profile. It is important that in this stage no slack is added to the waiting profile. We describe the way the start intervals are determined in more detail.

To determine all the start intervals we take two steps. The first step is similar to the start interval procedure described in Chapter 5. For all insertion points i we determine a start interval in the following way. We schedule all barges and sea vessels before insertion point i as early as possible, and all barges and sea vessels after insertion point i as late as possible. By doing so we take into account the sequence in which the ships are scheduled as well as the restricted opening times of the terminal. Let m_i and n_i be the respective start 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 $PST_{i+1} - PT$, where PST_{i+1} denotes the planned (and latest possible) starting time of the first scheduled ship after insertion point i. If $n_i < m_i$ then the start interval is empty and we do not consider it anymore. If $m_i \leq n_i$ then we maintain the start interval.

Once we have considered all possible insertion points i in the quay schedule, 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 $\left[m_i^{'}, n_i^{'}\right] \subset [m_i, n_i]$. If $n_i^{'} < m_i^{'}$ then the adjusted start interval is empty and we do not consider it anymore. Otherwise we maintain the start interval.

In Table 6.2 we give six possible elementary relations between a start interval and closing of the terminal. In Table 6.2 we denote the interval in which the terminal is closed with $[e^s, e^c]$. Note that $PST_{i+1} \notin [e^s, e^c]$. Let us consider the fifth situation in Table 6.2. We see that in this situation $m_i < e^s < n_i \leq e^c$. Assuming that we derived the start interval $[m_i, n_i]$ in the way we describe above, then $n_i = PST_{i+1} - PT$. However, if we start barge b in interval $[e^s, n_i]$ the completion time of barge b will be greater than PST_{i+1}. This means that we have to decrease n_i such that $n_i^{'} + PT + e^c - e^s \leq PST_{i+1}$. Rewriting the expression gives $n_i^{'} = n_i + e^s - e^c$. The example is just one illustration of

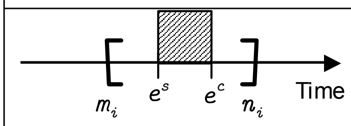
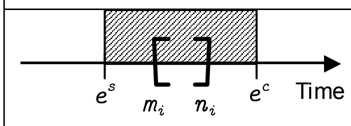
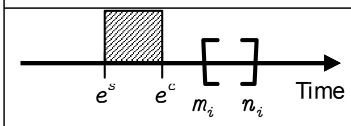
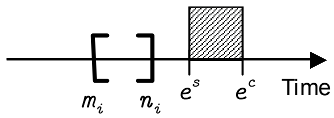
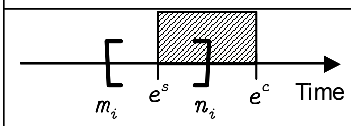
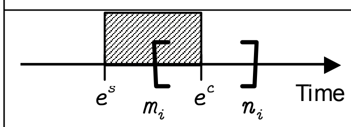
Possible situation	**Condition**	**New start interval**
	$m_i \leq e^s < e^c < n_i$	$m_i^{'} = m_i$ and $n_i^{'} = n_i$
	$e^s \leq m_i < n_i \leq e^c$	infeasible interval
	$e^s < e^c \leq m_i < n_i$	$m_i^{'} = m_i$ and $n_i^{'} = n_i$
	$m_i < n_i \leq e^s < e^c$	If $e^s < n_i + PT$ then $m_i^{'} = m_i$ and $n_i^{'} = n_i + e^s - e^c$. If $n_i + PT < e^s$ then $m_i^{'} = m_i$ and $n_i^{'} = n_i$.
	$m_i < e^s < n_i \leq e^c$	$m_i^{'} = m_i$ and $n_i^{'} = n_i + e^s - e^c$
	$e^s \leq m_i < e^c < n_i$	$m_i^{'} = e^c$ and $n_i^{'} = n_i$

Table 6.2: Six possible elementary relations between start intervals and closing of the terminal

how a start interval has to be adapted. In practice all other kind of (nested) situations can exist.

To give an illustration, we use the following example. Assume that we have a schedule as given in Table 6.1 and that we have to make a service-time profile for barge b with processing time $PT = 15$. In this example the terminal is closed from $t = 30$ until $t = 50$. After the first step in our procedure we find the start intervals as given by Table 6.3. A start interval i corresponds with insertion point i.

Start interval	Start time	End time	Insertion point
1	20	50	1
2	65	∞	2

Table 6.3: The start intervals in the example

In the example it is clear that start interval one overlaps with closing of the terminal, 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 terminal in our example has one quay and the resulting terminal waiting profile is depicted in Figure 6.4.

Construction of the service-time function

The construction of the service-time function is only depending on the preemptive events, which in our model is only the closing of the terminal. Since most terminals close in repetitive patterns, the service-time function can easily be constructed based on the opening times of the terminal. We apply the following procedure to construct the service-time function. Suppose the closing period of the terminal is repeated every T time units, called the *cycle length*, e.g., a day. Then for a sufficiently long horizon (nT, $n \in \mathbb{N}$) we add the following points to the service-time function $s(t)$. For opening of the terminal at time t^O we add $s\left(t^O\right) = PT$, with slope 0. Closing of the terminal results in two points, namely closing of the terminal at time t^C with corresponding $s\left(t^C\right) = PT + t^O - t^C$ and slope -1, and a point $t^C - PT$ with corresponding $s\left(t^C - PT\right) = s\left(t^C\right)$ and slope 0.

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 $[t_0, nT]$ the corresponding maximum service time. The maximum service time $\widetilde{S}(t)$ at time t can be calculated by $\widetilde{S}(t) = w(t) + s(t + w(t)) + s$, 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 reason this function holds, is that at time $t + w(t)$ a barge can be started by definition, since $w(t)$ is the time till opening of the next start interval, and a start interval has the property that the barge handling be started without violating any other appointments. In addition to the maximum service time we also have to determine the slope of the function. The slope of the function is -1 if $w(t) > 0$, and is equal to the slope of the service-time function otherwise.

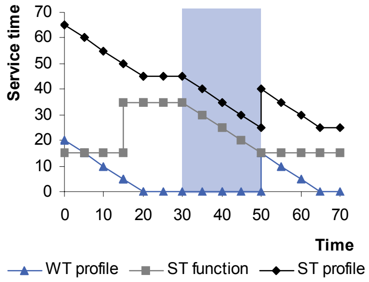


Figure 6.4: Example of a waiting (WT) profile, service-time (ST) function, and the ST profile

Time	Max service time	Slope
0	65	-1
15	50	-1
20	45	0
30	45	-1
50	40	-1
65	25	0

Table 6.4: Data representation of the service-time profile of Figure 6.4

The service-time profile of the example presented at the start of this section is given in Figure 6.4. The corresponding data representation is presented in Table 6.4. In the service-time profile we included 10 time units slack.

6.6 Experimental settings

To evaluate the Multi-Agent System for different levels of information exchange, we consider different scenarios and compare the results with the off-line scheduling algorithm. To obtain insight in the functioning of the Multi-Agent System, we consider also other port configurations besides the Port of Rotterdam. In this section, we describe the scenarios, the variables, and the parameters. Recall that all handling and sailing times are deterministic. As time unit we use minutes in our experiments.

6.6.1 Introduction

We consider different experimental settings, namely data set 1 to 4 (see Section 4.6), to evaluate the performance of the different levels of information exchange. Each data set is divided in different scenarios. We consider the effect of restricted opening times of terminals and unbalanced networks in separate experimental settings. We do not consider the effect of introducing sea vessels separately, since we expect that the effect will be similar to introducing restricted opening times. Instead, we consider an experimental setting with restricted opening times of terminals, unbalanced networks, sea vessels, and closing times of containers. In future research one might consider also other experimental settings, such as the effect of introducing sea vessels.

The primary objective of the off-line benchmark is minimizing the sum of the project tardiness and the activity tardiness, and the secondary objective to minimize the fraction of late barges. Note that if containers have no closing time, then the activity tardiness will be zero. We then weigh the project tardiness with a factor $\gamma = 1$. If containers do have a closing time (experimental setting 3), then we use different values for γ, as we describe in Section 6.6.4.

In the remaining sections we discuss the experimental settings in more detail. Experimental setting n corresponds with data set $n + 1$. All the data sets are based on the basic data set. A description of this data set can be found in Section 4.6.2.

6.6.2 Setting 1: Restricted opening times of terminals

For Experimental Setting 1 (restricted opening times of terminals) we generate 36 scenarios, varying along the following dimensions: the number of terminals per region (either four or nine terminals), three different network layouts (see Figure 4.10), three different utilization degrees of the network (50, 75, and 90%), and variable and fixed time windows for the barges. In all experimental settings, half of the terminals in a region (rounded down to the nearest integer) are closed during the night (6 p.m. - 6 a.m.) and the rest is opened 24h a day. To realize the desired utilization degree, the number of barges is reduced for the terminals that close during the night. The utilization degree of a terminal is calculated based only on the time the terminal is open. 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. The call size at a terminal is drawn from a normal distribution with mean 30 containers and a standard deviation of 10 containers. We evaluate all the scenarios for all levels of information exchange (Section 3.6). For every scenario, we performed 10 replications 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. The off-line benchmark is based on 28 days and we apply the same warm-up and cool-down period.

In addition, we perform a sensitivity analysis to analyze the impact of i) the distribution of terminals that are closed during the night over the different regions and ii) the time the terminals are closed. Aim of the experiment is to get insight in the impact of closing times on the performance of barges and terminals. With respect to the way terminals with restricted opening times are distributed over the regions, we consider two network configurations. In configuration A we close 50% of the terminals in every region (A, B, C). In configuration B we close 100% of the terminals in region A, 50% of the terminal in region B, and 0% of the terminals in region C.

6.6.3 Setting 2: Unbalanced networks

For Experimental Setting 2 (unbalanced networks) we generate six scenarios, varying along the following dimensions: three different network variants (see Figure 4.10), and a variable and fixed time window for the barge. Terminals have a different number of quays, different utilization degrees, and different processing times. We distinguish four terminal types (see Table 6.5).

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 6.5: 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

The call sizes are drawn from a normal distribution. We simulate different network layouts with specific configurations of terminal types, as given by Table 6.6. 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.

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

Table 6.6: The number of different terminal types per network layout

Terminals are opened 24h a day in all experiments. 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. We apply a warm-up period of 10 days and a cool-down period of 3 days.

To gain additional insight we also simulate the scenarios using service-time profiles with varying amounts of slack per terminal. We consider three options (1, 2, 3) and vary the slack depending on the type of terminal as denoted in Table 6.7. The choice for the slack values is based on previous experiments (see Section 5.6.1).

	Slack option		
Terminal type	1	2	3
Alpha	0	0	0
Beta	0	0	0
Gamma	30	30	30
Delta	30	60	90

Table 6.7: The amount of slack for different terminal types

Aim of the experiments is to get insight in the effect of an unbalanced network setting on the performance of barges and terminals and the performance of different controls in different settings.

6.6.4 Setting 3: Unbalanced networks, restricted opening times terminals, sea vessels and closing times containers

Experimental Setting 3 is an extension of Experimental Setting 2, with respect to the introduction of restricted opening times, sea vessels, and closing times of containers. The network configurations are similar to Table 6.6. However, the terminal types have some additional properties. The alpha terminals are closed from 6pm - 6am and do not handle sea vessels. The other terminal types are opened 24h 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 container sea vessels that visited the Port of Rotterdam at the major container 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 minutes. Based on these settings, we can 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. We assume that sea vessels arrive at a terminal with exponentially distributed interarrival times. 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). The slack factor used for the fixed time window is 2. The fixed slack y (slack per rotation) for the variable time window is equal to 1.0 and the variable slack x (slack per terminal) equal to 0.12.

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. Considering the experimental results in previous experiments we have chosen the following settings. First, we use network layout II, since the results of the other network layouts generally follow similar patterns. Every region has nine terminals. We consider all levels of information exchange considered in Experimental Setting 1, plus 'Slack Option 2' (see Table 6.7) introduced in Experimental Setting 2. 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 3 days. The off-line benchmark is based on 10 replications of 15 days and we reduced the warm-up and cool-down period to 4 respective 3 days. The reason is that in this experimental setting a lot of activities are performed every day and the system needs less time to get in steady state. Moreover, the remaining number of days is still acceptable for our analysis.

6.7 Results

In this section, we present the results and insight obtained after simulating all the experimental settings described in Section 6.6. We describe the results of Experimental Settings 1, 2, and 3 respectively.

6.7.1 Setting 1 - Restricted opening times of terminals

The performance of different levels of information exchange in case terminals have restricted opening times is comparable to the results presented in Chapter 5. We therefore depict only the comparison with the off-line benchmark in terms of average tardiness and average lateness (see Figure 6.5 and 6.6).

If we look at the way barges deal with restricted opening times for different levels of information exchange, we see 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. The reason is that barges limit the risk to stay overnight at

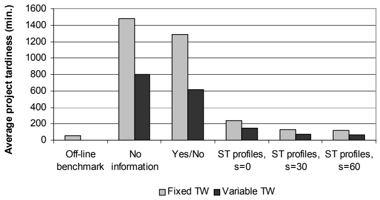


Figure 6.5: The average project tardiness per barge averaged over all considered experiments, compared with the off-line benchmark

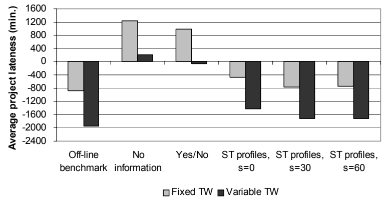


Figure 6.6: The average project lateness per barge averaged over all considered experiments, compared with the off-line benchmark

a terminal which would lead to a significant longer sojourn time in the port. In Figure 6.7 and 6.8, we depict the average fraction of nights barges were present at a terminal (with restricted opening times), averaged over all terminals and all scenarios, and specified for a utilization degree of 50 and 90%. Additionally, we denoted the fraction of times a terminal started processing a barge at opening of the terminal. It is clear that with service-time profiles communication barges arrive during the night. This means in practice that terminal with restricted opening times will see a reduced occupation of their quays at the end of the day.

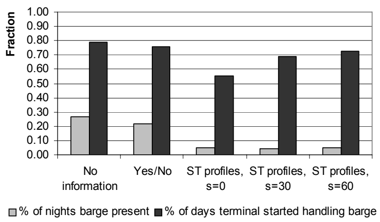


Figure 6.7: Fraction of barges during the night present at the terminal and fraction of barges for which the processing has been started at the start of the day. Results for different communication protocols, averaged over all scenarios with a 50% utilization degree

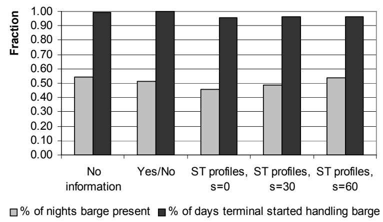


Figure 6.8: Fraction of barges during the night present at the terminal and fraction of barges for which the processing has been started at the start of the day. Results for different communication protocols, averaged over all scenarios with a 90% utilization degree

The sensitivity analysis reveals that an increase of the duration of the closing of the terminals leads to an increase of the average sojourn time of barges in the port. The extent to which the sojourn time increases depends on the way the terminals (with restricted opening times) are distributed over the different regions, and on the utilization degree of the network. We depicted the results in Figures 6.9 to 6.12.

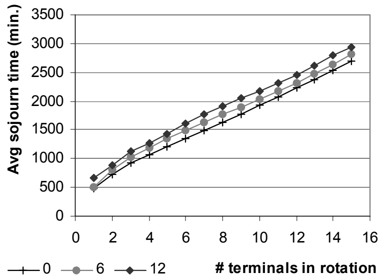


Figure 6.9: Average sojourn time of barges in a network with 50% utilization degree and network configuration A. Results denoted for varying closing times of terminals (in hours)

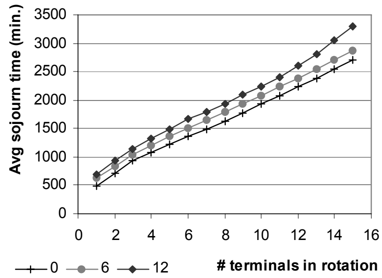


Figure 6.10: Average sojourn time of barges in a network with 50% utilization degree and network configuration B. Results denoted for varying closing times of terminals (in hours)

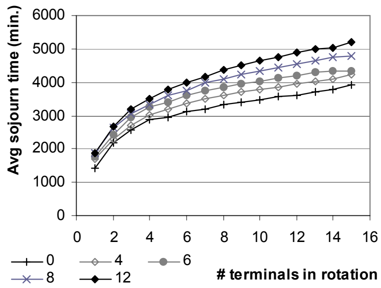


Figure 6.11: Average sojourn time of barges in a network with 90% utilization degree and network configuration A. Results denoted for varying closing times of terminals (in hours)

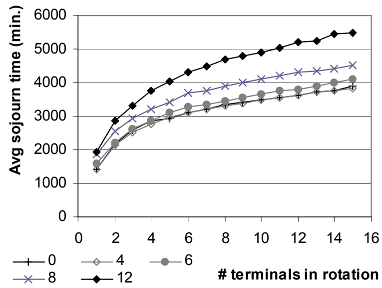


Figure 6.12: Average sojourn time of barges in a network with 90% utilization degree and network configuration B. Results denoted for varying closing times of terminals (in hours)

If terminals (with restricted opening times) are equally distributed over the regions (configuration A) or the network utilization is low (say 50%), then the sojourn time increases linearly in the duration of the closing. If terminals are divided unequally over the regions (configuration B) and the network utilization is high (say 90%), then we see that, the longer the terminal closes, the greater the impact is on the sojourn of barges. The reason for the increase in sojourn time is probably an increase in variability in the system. From the factors determining the cycle time of a product, namely the variability, the utilization degree and the processing time (Hopp and Spearman, 2000), only the variability is affected.

6.7.2 Setting 2 - Unbalanced networks

We now look at the results for Experimental Setting 2, regarding the unbalanced networks. In Figures 6.13 and 6.14 we depicted the performance of the different communication protocols on the simulated scenarios. We omit the comparison with the off-line benchmark, since we could only calculate a limited time horizon (about eight days) with the off-line benchmark which is not realistic anymore if we also have to apply a warm-up and cool-down period. From the results it is clear that waiting profiles communication outperforms the other levels of information exchange.

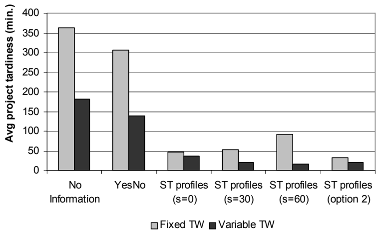


Figure 6.13: Average project tardiness per barges, averaged over all scenarios and specified for fixed and variable time windows

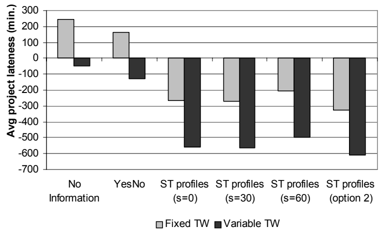


Figure 6.14: Average project lateness per barges, averaged over all scenarios and specified for fixed and variable time windows

In addition to the regular amount of slack added to the service-time profiles (0, 30, and 60 minutes), we also considered options in which terminals add a different amount of slack (see Table 6.7). In Figures 6.13 and 6.14 we depicted also the results of *slack option 2,* which performs not always better in terms of project tardiness, but does lead to lower project lateness. However, the performance of all three options of Table 6.7 are very similar and (in case of fixed time windows) clearly better than when we let every terminal add the same amount of slack to their service-time profile. We observe that the average waiting time at a terminal can decrease significantly by introducing a certain amount of slack. Note that waiting time at a terminal is not depending on fixed or variable time windows, since barge do not take these time windows into account in their decision, except for the yes/no information exchange. In Table 6.8 we show an example of the average waiting time per terminal type for the alternative amounts of slack in a line network layout. The 0, 30, and 60 minutes slack mean that all terminals add this amount of slack to the service-time profile. Slack options 1, 2, and 3 are described in Table 6.7.

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.

	slack in min.			slack option			
Terminal type	0	30	60	1	2	3	No information
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

Table 6.8: Average waiting time per terminal type for alternative slack policies. The letters A, B, and C refer to the options mentioned in Table 6.7

6.7.3 Setting 3 - Unbalanced networks, restricted opening times terminals, sea vessels and closing times containers

Experimental Setting 3 is not easy to analyze, since it includes the extensions we considered in previous experimental settings. If we look at the average performance of the different levels of information exchange, we find the same pattern as seen before. We therefore focus on the comparison with the off-line benchmark. In Figure 6.15 we depict the performance of different controls, in terms the average activity tardiness, for a project tardiness penalty of zero. We only depicted *slack option 2* as slack policy for the waiting profiles, since the other slack policies lead to similar (though slightly worse) results. From the picture it is clear that the off-line benchmark outperforms the other controls. Figure 6.16 is similar to Figure 6.15, but now we depict the average project tardiness for a project tardiness penalty of ten. Also here we see that the off-line benchmark outperforms the other controls, especially for variable time windows.

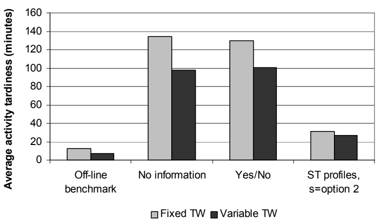


Figure 6.15: Average activity tardiness for different levels of information exchange, compared with the off-line benchmark. Results for project tardiness penalty equal to zero

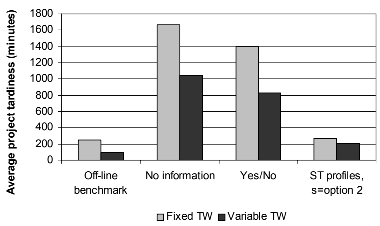


Figure 6.16: Average project tardiness for different levels of information exchange, compared with the off-line benchmark. Results for project tardiness penalty equal to ten

Analysis of the results reveals that raising the project tardiness penalty from zero to ten does not significantly affect the average project tardiness, but reduces the average activity tardiness slightly with about 5%. The explanation

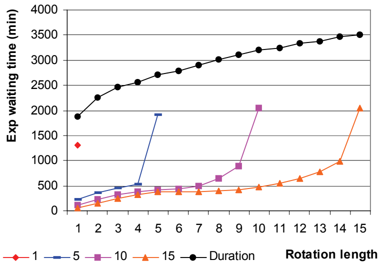


Figure 6.17: Average waiting time at t0 related to the position of the terminal in the rotation. For comparison we also plotted the average duration for different rotation lengths (1, 5, 10, and 15) and a project tardiness penalty of zero

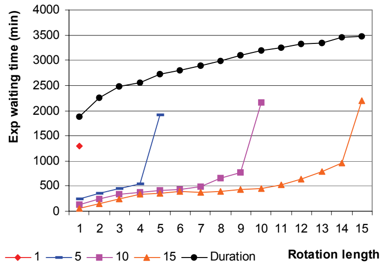


Figure 6.18: Average waiting time at t0 related to the position of the terminal in the rotation. For comparison we also plotted the average duration for different rotation lengths (1, 5, 10, and 15) and a project tardiness penalty of ten

for the fact that project tardiness is not significantly affected is that some terminals function as bottleneck, which means that the sojourn time of a barge is mainly determined by the waiting time at this terminal. In Figure 6.17 and 6.18 we depicted the waiting time announced by terminals at time t_0 (the time a barge plans its rotation), related to the position of this terminal in the rotation of the barge. We plotted this for different rotation lengths (1, 5, 10, and 15). In addition, we plotted the average duration of different rotation lengths. We conclude the following. First, it is clear that terminals with high waiting times at time t_0 are usually visited at the end of the rotation. Second, the waiting time at this terminal determines to a large extent the total sojourn time of the barge in the port. Third, we see that most terminals on average have some waiting time at time t_0 of about 400 minutes. This is probably due to the restricted opening times and sea vessels that block capacity for a significant amount of time. The fact that activity tardiness is only reduced by about 5% is probably caused by the fact that barges plan rotations successively and have very limited possibilities to reduce the activity tardiness, unless it is possible to swap activities of two barges at the same terminal. Moreover, terminals do not take into account the due date of a barge in their scheduling procedure. If terminals would give the barges with a due date a higher priority, the average tardiness might be reduced.

Figure 6.19 depicts the average sojourn time of barges for different rotation lengths in Experimental Setting 2 and 3. It is clear the sojourn of 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 Figure 6.20. Figure 6.20 shows the average amount of sailing, service, and waiting time in an average rotation (seven terminals) in Experimental Settings 2 and 3.

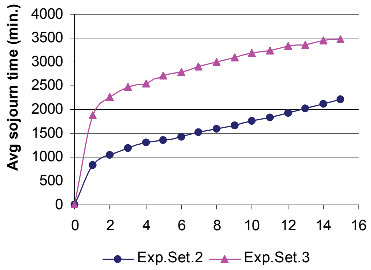


Figure 6.19: Average sojourn time of barges in Experimental Setting 2 and 3, related to the length of the rotation

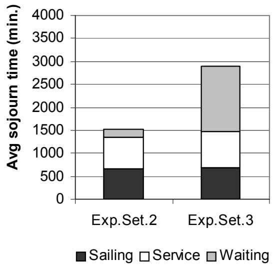


Figure 6.20: Average service, sailing, and waiting time for a rotation with 7 terminal visits. Results for Experimental Setting 2 and 3

6.8 Conclusions

In this chapter we considered practical extensions to make the model we developed in Chapter 5 suitable for a realistic situation. These extensions are restricted opening times of terminals, unbalanced networks, sea vessels, and closing times of containers. It turns out that waiting profile communication is not sufficient in these situations. The reason is that, besides waiting time, also the service time of a barge becomes time-dependent, e.g., when the barge handling cannot be finished before closing of the terminal. We therefore propose service-time profiles, which are an extension of the waiting profiles and include time-dependent service-times.

The change from waiting to service-time profiles, as well as the other practical extensions, requires some adaptations of the barge and terminal operator agent algorithms. We explained, e.g., how a service-time profile can be constructed and how barge operator agents can deal with closing times of containers. We performed different experiments to gain insight in the effect of a certain extension, such as restricted opening times, on the performance of barges and terminals. We compared the results with an off-line benchmark.

The experiments revealed that also in a more realistic setting service-time profiles in general perform well. If we look more specifically at the results, we saw that if terminals have restricted opening times, barges prefer not to visit these terminals when the terminals are about to close. The reason is that they want to reduce the risk to stay overnight at the terminal. If the utilization degree of the terminals increases, however, then more barges tend to stay at a terminal during the night. We found that the longer the terminals are closed, the longer the sojourn time of barges in the port. In case of unbalanced networks, we saw that an increase in total capacity of the network reduces the average sojourn time of barges. Moreover, the average waiting time at a terminal can be reduced further when terminals add different amounts of slack to their service-time profile. If we introduced also sea vessels and closing times of containers, then we found that the Multi-Agent based approach based on service-time profiles performs well compared to other levels of information exchange. However, the approach has some difficulty to minimize the activity tardiness. The reason is that barges plan rotations successively and have no opportunity to swap activities with other barges. Moreover, terminals do not take into account the closing times of activities during their scheduling. The latter two issues are interesting extensions of our model. Finally, we found that the introduction of sea vessels and closing times of containers lead to significant sojourn times of barges in the port, which is mainly caused by increased waiting time at terminals.

Based on the experimental results we think that an interaction protocol based on service-time profiles is promising in practice. Through the service-time profiles terminals can reveal to barges when the terminal can be visited, without giving too much insight in the operations of the terminal. We expect that the Multi-Agent based solution using service-time profiles may lead to a significant improvement of barge and terminal operations in the current situation.

Chapter 7

Extensions to the model

7.1 Introduction

In the Chapters 5 and 6 we developed a Multi-Agent system based on the assumptions that i) no disturbances occur during the operations of barges and terminals, and ii) barge and terminal operators want to make appointments for the handling of a barge. Both of these assumptions impact the design and performance of our Multi-Agent system. In this chapter we investigate the effect of relaxing both assumptions. For convenience we return to the port settings of Chapter 5. In Chapter 5 we considered simplified port settings, without restricted opening times, unbalanced networks, sea vessels, and closing times of containers.

In Section 7.2 we consider different degrees of cooperativeness of terminals[1]. We explore what the effect will be when terminals become less cooperative, which means that they share very limited information and make no appointments for the handling of barges. In Section 7.3 we make a first exploration into the effect of with dealing with disturbances in our Multi-Agent system. We discuss the concepts *disturbances* and *uncertainty,* the way terminal and barge operators can deal with it, and we make some remarks regarding the expected functioning of the Multi-Agent system.

7.2 The degree of cooperativeness of terminals

In the previous chapters we developed and evaluated an agent based solution to the barge handling problem. Underlying assumption in this solution is that terminals are fully cooperative in the sense that they are willing to make guaranteed agreements with barges about maximum waiting (or service) times and

[1] This section is based on the paper: A.M. Douma, P.C. Schuur, and R.G.M. Jagerman (2008). Degree of cooperativeness of terminals and the effect on the barge handling process. HMS2008 conference proceedings, Italy.

that they give insight in their occupation during the day. However, in practice the attitude of terminals might be less cooperative, e.g., terminals might not keep the agreements with barges or give limited insight in their occupation. The aim of this section is to provide insight in the effect of the degree of cooperativeness of terminals on the barge handling process.

In Section 7.2.1 we describe literature related to the concept of cooperativeness. In Section 7.2.2 we introduce the different degrees of cooperativeness of terminals. We describe these degrees in more detail in Section 7.2.3 and 7.2.4. Section 7.2.5 describes the experimental settings and the results of the simulations. In Section 7.2.6 we draw some conclusions.

7.2.1 Related literature

A new element in this chapter is the concept 'degree of cooperativeness'. In the literature on Multi-Agent systems the concept cooperation has been frequently discussed in different meanings. Cooperative agents are, e.g., considered as agents working together to achieve the same goal, in contrast to agents that are self-motivated and maximize their own benefits (Kraus, 1997). Sandholm (1999) states that self-interested agents can be assumed to be cooperative, if they use the strategies imposed by the designer and not choose a strategy themselves. The latter might be more likely in problems with competing self-interested agents. In these situations the design of the communication protocol becomes important, to let the agents exhibit desired behavior (Sandholm, 1999). The concept of cooperation in Multi-Agent systems is also strongly related to the field of (Cooperative) Game Theory. In a game (self-interested) players usually have a choice to adopt a cooperative attitude or not and they make a decision based on expected payoffs. This choice might be in favor of being cooperative, especially when players have long-term relationships and face each other in repeated games (see, e.g., Mailath and Samuelson, 2006).

The long-term relationships between terminals and barges might influence the decisions both actors make and the service they are willing to offer. It turns out that in the barge handling problem, the behavior of terminals is difficult to regulate within the system (see Section 7.2.2 for an explanation). However, when terminals offer better services to barges, their relationship might improve in the long term. Services can be, e.g., guarantees on waiting times (Kumar et al., 1997; Whitt, 1999).

In this chapter we give insight in the effect of different degrees of cooperativeness of terminals on the barge handling process. The results can be used by terminal operators to decide which degree of cooperativeness they should adopt when a Multi-Agent system is implemented.

7.2.2 Degree of cooperativeness

In the Multi-Agent system proposed in the previous chapters we assume that terminals are 'fully' cooperative. What we mean is that terminals give barges first a waiting profile (expressing the maximum waiting time during a certain time period) and, second, make appointments which guarantee barges a maximum waiting time giving a latest arrival time at the terminal. In the current situation, however, terminals have a dominant position in the port. For a terminal it is of little importance if a barge has to wait a few hours. Terminals can even benefit from long queues, since this reduces their risk of quay idle time. This is not in the interest of barges, but in the current situation they have no power base to force a terminal to behave differently. The terminals, on the other hand, can force barges to show desired behavior, by refusing their processing and let them wait some additional time to be processed. From an implementation perspective we think it is interesting to investigate to what extent increasing cooperativeness would influence the barge handling process. We therefore introduce the concept 'degree of cooperativeness'.

With degree of cooperativeness we mean i) the extent to which a terminal gives insight in its occupation during the day and ii) the extent to which a terminal is willing to keep an appointment. We consider three degrees of cooperativeness:

1. *Fully cooperative*: a terminal issues waiting profiles and keeps the appointments made with barges.
2. *Partly cooperative*: a terminal issues waiting profiles, but processes barges first-come first-served (FCFS).
3. *Lowly cooperative*: a terminal gives (limited) insight in the current state of the barge queue at the terminal, and processes barges first-come first-served (FCFS).

Note that appointments are only made in the fully cooperative case. The model and the results for the fully cooperative case are already presented in Chapter 5. In this chapter we discuss the lowly and partly cooperative case. For the fully and partly cooperative case we assume that waiting profiles are issued only once per rotation. In the lowly cooperative case, however, a terminal informs the barge about the current state of the queue at any moment a barge asks for it, even repeatedly. The model we apply for the lowly cooperative case is different from the model we use for the fully and partly cooperative case and is introduced in Section 7.2.3.

One might argue that in the lowly cooperative case terminals are not that uncooperative, since they provide information at any moment a barge asks for it. We still stick to the label 'lowly cooperative' for two reasons. First, the quality of information is low (only the current queue length). Second, this label can also be used in the case of absolute lack of terminal information, provided that barges are learning about queue lengths by other means such as barge transponders, eye-sight, or friendly colleagues.

We first describe the lowly cooperative and then the partly cooperative case.

7.2.3 Terminals are lowly cooperative

If terminals are lowly cooperative we assume that no appointments are made between barges and terminals, and that terminals give limited insight in the current state of the barge queue. Barges decide during their stay in the port in which sequence terminals are visited. A decision can be based on all kinds of available information, such as the number of barges waiting or the total waiting time in the queue. We first describe the model we use for this degree of cooperativeness and discuss some alternative decision rules.

Model

The model we use to simulate the lowly cooperative case is different from the models in Chapter 5 and 6. In these chapters we assume that barge operator agents plan a rotation on arrival in the port and do not reconsider their rotation later on. In the lowly cooperative case barges do not plan a rotation once on arrival in the port, but they construct their rotation by choosing terminals successively during their stay in the port. We have modeled this as follows. On the geographical layout of the port we map a network of decision points. Decision points are virtual points at which a barge operator agent has to make a decision. The idea is that barges sail (virtually) from decision point to decision point. In this way barges make decisions as late as possible, which gives an agent the possibility to include the most recent information in the decision. Figure 7.1 gives a conceptual representation of the network of decision points.

We define the following decision points (see Figure 7.1). On arrival in the port the barge operator agent first chooses which region (cluster of terminals) it wants to visit. This decision has to be made in node *start*. On arrival in this region it either decides to pass the region or to visit a terminal (node *choose terminal or leave region*). If it decides to visit a terminal, the barge operator can decide, on arrival at the terminal, to enter the queue (node *decide to enter queue at terminal*). If it enters the queue it can reconsider from time to time whether it keeps on waiting at the current terminal or leaves the queue to visit another terminal (node *decide to keep waiting*). After it has visited a terminal it can decide to go to another region or to stay in the current region (node *choose region*). If it decides to stay in the current region, the agent has to choose which terminal to visit or yet sail to another region. If it has decided to go to another region it sails there and on arrival in this region it again has to decide either to visit a terminal or to go to another region.

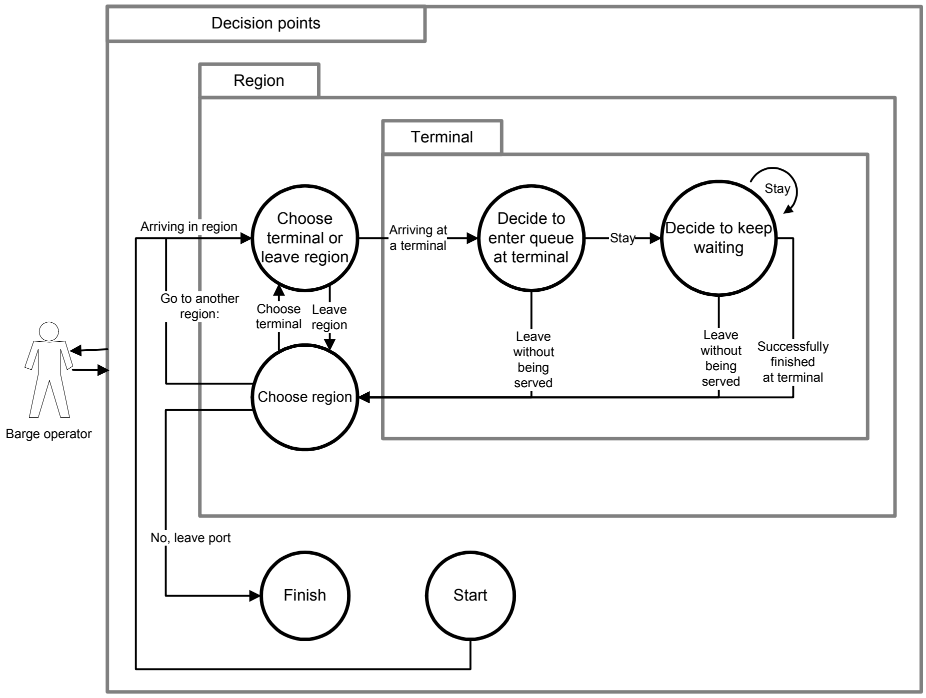


Figure 7.1: Decision points for a barge operator agent in the lowly cooperative case

The decisions that have to be made at the decision points can be based on different kinds of information, depending on the willingness of terminals to share this information. In the next section we discuss four alternative decision rules.

Alternative decision rules

In this section we present four decision rules. A barge uses a decision rule throughout its rotation. Every decision rule covers decisions in each of the following nodes: *start*, *choose terminal or leave region,* and *choose region.* For simplicity we assume that in node *decide to enter queue at terminal* a barge always decides to enter the queue, and in node *decide to keep waiting* to keep on waiting until the processing has been completed.

We explore four decision rules (DRs). The first decision rule (DR1) uses *no information* about the state of the terminals in the port. Decision rule 2 (DR2) uses information about the *number of barges* waiting in the queue of each terminal in the rotation belonging to the region considered. The third decision rule (DR3) uses information about the *estimated waiting time in the queue* of each terminal in the rotation belonging to the region considered. Decision rule 4 (DR4) uses information about the estimated waiting time in the queue of each terminal in the rotation and the estimated average waiting time in the queue of all terminals in the port. The latter gives the barge operator insight in the average busyness of all terminals of the port. The four decision rules are a choice, one can of course define several other decision rules using other kinds of information. For now, these four decision rules are sufficient to give us insight in the effect of different kinds of information on the performance of barges in the lowly cooperative case.

When describing the decision rules (DR) we also explain what kind of information a terminal is assumed to give to a barge, either the number of barges waiting in the queue or the estimated total waiting time in the queue. For convenience we introduce the following notation: let L_j denote the number of barges waiting in the queue at terminal j and W_j the estimated waiting time in the queue at terminal j. We denote the total set of terminals in the port by $\mathcal{N}$. Suppose that a barge is in region R. Then let $\mathcal{N}^R \subset \mathcal{N}$ denote the set of all terminals in R that still have to be visited by the barge. The set $\mathcal{N}^R$ is a dynamic set and is updated every time the handling of a barge is successfully completed at a terminal. In one of the decision rules (DR4) we use the *general queue value.* The general queue value $\overline{W}$ expresses the average W_j of all terminals $j \in \mathcal{N}$. The information to determine this value is provided by all barges, every time a barge enters the queue of a terminal. The general queue value is updated by exponential smoothing (with a smoothing factor of 5%).

We describe for every decision rule in every decision node, the decision that is made and the information that is needed from the terminals.

DECISION NODE: START

Decision rule 1-3 (DR1-3)

- Decision: Solve a traveling salesman problem (assuming deterministic sailing times) and head to the region of the first planned terminal.
- Information: none

Decision rule 4 (DR4)

- Decision: Solve a traveling salesman problem (assuming deterministic sailing times). Based on the solution make the following decisions: i) go to the region of the first planned terminal, and ii) determine how often each region in the port is visited. Note that in a line network a barge will visit regions A and B twice, and region C only once, due to the sailing distances between regions. In a triangular network, only region A is visited twice. See Figure 4.10 for a picture of the network layouts.
- Information: none

Figure 7.2: The decision made in each of the four decision rules in node 'Start'

DECISION NODE: CHOOSE TERMINAL OR LEAVE REGION

Decision rule 1 (DR1)

- Decision: Go to the first planned terminal $\widehat{j} \in \mathcal{N}^R$
- Information: none

Decision rule 2 (DR2)

- Decision: Go to terminal $\widehat{j} \in \mathcal{N}^R$ where $\widehat{j} = \arg\min_{j \in \mathcal{N}^R} L_j$
- Information: L_j of every terminal $j \in \mathcal{N}^R$

Decision rule 3 (DR3)

- Decision: Go to terminal $\widehat{j} \in \mathcal{N}^R$ where $\widehat{j} = \arg\min_{j \in \mathcal{N}^R} W_j$
- Information: W_j of every terminal $j \in \mathcal{N}^R$

Decision rule 4 (DR4)

- Decision: Go to terminal $\widehat{j} \in \mathcal{N}^R$ where $\widehat{j} = \arg\min_{j \in \mathcal{N}^R} W_j$, if at least one of the following two conditions is true: either i) $W_{\widehat{j}} \leq \overline{W}$, or ii) the region is not visited again. Otherwise, leave the current region.
- Information: W_j of every terminal $j \in \mathcal{N}^R$

Figure 7.3: The decision made in each of the four decision rules in node 'Choose terminal or leave region'

DECISION NODE: CHOOSE REGION

Decision rule 1-3 (DR1-3)
- Decision: Go to the region of the first planned terminal
- Information: none

Decision rule 4 (DR4)

- Decision: Leave the current region if one of the following two conditions is true: either i) all terminals are visited, or ii) no terminal $\widehat{j} \in \mathcal{N}^R$ has a $W_{\widehat{j}} \leq \overline{W}$ and the region will be visited again. Otherwise stay in the current region.
- Information: W_j of every terminal $j \in \mathcal{N}^R$

Figure 7.4: The decision made in each of the four decision rules in node 'Choose region'

The four decision rules are meant to explore the value of various kinds of information. In future research these rules may become more sophisticated, taking into account opening times of terminals and arrivals of sea vessels (like the port settings studied in Chapter 6).

Note that the network of decision points is a virtual network. If we say in decision rule 1 for instance that a barge heads to the region of the first planned terminal, this actually means that the barge first determines what the next region is. If this is the same region as it is in now, then it directly sails to the next planned terminal. If this region is different, then is will physically move to the other region and decide (on arrival in this region) which terminal to visit.

In future research one might consider other and even hybrid decision rules, where barges use different decision rules in their rotation on a cherry picking basis.

7.2.4 Terminals are partly cooperative

If terminals are partly cooperative they issue a kind of waiting profile which can be used by barges to determine their best rotation. However, the maximum waiting times expressed in the waiting profile are not guaranteed. On the contrary, barges are processed in the order they arrive at the terminal (FCFS).

The model we use is similar to the model in Chapter 5 and 6. Barges plan their rotation on arrival in the port and use the waiting profiles to minimize their expected sojourn time in the port. Once they have determined their best rotation they announce their expected arrival time to the terminal, assuming that the maximum waiting times in the waiting profile are valid maximum waiting times. However, during execution of the rotation, the waiting times might be different from what is announced, since other barges might have arrived earlier. Waiting profiles are therefore not more than an indication of the busyness of the terminal during certain periods of the day. The announced expected arrival times of the barges are therefore also not more than an indication, subject to the waiting times at other terminals during the rotation.

Barges do not update their rotation during execution, but visit the terminals in the sequence determined on arrival of the port.

7.2.5 Experimental settings and results

We use simulations to evaluate the performance of barges and terminals in the partly and lowly cooperative case. For the fully cooperative case we use the simulation results of Chapter 5. To simulate the partly cooperative case we adjusted our simulation model used in Chapter 5 such that terminals still issue waiting profiles but process barges FCFS. For the lowly cooperative case we developed a new simulation model. In our experiments we consider data sets 0 and 1, introduced in Section 4.6. All scenarios have a run length of 100 days. We apply a warm-up period of ten days and a cool-down period of three days.

In this section we describe the results of the simulations. We successively describe the performance of the various degrees of cooperativeness, compare the performance of the four different decision rules in the lowly cooperative case, evaluate the impact of making appointments compared to processing barges FCFS, and finally describe some aspects that have to be taken into account when drawing conclusions regarding the performance of various degrees of cooperativeness.

The performance of the various degrees of cooperativeness

If we compare the different degrees of cooperativeness in terms of average project tardiness, we find the results as given in Figure 7.5 and 7.6. The results for the fully cooperative case correspond with waiting profiles with 30 and 60 minutes slack. The results of the lowly cooperative case are based on decision rule 4 (DR4), which outperforms the other decision rules (we elaborate on this in one of the next sections). In the figures we see two interesting results. First, the fully and the lowly cooperative case perform quite similarly. This might be surprising. The advantage of the lowly cooperative case compared to the fully cooperative case is that barges are processed FCFS. This means that a terminal will never be idle while at the same time a barge is waiting in the queue. In case of appointments, terminals are sometimes forced to stay idle and wait for a barge which is about to arrive, despite the fact that there are other barges waiting in the queue. If the workload in the system is equally distributed over all terminals, then we expect that -in this simplified port setting- the lowly cooperative case may realize an even higher throughput than the fully cooperative case.

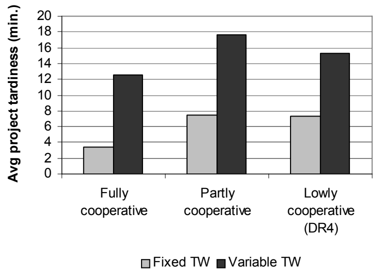


Figure 7.5: The average project tardiness averaged over all scenarios with a 50% utilization degree. Results are specified for fixed and variable time windows

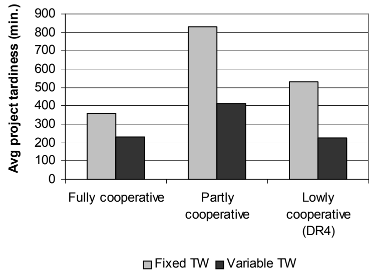


Figure 7.6: The average project tardiness averaged over all scenarios with a 90% utilization degree. Results are specified for fixed and variable time windows

However, we need to make two remarks regarding the results. First, in our data set we did not include sea vessels or restricted opening times of terminals. This might have a big impact on the decision rules in the lowly cooperative case, since then information about down-time of the terminals becomes important. Second, to interpret the performance of the fully and lowly cooperative case in the right way, we also have to examine the relation between the duration of a rotation and the number of terminal visits (calls) in a rotation. We get back to this in one of the next sections.

A second interesting result we derive from Figures 7.5 and 7.6, is that the lowly cooperative case outperforms the partly cooperative case. The reason for this is that barges in the partly cooperative case determine a rotation (and announce the corresponding arrival times) based on the issued waiting profiles. This has some disadvantages. First, the waiting profiles might be outdated since barge arrivals might be significantly different from the announced arrival times. This effect is larger for higher utilization degrees. Second, barges assume that in the waiting time of one terminal another terminal can be visited. This is not true, since terminals process barges FCFS. The fact that lowly cooperative outperforms partly cooperative is interesting from a barge perspective. This suggests that, if terminals are reluctant to provide any information to the barges, then barges will surely benefit from joining their forces and exchange information on a mutual basis.

If we consider the average waiting time of a barge at a terminal we find the numbers presented in Table 7.1. The average waiting times are measured for a scenario with network layout II, 9 terminals per region, terminals have one quay and a utilization degree of 90%. The results indicate that a low average

Degree of cooperativeness	**Avg waiting time at terminal (min.)**
Fully	181
Partly	304
Lowly (DR4)	199

Table 7.1: Average waiting time of a barge at a terminal for different degrees of cooperativeness. These results were obtained for data set 0: scenario network layout II, 9 terminals per region, terminals have one quay and a 90% utilization degree of terminals

project tardiness corresponds with low waiting times at terminals. We also see that the fully and lowly cooperative case perform quite similarly.

Comparing the alternative decision rules in the lowly cooperative case

In Figure 7.7 and 7.8 we depict the performance of the different decision rules for a utilization degree of 50 and 90%, and for one and two quays. It is clear that the use of more information leads to a significant improvement of the average project tardiness of barges. More information allows barges to distribute their total workload more evenly over all the terminals, thus reducing the average waiting times at terminals.

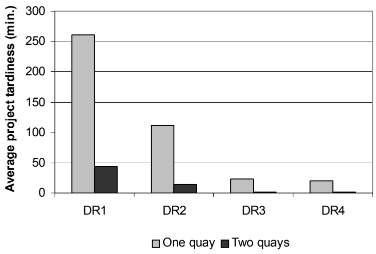


Figure 7.7: Average project tardiness of barges for different decision rules (DR) in the lowly cooperative case. Results for a utilization degree of 50%

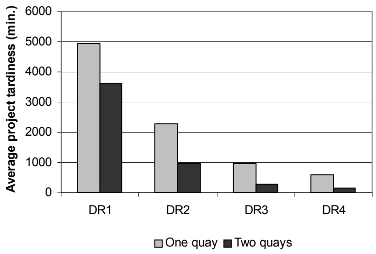


Figure 7.8: Average project tardiness of barges for different decision rules (DR) in the lowly cooperative case. Results for a utilization degree of 90%

Differences in appointment based processing and FCFS processing

When we take a closer look at the relation between the average total waiting time and the length of the rotation, we find the relations depicted in Figure 7.9 (for the fully cooperative case) and in Figure 7.10 (for the partly and lowly cooperative case). Note that Figure 7.9 is the same as Figure 5.7 in Chapter 5. The depicted relations are again for a scenario with network layout II, 9 terminals per region, terminals have one quay and a 90% utilization degree.

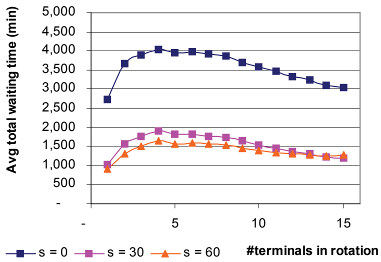


Figure 7.9: Average total waiting time for different levels of slack and different rotation lengths. This is the situation in the fully cooperative case

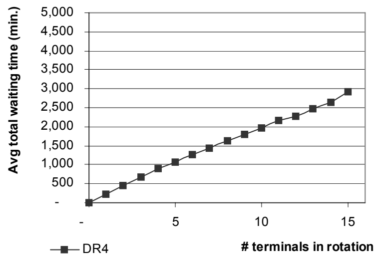


Figure 7.10: Average total waiting time related to different rotation lengths. This is the situation in the lowly and partly cooperative case

In Figure 7.10 we see that for the partly or lowly cooperative case the relation between the average total waiting time and the number of calls in a rotation is rather linear. In the fully cooperative case this relation is totally different as we have explained in Chapter 5 (see Figure 7.9). A barge in the fully cooperative case can make appointments with bottleneck terminals and visit in the mean time other terminals to load and unload containers. This is different in the lowly and partly cooperative case, because terminals process barges FCFS. On every arrival at a terminal a barge has to enter the queue and wait for its service until all earlier arrived barges are processed.

The relations depicted in Figures 7.9 and 7.10 hold especially when the utilization degree of terminals is high (more than 75%). The differences between the three degrees of cooperativeness are less apparent for lower utilization degrees. In the fully cooperative case the relation is even depending on the amount of slack added to the waiting profile. The reason why the differences becomes less apparent is that terminals have relatively much time to process barges and waiting times at terminals are low. It is therefore less important to visit the terminals at the 'right' time than when terminals are highly occupied. This will be different if terminals also close during the night or process sea vessels.

To see the impact of processing barges based on appointments or FCFS, we depict for comparison two situations (see Figure 7.11 and 7.12). In both situations barges receive no information of terminals. The only difference is that barges are processed based on appointments or FCFS. In case of appointments this corresponds with the level of information exchange 'No information' (see Section 3.6). In case of FCFS processing this corresponds with decision rule 1 (see Section 7.2.3). In both situations barges plan a rotation by solving a TSP, so they do not take any waiting information into account. Clearly, FCFS leads to a significant improvement over making appointments.

The reason why FCFS performs better is that terminals in this case are never

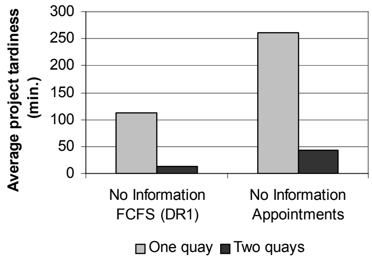


Figure 7.11: *Average project tardiness per barge. Results aggregated for data sets 0 and 1, specified for a 50% utilization degree*

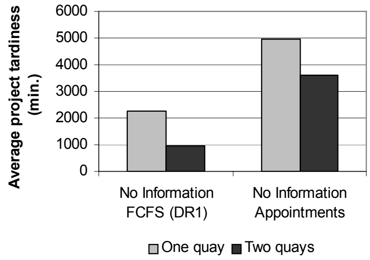


Figure 7.12: *Average project tardiness per barge. Results aggregated for data sets 0 and 1, specified for a 90% utilization degree*

idle while there are barges waiting in the queue. In case of appointments, terminals are sometimes forced to stay idle to wait for a specific barge.

Interpretation of the differences for the various degrees of cooperativeness

We need to be cautious in the comparison of the performance of the different degrees of cooperativeness. The reason for this is a bit hard to explain and has to do with the fixed and variable time windows. Consider the situation of *fixed time windows*. In this situation all barges, regardless how many terminals they visit, get the same time window. If we then look at Figure 7.9, the situation that terminals are fully cooperative, one can imagine that almost all barges (regardless of the number of terminals they visit) can have difficulty to meet the time window. If terminals are less cooperative (see Figure 7.10) this mainly holds for the barges that visit many terminals. We illustrate this in Figure 7.13 by depicting the average project tardiness per rotation length. The results are given for a scenario with network layout II, 9 terminals per region, terminals have one quay and 90% utilization degree.

If we use *variable time windows* we have a different situation. Barges then get a time window which increases linearly with the number of terminals they have to visit. Consider again Figure 7.9. One can imagine that in this case, when terminals are fully cooperative, barges with small rotation lengths have trouble to meet their time window. In the lowly and partly cooperative case this holds more equally for all barges (see also Figure 7.14).

The partly and lowly cooperative situations therefore seem to be better suited for variable time windows, since the sojourn time in the port increases linearly with the number of terminal visits. The fully cooperative situation, on the contrary, better suits fixed time windows, since the sojourn time of barges is more constant in the number of terminals they visit. The reason we mention

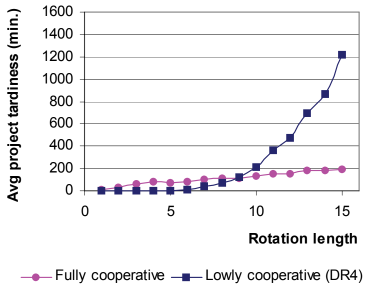


Figure 7.13: Average project tardiness for different rotation lengths in case of fixed time windows. Specified for the fully and lowly cooperative case

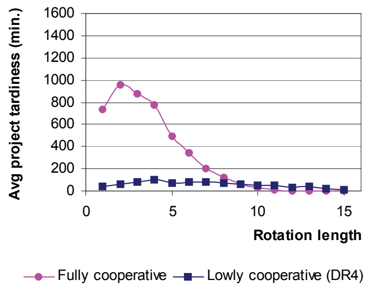


Figure 7.14: Average project tardiness for different rotation lengths in case of variable time windows. Specified for the fully and lowly cooperative case

this is that both models may show an equal number of late barges, but the group of barges which is late differs. We therefore need to be a bit cautious interpreting the results.

Barge operators can use these insights in the following way. If terminal operators are lowly or partly cooperative, then the average barge port sojourn time increases (more or less) linearly with the number of terminals a barge visits in the port. If terminals are fully cooperative, on the other hand, then the average barge sojourn time is mainly determined by the waiting time at the bottleneck terminals and less on the number of terminal visits. This means that in the former situation it can become attractive to reduce the average rotation length of a barge, and in the latter situation to obtain a convenient time slot at the bottleneck terminal.

7.2.6 Comparing the three degrees of cooperativeness

Our simulation results suggest that the way terminals deal with barges, influences to a large extent the performance of these barges. If terminals are fully cooperative, then there is more certainty about the barges that have to be processed, whereas in the lowly cooperative case terminals are not restricted by appointments and have more flexibility in their operations.

Although being fully cooperative seems to be a good alternative, there is also a down-side for terminals. If terminals are fully cooperative, there is no real incentive for barges to reduce the number of terminals in a rotation. Barge operators can do so if they organize their hinterland operations in a different way, by letting barges only pick-up containers in the hinterland destined for a limited number of terminals in the port. Reducing the number of terminal visits leads only to a small reduction of the port sojourn time. However, more terminal visits per barge imply more dependencies between terminals, and on average smaller call sizes per barge. This results in more idle time of the crane during mooring of the barges. FCFS processing gives barges a clear incentive to reduce the number of terminal visits and is more robust against disruptions. However, FCFS leads also to several disadvantages during operations. First, terminals do not exactly know when a barge is to be processed and when containers need to be stacked at the quay. Second, barge operators need to stow their barges very flexibly to be able to visit terminals in a different order. This affects the utilization degree of the barges. If barges make appointments, there is more certainty about their rotation, which enables them to increase their ship utilization. Third, there is more uncertainty in sojourn times of barges in the port, which makes the sailing schedules offered to carriers less reliable.

In general we conclude that if terminals process barges more or less FCFS, then barges are better off when no appointments with terminals are made. The reason why is that FCFS processing results in very efficient performance of the terminals (they always process barges if there are waiting any) and that appointments (which are not met by terminals) pretend certainty about the handling times, but in fact frustrate the rotation planning of barges. We also conclude that the more equally the workload of all barges is spread over the network (in case of FCFS processing), the lower the average waiting times at terminals and the better the performance of the system.

It is subject to discussion which degree of cooperativeness is desirable from both a terminal and a barge perspective. If terminals are fully cooperative, we expect that the balancing of the total workload over all terminals can be done more effectively than when terminals are less cooperative. Especially, when terminals also have restricted opening times and handle sea vessels. This will generally result in shorter waiting times for barges at terminals so that more barges can depart the port timely. Another benefit of being fully cooperative is that terminals then have more possibilities to manage their workload over the day. If terminals are partly cooperative, the value of waiting profiles may deteriorate dramatically, since terminals process barges at the time they prefer, but also barges might decide to visit terminals in a different sequence. Waiting profiles may then become misleading, since barges and terminals can act freely upon this information making it less relevant. If terminals are lowly cooperative, on the contrary, we expect that especially the introduction of sea vessels and opening times at terminals makes it hard to decide which terminal to visit when. Especially, in a situation, such as in Rotterdam, where barges visit several regions twice and have the option to go to a terminal either the first time a region is visited or the second time.

One can imagine that in practice also a mix of degrees of cooperativeness can be found among terminals; for instance, terminals that participate in the Multi-Agent system (and are fully cooperative) and terminals that are not willing to participate. The latter terminals will cause a lot of uncertainties about the time a barge needs to be processed at those terminals, which makes it harder to make appointments with fully cooperative terminals. In this hybrid setting it is essential for barges that they make appointments such, that they can visit the less cooperative terminals in between the appointments.

The flexibility in case of FCFS processing might be especially attractive when terminals frequently deal with disturbances. In the next section (Section 7.3) we make a first explorative step to study how barge and terminal operators can deal with disturbances in case terminal operators decide to be fully cooperative. Recall that being fully cooperative implies that a terminal operator makes (and keeps) appointments with barge operators.

7.3 Dealing with disturbances and uncertainties

In the previous chapters we did not consider disturbances during the operations of barges and terminals. However, in practice there are several uncertainties terminal and barge operators have to deal with. These uncertainties are partly the result of the way the alignment process is currently organized and partly inherent to the operations of barges and terminals. The former kind of uncertainties we try to reduce by the introduction of our Multi-Agent system, however, the latter kind of uncertainties have to be dealt with within the system.

In the previous section (Section 7.2) we argued that the flexibility terminals have, in case they process barges FCFS, might be especially attractive when dealing with disturbances. However, FCFS processing also introduces uncertainty for terminals and barges, e.g., in arrival and waiting times at terminals. In this section we assume that terminals are fully cooperative. We reflect on uncertainties barge and terminal operators experience and we aim to provide some directions of how terminal and barge operators can deal with uncertainty in their operations. We do not intend to give an extensive discussion of the topic.

The outline of this section is as follows. In Section 7.3.1 we discuss several uncertainties in the operations of barge and terminal operators nowadays, and we mention some relevant dimensions of uncertainty. Section 7.3.2 introduces the concept of robust scheduling. In the Sections 7.3.3, 7.3.4, and 7.3.5 we discuss successively how the Multi-Agent system and the operations of the terminal and barge operators can be made more robust. Section 7.3.6 discusses the way barge and terminal operators can deal with large disruptions. In Section 7.3.7 we mention some drawbacks of buffering uncertainty. In Section 7.3.8 we do some preliminary tests on dealing with uncertainty. Finally, we draw some conclusions in Section 7.3.9.

7.3.1 Different uncertainties in practice

In practice terminal and barge operators deal with different uncertainties. Several examples are given in Connekt (2003) and Moonen and Van de Rakt (2005) for our case of reference, the Port of Rotterdam. Terminals deal with uncertainty in, e.g., the arrival times of sea vessels, the availability of resources, the arrival times of barges, the workload of barges, and the number of barges that arrive during a day or week. Barges deal with uncertainty in, e.g., the handling and waiting times at terminals, the (technical) performance of the barge, the sailing times between terminals, the availability of containers, and the arrival times in the port.

Uncertainties differ in the frequency with which they happen and in their duration (or their impact). To give an example: a crane breakdown at a terminal happens incidentally and reduces the terminal capacity for a significant amount of time. A small deviation in the call size of a barge happens frequently and the duration (or impact) is relatively small. Moreover, uncertainty is also related to time. The expected arrival time of a sea vessel is more uncertain one month prior to the arrival of the sea vessel than when the sea vessel is about to arrive at the terminal. Moreover, Connekt (2003) shows that uncertainties in the port can have different causes and can be a cause for uncertainty itself. Through these interrelations, a small disruption can indirectly have a major impact on several processes.

In the next sections (Sections 7.3.2 to 7.3.5) we focus on disturbances that happen frequently or have a limited impact, such as deviations in the call sizes of barges. In Section 7.3.6 we discuss disturbances that happen less frequently and have a major impact, such as crane breakdowns.

7.3.2 Robust scheduling

Terminal and barge operators can undertake different actions to deal with uncertainty in their respective rotation plans and quay schedules. Recall, that we focus in this section on disturbances that happen either frequently or have a limited impact. There is a large body of literature on scheduling under uncertainty (see, e.g., Leon et al., 1994; Jensen, 2001; Leus, 2003; Wullink, 2005; Herroelen and Leus, 2005). In general, two approaches are mentioned to deal with uncertainty, namely proactive and reactive approaches (see, e.g., Leon et al., 1994; Leus, 2003). Proactive approaches concern actions prior to the occurrence of a disturbance, and reactive approaches concern actions in response to a disturbance. With respect to *proactive approaches*, usually two options are considered (Leus, 2003). These are the creation of a baseline schedule (predictive schedule) or creating no baseline schedule (full dynamic scheduling). A baseline schedule serves different functions (Mehta and Uzsoy, 1998; Leus, 2003). First, it allows for the allocation of resources to the different activities. Second, it serves as basis to plan external activities, which are related to the execution of the activities in the baseline schedule. Third, a baseline schedule can be useful when uncertainty is not purely stochastic, but 'manageable' to a certain extent. Full dynamic scheduling means that a schedule is created dynamically as time evolves, by deciding on freeing of a resource which activity is to be started next (Leus, 2003) Full dynamic scheduling can be done by means of dispatching policies, such as 'earliest due date first' (see, e.g., Pinedo and Chao, 1999). The lowly cooperative case in Section 7.2 is an example of a dynamic scheduling approach, since terminals make no baseline schedule and process barges FCFS.

Reactive approaches are meant to repair the baseline schedule after a disturbance has occurred. There are several techniques to repair the baseline schedule, varying from simple policies (right shifting of activities) to more complex actions (full re-scheduling). The result of these actions might be that the new baseline schedule strongly deviates from the old baseline schedule. The question is whether this is desirable.

Robust scheduling is a form of proactive scheduling. Different authors adopt different (though similar) definitions of robust scheduling. Jensen (2001) defines a robust schedule as a schedule that is expected to perform well after a minimal amount of modification when the environment changes. Leus (2003) defines a robust schedule as a schedule that is protected as well as possible against uncertain events that can occur during the execution of the schedule. The definition of robustness or a robust schedule is often further specified in the literature. We mention the following 'forms' of robustness:

- Stability: a schedule is called stable if it is able to suppress propagation of disruptions both within the individual project and towards other projects (Leus, 2003).
- Flexibility: a schedule is called flexible if it can be easily repaired, i.e., changed into a new high-quality schedule (Sevaux and Sörensen, 2002).
- Nervousness avoidance: a schedule is called nervousness avoiding if it can avoid frequent, large changes (after Jensen, 2001).

Note that different authors use different 'forms' and definitions of robustness. Leus (2003), e.g., mentions the term 'quality robustness' as the insensitivity of the schedule performance in terms of the objective value, and 'solution robustness' as the insensitivity of the activity start times to changes in the input data. Both forms of robustness are related to stability.

We use the mentioned forms of robustness, since they are useful to start thinking about robust terminal quay schedules and barge rotations.

7.3.3 A robust system

In this section we analyze the barge handling problem from a robustness perspective. We still focus on disturbances that happen either frequently or have a limited impact. We first consider the problem on a system level. In the Sections 7.3.4 and 7.3.5 we consider the operations of terminals and barges separately. To analyze the problem on a system level, we use the model of the off-line benchmark. In Figure 4.8 we see that there are all kinds of dependencies between activities. In Figure 7.15 we give an example to clarify these dependencies. Recall that an activity is the handling of a barge at a terminal. Every activity is succeeded by an activity on the terminal and is possibly succeeded by an activity at another terminal.

From a terminal perspective these interdependencies of activities are not desirable but inescapable. For a terminal operator it is therefore important to reduce the dependency on the operations of other terminals as much as possible (the vertical arcs in Figure 7.15). This can be influenced by the barge operators. When they schedule terminal visits tight to each other, then there is little slack in between terminal visits to recover from a disturbance. A terminal operator, however, does not know how riskful a barge operator has planned his rotation and how uncertain the barge's arrival time is. Barge operators, on the other hand, depend on the performance of terminals and indirectly on the performance of other barges (the horizontal arcs in Figure 7.15). If terminal operators plan their quays such that uncertainties can be recovered without violating appointments made with barges, then disruptions can fade out quickly. If terminals, however, make schedules which are vulnerable for a disruption, it will certainly affect the operations of other barges and indirectly other terminals.

The dependencies we just described are related to the first form of robustness, namely stability. It is important that disruptions cannot propagate from one terminal or barge to another. For the other actor this uncertainty is uncontrollable and can therefore hardly be influenced. In the design of our Multi-Agent system we took this kind of robustness already into account. We did this by defining appointments in a specific way (see Sections 3.3.2 and 3.5.2). The idea is that barges become responsible for a timely arrival at a terminal. This implies that a barge operator, although it is opportunistic, uncouples the operations of terminals by including more slack in between terminal visits to guarantee a timely arrival.

To let this mechanism work, it is important that terminals really keep their appointments and force barges to do so as well. If terminal operators take appointments not very strict, but expect barge operators that they do, then barge operators will probably perceive this as (highly) unfair. It means that barge operators have to buffer the uncertainty that terminal operators create. To motivate barge operators to keep their appointments, we think it is necessary

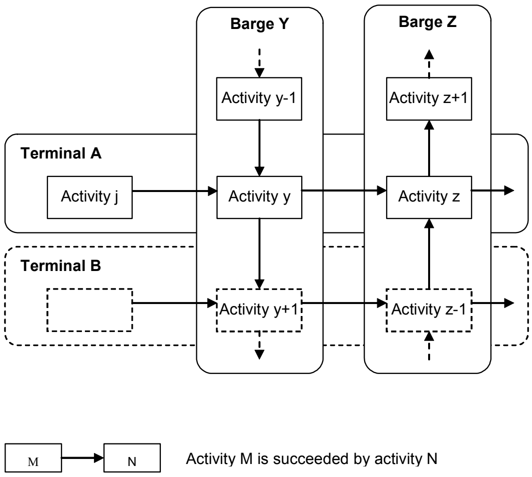


Figure 7.15: An example of the interrelation of activities of terminals and barges

that terminal operators keep their appointments as well. If either barge or terminal operators start to take appointments not serious anymore, then the reliability of the whole system will reduce.

7.3.4 Robust quay schedules

In this section we focus on the terminal operator and the way he can deal with disturbances that happen either frequently or have a limited impact. To deal with uncertainties, a terminal operator can choose to apply a proactive (or robust) scheduling or a (full) dynamic scheduling approach. In our opinion, a robust scheduling approach (including the construction of a baseline schedule) has several advantages. First, it allows a terminal operator to make reliable appointments and to influence his workload during the day. Second, it allows for a timely initiation of activities that precede the barge handling process, such as the release of containers and the stacking of containers at the quay side. Third, it allows for anticipating uncertainties.

To analyze how a terminal operator can create a robust base line schedule, we describe the three forms of robustness (stability, flexibility, and nervousness avoidance) successively.

A *stable* baseline schedule for a terminal operator is a baseline schedule that is able to suppress the effect of disruptions. This means, among others, that the baseline schedule has to enable the terminal operator to keep as many appointments as possible. The latter implies that the terminal operator has to focus on solution robustness, i.e., the insensitivity of the activity start times to changes in the input data (Leus, 2003). To realize solution robustness, the terminal operator has to schedule barges such that the probability that the handling of a barge is started later than its latest starting time is negligible.

A *flexible* base line schedule means that a schedule can be easily repaired after a disruption. The possibilities to repair a schedule are determined to a large extent by the appointments that are made. If a particular barge is planned close to a sea vessel or closing of the terminal, then a small disruption might cause the handling of this barge to be delayed significantly. A terminal operator has to take this into account when making appointments.

A *nervousness avoiding* schedule means that the (baseline) schedule can avoid frequent and significant changes. Nervousness of a schedule becomes an issue when processes to prepare the barge handling have already started off, such as the stacking of containers at the quay side. A change in the quay assignment of a barge in the latter case, results in rework (containers have to be transported to another quay), a waste of resources, and is annoying for the people in the operations. We think that schedule nervousness in the hours prior to execution of a schedule should therefore be prevented as much as possible.

If terminal and barge operators decide to use a Multi-Agent system based on appointments, then terminal operators are restricted in their flexibility to deal with dynamic barge arrivals and other uncertain events (see the results in Section 7.2). Terminal operators might therefore tend to be less strict in keeping appointments to increase their planning flexibility. However, considering the importance of reliable appointments (as argued in Section 3.5.2), we stress that terminal operators should give high priority to keeping appointments. This implies that the terminal operator has to make appointments such that a certain level of solution robustness is achieved and that, e.g., small deviations in the call size of a barge, do not require a re-scheduling or cause violation of appointments with other barges.

A way for terminal operators to deal with uncertainty might be the following. Terminal operators fill a significant part of the day with appointments, but also leave each day a certain amount of time unplanned, the so-called slack time. This slack time can be distributed over the day. Terminal operators can use the slack time to recover from disturbances or to deal with unforeseen activities, such as rush orders. If the slack time is not necessary for these purposes on a specific day, then the terminal operator can use it to process barges earlier or to process barges that did not make an appointment and are waiting for an opportunity to be processed. The latter can be interesting for barges that experienced a delay in their rotation and were not able to keep their appointment with the terminal.

7.3.5 Robust rotations

We now consider the way barge operators can deal with disturbances that either happen frequently or have a limited impact. Also for a barge operator we think that there are several benefits of proactive (or robust) scheduling compared to (full) dynamic scheduling. Robust scheduling (including the creation of a baseline schedule or rotation) has as main advantage that it allows the barge operator to make (reliable) appointments. Reliable appointments have many advantages, since they allow for optimization of the operations (see Section 3.5.2).

When barge operators make appointments with terminal operators, they agree on a latest arrival time. This implies that they have to create a robust rotation to deal with uncertainties in, e.g., sailing times between terminals, sailing times to the port, or the departure time at terminals. Robustness for the barge operator concerns especially stability and flexibility. To create a stable rotation it might be necessary to include slack in between terminal visits to prevent an uncertainty at a terminal to disrupt the whole rotation. Flexibility is more difficult to realize, since appointments fixate to a large extent the rotation of a barge. If a barge departs delayed at one terminal and arrives delayed at another, then it can not simply right shift all its appointments at other terminals. The options it has are either to make a new appointment, or to wait at the terminal for an opportunity to be processed without having an appointment.

A possible repair strategy for a barge operator is to replan his rotation, e.g., by canceling some of the appointments with terminals and making new ones. However, replanning of appointments causes uncertainty in the operations of terminals, which is not desirable. Terminal operators might penalize this by keeping track of the reputation of a barge operator and taking this reputation into account when, e.g., issuing waiting profiles.

7.3.6 Dealing with large disruptions

In the previous section (Sections 7.3.2 to 7.3.5) we focused on disturbances that happen either frequently or have a small impact. These uncertainties are 'manageable' to a certain extent and can be dealt with using robust scheduling techniques. However, disturbances that happen infrequently and have a major impact on the operations are much harder to anticipate. For these kinds of disruptions terminal and barge operators need to have appropriate strategies.

Suppose a major disturbance occurs at one of the terminals, e.g., a crane breakdown. If this happens, then it is not likely that the terminal operator has included enough slack in his baseline schedule to guarantee a stable schedule for this disturbance. To deal with the disturbance, the terminal operator might have the strategy to cancel all appointments with barges for the duration of the disturbance. This implies that barge operators have to make a new appointment and possibly prefer to reconsider their whole rotation. Barge and terminal operators might use different strategies to deal with different kinds of disturbances.

A major disturbance is likely to have a great impact on the whole system, since a lot of actors are affected directly or indirectly. Dealing with a major disturbance is then not the problem of only a single actor but of all actors concerned. To deal properly with a disturbance, it is therefore important that all actors in the system are timely informed about a disturbance (communication) and possibly take over tasks of the actors that are affected by the disturbance (cooperation). The latter requires that actors look beyond their immediate interests and are willing to cooperate with competitors in case of a calamity.

Given the common interests of all actors, we suggest that the port authority plays a central role in managing a large disturbance. Prior to the disturbance, the port authority can make agreements with all the actors in the port on how to deal with certain disturbances, e.g., a crane breakdown. These agreements can be put down in a covenant, which is in force when an 'emergency' takes place. Let us give an example of an agreement. Suppose a crane breaks down and the associated terminal operator expects that the breakdown will last for several hours, then an agreement might be that the nearest terminal takes over part of the tasks. During the disturbance, the port authority can facilitate the communication of the disturbance to all the actors concerned and give suggestions on how to circumvent the disturbed parts of the system. This is similar to the task of the road authority, which informs car drivers about traffic jams and gives advice on alternative routes.

In this way, mechanisms are established to deal properly with a major disturbance and to minimize the disrupting effects on all parts of the system.

7.3.7 Strategic use of mechanisms to deal with uncertainty

In Section 1.1.1 we already mentioned that some actors nowadays exhibit strategic behavior to cope with uncertainty and to gain individual advantage. In the design of our Multi-Agent system we introduced self-regulating mechanisms to punish strategic behavior. However, disturbances and the way actors deal with uncertainty may give new opportunities for misuse of the system. Well-known examples of misuse in the (strategic) interaction of players are free riding and the 'tragedy of the commons' (see, e.g., Hardin, 1968; Gintis, 2000). In addition, in different fields (such as sociology, psychology, and economy) evidence is found that players not only respond rationally to the (strategic) actions of others, i.e., with only regard about their personal payoffs. Instead, people are sensitive for reciprocal behavior, i.e., also being led by the outcomes or intentions of other players. Reciprocal social values are, e.g., fairness, altruism, or revenge (see, a.o., Camerer, 1997; Falk and Fischbacher, 2006). Let us give two examples of (non)-intentional misuse of mechanisms to deal with uncertainty.

- If a terminal operator deals with a major disruption, he can cancel all appointments for a certain period of time to isolate the disturbance to only a few activities. However, the terminal operator might misuse this mechanism to recover from a poor planning, or to be able to process other barges or sea vessels.
- If terminals keep 20% of the time free from appointments to deal with uncertain events, they can possibly use this time also to process barges that were not able to keep their appointment due to an unexpected event. However, barges can misuse this flexibility, by trying to be processed during this 20% of slack time instead of making an appointment, especially if the latter results in more (yet guaranteed) waiting time. In this way fewer barges make an appointment and the 20% slack time might slightly increase to 60% or 80%. The system can then move towards a Multi-Agent system without appointments and deal with a lot of uncertainty again.

In addition to two examples of misuse, we also give an example of reciprocal behavior of actors when they are confronted with, e.g., free riders. Suppose there is one terminal operator that free rides in the system by not keeping his appointments. This terminal operator benefits from reliable arrival times of barges (due to non free riding other terminals), but creates a lot of uncertainty by letting barges depart later than their latest departure time. Other terminal operators may consider it as unfair that one terminal benefits from the actions of others. The risk is that these other terminal operators start to mimic the behavior of the free rider.

The risks we mentioned illustrate that the implementation of the Multi-Agent system is not easy, since it requires the commitment and discipline of actors to stick to the 'rules of the game'. We recommend to investigate these risks in future research.

7.3.8 Some preliminary tests

To get a feeling of the impact of uncertainty on the operations of barge and terminal operators, we do some preliminary tests. We simulate the effect of a blockage of the waterways leading to the port. A blockage implies that during the blockage the arrival intensity of barges reduces significantly and after the blockage increases significantly for some time.

The reason we choose to simulate a waterway blockage is that it is a major disturbance and in fact happened in March 2007 (Nieuwsblad Transport, 2007b).

Figure 7.16: Container dregded up from the river Rhine near Cologne. In March 2007 a barge lost 31 containers by making a turn on the river Rhine. The accident caused a blockage of shipping traffic for five days. Picture taken by the author on March 29, 2007

A barge lost 31 containers on the Rhine near Cologne. For five days no ships could pass the place of the accident. We visited the place ourselves on March 29, 2007 and we observed numerous barges at anchor in the Rhine waiting for the blockage to be resolved. Figure 7.16 presents a picture we took during our visit of the place of accident. Once the blockage was removed, the Port of Rotterdam was confronted with a high arrival intensity of barges for several days.

To simulate the effect of a waterway blockage we use scenarios from data set 0 and data set 4 (see Section 4.6). We consider scenarios from data set 0 to evaluate the effect of a disturbance in networks with different utilization degrees. We expect that the utilization degree of the network has a major impact, considering theory on queueing systems.

We use the Multi-Agent system developed in Chapter 6. We first consider data set 0 and focus on scenarios with network layout II, 9 terminals per region, and fixed time windows (see Section 4.6). During the generation of the scenarios we let barges arrive with an arrival intensity of one per time unit. However,

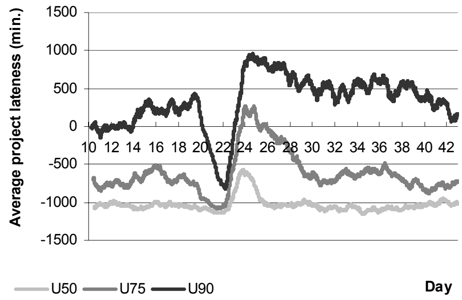


Figure 7.17: The average project lateness during the simulated horizon, assuming that a blockage of the waterway results in a reduced barge arrival intensity from day 20 to 21, and an increased barge arrival intensity on day 22 and 23. Results for scenarios from data set 0: network layout II, 9 terminals per region, 1 quay per terminal, and fixed time windows

on day 20 and 21 we reduce the arrival intensity of barges to 0.5, and on day 22 and 23 we increase the arrival intensity to 1.5 arrivals per time unit. We simulate all scenarios for 50 days, 10 replications per scenario, and we apply a warm-up and cool-down period of 10 and 3 days respectively.

In Figure 7.17 we depict the moving average of the average project lateness of successive barges during the simulated horizon. The results are specified for scenarios with 50, 75, and 90% utilization degree. From the picture we make the following observations. Batch arrivals of barges can clearly be considered as disturbances in the system. During the blockage we see a reduction and after the blockage an increase of the average project lateness. The impact of the blockage (and the resulting batch arrival of barges) depends on the utilization of the network, as we expect. The higher the utilization degree, the greater the impact and the more time it takes before the system has recovered from a disruption.

In data set 0 we consider a rather unrealistic situation. We therefore also simulate a scenario from data set 4. We apply the same arrival pattern as in data set 0. We again focus on a scenario with network layout II, 9 terminals per region, all regions in a line, and fixed time windows (see Section 4.6). We omit closing times of containers. We simulate 30 replications of 50 days each, and apply a warm-up and cool-down period of 10 and 3 days respectively.

In Figure 7.18 we depict the moving average of the average project lateness of successive barges during the simulated horizon. The results indicate that also in a more complex scenario similar effects can be observed. Note that terminals in data set 4 have different utilization degrees. The average utilization degree of all terminals in the data set is about 55%. This explains why the effect of

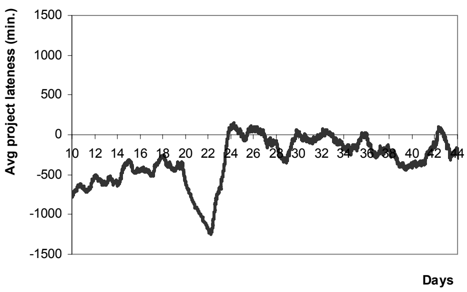


Figure 7.18: The average project lateness during the simulated horizon, assuming that a blockage of the waterway results in a reduced barge arrival intensity from day 20 to 21, and an increased barge arrival intensity on day 22 and 23. Results for a scenario from data set 4: network layout II, 9 terminals per region, and fixed time windows

the disturbance is not as strong as in scenarios with a 90% utilization degree in data set 0.

7.3.9 Dealing with uncertainty: conclusions

In practice terminal and barge operators deal with several uncertainties during their operations. These uncertainties are partly the result of the way the alignment process is currently organized, and partly inherent to the operations of barges and terminals. The former kind of uncertainties we try to reduce by introducing our Multi-Agent system, but the latter kind of uncertainties has to be dealt with within the system. In this section we made a first step in dealing with uncertainties in our Multi-Agent system by exploring the kinds of uncertainties terminal and barge operators deal with, discussing techniques provided in the literature to deal with uncertainties, and sharing some ideas about the way our Multi-Agent system and the operations of barge and terminal operators can be made more robust. We also discussed some drawbacks of dealing with uncertainty in our system and we did some preliminary tests.

Our Multi-Agent system has a lot of potential to reduce the uncertainties that are the result of a poor alignment process in practice nowadays. Appointments play a major role in realizing this reduction, since barge and terminal operators commit themselves to the appointment and thus reduce the uncertainty in each others operations. However, uncertainties in the current alignment process are sometimes caused by uncertainties that are inherent to the operations of barges and terminals. It is therefore important that barge and terminal operators can create robust plans, such that a certain level of uncertainty can be captured and that almost all of the appointments can be kept. We mentioned three forms of robustness (stability, flexibility, and nervousness avoidance) that can help terminal and barge operators to think about creating robust plans.

Disturbances differ in their frequency of occurrence and their impact. Barge and terminal operators can more easily anticipate uncertainties that happen frequently or have a small impact, than disturbances that happen occasionally and have a big impact. For the latter disturbances, terminal and barge operators need appropriate repair strategies. We suggest a central role for the port authority in managing a large disturbance, since it probably affects many actors in the port system.

As we have seen, using a Multi-Agent system based on appointments has many advantages for the port. However, the success of this system is determined by the attitude of each of the actors and the way (and extent to which) they can deal with uncertainties. If terminal and barge operators are not able to deal with a certain level of uncertainty and are thus not able to keep a large part of their appointments, then the reliability of appointments becomes questionable. It is likely that appointments are taken less and less seriously and that our Multi-Agent system slightly deteriorates to the current situation.

Chapter 8

Distributed planning in the Port of Rotterdam

8.1 Introduction

In the previous chapters we modeled a Multi-Agent system and evaluated its performance using fictitious data sets. In this chapter we develop and evaluate a realistic model of the Port of Rotterdam with respect to the barge handling problem. The aim is to evaluate the performance of our Multi-Agent system in the Port of Rotterdam. In this section we describe the way we model the realistic situation, our assumptions, and the outline of the chapter.

8.1.1 Modeling a realistic situation

In making a realistic model of the Port of Rotterdam we faced three difficulties. The first difficulty is the availability and reliability of data. It turns out that registrations of activities in the port are limited, ambiguous, incomplete, distributed, and sometimes even contradicting. The second difficulty is the extent of complexity and detail we can include in our model, e.g., weather influences. The third difficulty is that structural changes in the port take place frequently, which means that a model of the port today may not be valid anymore tomorrow.

These difficulties imply that i) it will be hard to validate our model, ii) certain *less realistic* assumptions will influence the results, and iii) a detailed model may have limited predictive value. 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.

In this chapter we have multiple goals. First, to develop a realistic model of the Port of Rotterdam. Second, to evaluate the Multi-Agent systems developed in Chapter 6 and Section 7.2. Third, to perform a sensitivity analysis to get insight in the performance of the Multi-Agent systems in different situations.

Considering the quality of the information we have at our disposal we do not aim to quantify the savings that can be realized using a Multi-Agent system. Instead we focus on the relative performance of our Multi-Agent system in different situations and in comparison with, e.g., the off-line benchmark, and we analyze which factors impact the performance of the Multi-Agent system in general.

8.1.2 Research setup

We have chosen for the following research setup. First, we model a base case, which is a realistic model of the Port of Rotterdam for the time frame 2006-2007. Second, we consider two scenarios to gain insight in the sensitivity of our model for different factors. We evaluate the base case and the scenarios with our Multi-Agent system (developed in Chapter 6 and Section 7.2) and with the off-line benchmark. We focus on the steady state performance of our models.

8.1.3 Assumptions

We make the following assumptions. Barges arrive over time with independent stochastic interarrival times. We assume that the arrival intensity of barges is the same every day, and every time unit of the day. On arrival in the port the barge operator decides which rotation a barge is going to execute. Decisions of both barges and terminals are made in real-time and we assume that two barge operators do not plan rotations simultaneously, but one after another. With respect to a terminal, we assume that it handles both barges and sea vessels. With respect to barges, terminals only have information about barges that arrived in the port. The service of a barge is not preempted for the service of another barge. Terminals have fixed capacity and can have restricted opening times, during which barges can be processed. Opening times of terminals are hard, i.e., no work is done in overtime. The time to handle a container, the mooring time, and the sailing time between terminals are deterministic. Sea vessels arrive with stochastic interarrival times at terminals and processing takes a stochastic amount of time. Arrival times of sea vessels are known to the terminal prior to the planning horizon. Sea vessels have absolute priority over barges.

With respect to a barge we assume that, on arrival in the port, it needs to make a decision about the sequence in which it visits the terminals concerned. On arrival in the port, the barge has information about the terminals it has to visit, the number of containers it has to (un)load at each terminal, and the mooring time at and the sailing time between terminals. It has no information about the state of the network, such as waiting times at terminals. We consider no capacity or stowage constraints for the barge. Barges visit terminals only once. We define closing of a terminal as *preemptive downtime*, and the processing of a sea vessel as *non-preemptive downtime*. Preemptive downtime means that the handling of a barge may start before the downtime and finish after it (Schutten, 1998). Non-preemptive downtime means that the handling of a barge might not be interrupted by a downtime. We do not consider disturbances and their effect on the operations of barges and terminals. These assumptions allow barge and terminal operator agents to make reliable appointments, since no unexpected delays occur. Introducing disturbances requires the introduction of a kind of penalty, e.g., penalty costs, if appointments are not met by one of the parties. This is part of future research.

In this chapter we do not consider closing times of containers, since this requires a lot of additional factors to be included in our model. The results in Chapter 6 indicate that closing times of containers can be included in the algorithms and we suggest to take closing times into consideration in a more detailed study into the Port of Rotterdam.

8.1.4 Outline of the chapter

In Section 8.2 we describe the data sources we consulted to collect realistic data. Section 8.3 describes the input and output of the realistic model. In Section 8.4 we describe the model parameters we use in the base case. Section 8.5 gives a description of the base case and the two scenarios we consider. In Section 8.6 we describe the results and insights obtained from our experiments. In Section 8.7 we conclude with a discussion of the results and in Section 8.8 we draw conclusions regarding the implications of our model in practice.

8.2 Data sources

To model the base case we use the following sources of information.

- Initi8 B.V. The company Initi8 B.V. (project partner within Transumo) provided information about the following aspects:
  - The utilization degrees of terminals.
  - The sailing times between terminals, taking into account whether a barge has to sail upstream or downstream.
  - The number of quays per terminal, i.e., dedicated barge quays, barge and sea vessel quays, and dedicated sea vessels quays.
  - The opening times of terminals.
  - The average and standard deviation of the barge call size at a terminal (in number of containers).

  - The mooring time of barges.
  - The average time to move a container with a terminal crane and the average number of calls in a barge rotation, specified for barges sailing to Antwerp, the Rhine areas, and Domestic areas.

  This information is partly based on historical data and partly on educated guesses.

- *Port of Rotterdam.* The Port of Rotterdam publishes several reports on the performance of the port, e.g., the number of transshipped containers. Besides they provided us with information about container sea vessels processed at the major terminals in the port in the period 2005-2007. Another source of information were interviews with experts working at the port authority.

- *Terminal company information.* We used annual report information of the ECT terminal and capacity information of the APM terminal, provided at their respective websites.

- *Other sources.* We also consulted several other sources of information, like master theses, several interviews, and web resources like Google Earth.

In Section 8.4 we describe how we use these data sources. The data we use is mainly based on the period 2006-2007. We like to stress again that it is hard to get realistic data describing objectively and unambiguously the relevant parameters of our model. This will change in the next couple of years as we expect. To give an example, in 2008 the Port of Rotterdam will start a project to equip several barges with transponders. The aim is to get objective registrations of sojourn times of barges at terminals. It is to be expected that in the future these kinds of projects are more widely introduced. Consequently, information will become more easily available and make the performance of barges and terminals more transparent.

8.3 Input and output of the realistic model

To develop a realistic model we use the approach given in Section 4.5. We describe the input of the model, the output of the model, and the results of the simulation successively. We use some of the outputs and results to validate the model.

For terminals we consider the following attributes as input: the number of quays, the opening times, the fraction of capacity assigned to sea vessels and barges, the utilization degree of the capacity assigned to barges or sea vessels, the distribution of the call size, the location, the sailing times to other terminals, and the time needed to move a container. For barges we consider as input: the interarrival distribution, the distribution of the rotation length, the distribution of the mooring time at a terminal, and the way the time window of a barge is determined. For sea vessels we consider as input: the interarrival distribution, the way sea vessels are assigned to terminal quays, and the distribution of the processing times.

For terminals we get the following output: the total volume of containers transshipped, and the number of barges and sea vessels processed. For barges we get as output: the arrival times of barges in the port, the time windows of barges, which terminals a barge has to visit, the number of barge arrivals per day, and the average utilization degree of barges. For sea vessels we get as output: the arrival times of sea vessels and which quay(s) of a terminal a sea vessel is assigned to.

After simulation of a realistic scenario we can derive, among others, the following results for terminals: the average waiting times of barges at terminals, the arrival times of barges at a terminal, and the distribution of the workload of a terminal over the day. For barges we can derive the average total waiting, handling, and sailing time in a rotation, the total sojourn time in the port, and the extent to which a barge leaves the port late.

8.4 Estimating the model parameters

In this section we describe the parameters of the base case. We model the base case analogously to the way we create a scenario, as described in Section 4.5. For determining the model parameters we have used the data sources mentioned in Section 8.2. The data obtained from these sources has undergone several adjustments to get a setting, which is internally consistent and consistent with different sources of information. We therefore validated the input and output parameters of our model with the different data sources, by checking the answers on, e.g., the following questions. First, is the total number of containers transshipped per year in our model reasonable compared with figures reported by the Port of Rotterdam? Second, is the capacity of terminals reasonable with respect to the number of containers transshipped? Third, is the call size at terminals reasonable and do barges have a realistic utilization degree? Fourth, do we get a realistic picture of the port or do we see anomalies, e.g., terminals that are not accessible by barges for more than a week because of sea vessels?

The figures we present in this section lead, to our opinion, to a reasonable approximation of the activities performed in the Port of Rotterdam and result in a setting which is consistent. In this section we describe some general observations. In addition, we describe successively characteristics of barges, terminals, and sea vessels, which are used as input for the realistic model.

Concerns	TEU	Containers
Transshipped in the port	9,653,232	5,846,433
Transshipped by ECT	5,500,000	3,300,000

Table 8.1: Containers transshipped in 2006. Figures about the port obtained from the Port of Rotterdam. Figures of the ECT obtained from the company's website

Area	Fraction of barges	Average # of calls
Antwerp	20%	6
Rhine	60%	7
Domestic	20%	6

Table 8.2: Characteristics of barges sailing to different hinterland areas

8.4.1 General observations

As starting point of our analysis we take the total number of containers transshipped in 2006, as published by the Port of Rotterdam, and the number of containers that the ECT terminals (according to the annual figures on their website) have transshipped in 2006 (see Table 8.1). The APM terminal, a large terminal on the Maasvlakte, does not publish these figures on their website, except for their capacity of 2.7 million TEU. Assuming that the APM terminal has a similar utilization degree as the ECT Delta terminal (about 85%), we see that the ECT and APM together handled approximately 80% of the total container volume in the port.

We note that the figures presented by the Port of Rotterdam concern only the containers that come from or go to the sea. Movements at, e.g., empty depot terminals are not counted in these statistics. The total number of containers loaded and unloaded is therefore larger than the numbers in Table 8.1. Although in January 2007 in total 33 container terminals were active in the port, the ECT terminals (seven in total) and the APM terminal certainly account for the major share. We therefore focus on these terminals in particular in determining the terminal characteristics.

8.4.2 Barge characteristics

In Section 2.5 we mention that the hinterland of barges can be divided in three markets (leaving out the inter-terminal transport): Antwerp, Rhine, and Domestic. Barges that sail to these areas have different characteristics. In Table 8.2 we denote per hinterland area, the fraction of barges sailing to these areas and the average number of calls in a rotation.

Unfortunately, we have no data about the distribution of the number of calls in a rotation. Based on expert opinions we assume a triangular distribution between 1 and 15, and a mean equal to 7. With respect to mooring time of a barge we assume a mean time of 10 minutes, uniformly distributed between 5 and 15 minutes. For convenience we stick to our concept of fixed and variable time windows, instead of using historical information about time windows of barges. To determine the time window of a barge we apply the same approach as in Section 4.6.2. We assume that barges arrive according to a Poisson process. The mean interarrival time depends on the number of barges arriving per time unit, which is depending on the total capacity in the network and the desired utilization degree of terminals (see also Section 4.5). We distinguish two port entrance and exit points, namely the Van Brienenoord-bridge and the Spijkenisse-bridge (see Figure 2.2). We assume that all barges sailing to Rhine and Domestic markets enter and leave the port via the Van Brienenoord-bridge. With respect to barges going to or coming from Antwerp, we assume that 50% of the barges leave and enter the port via the Spijkenisse-bridge and the other 50% via de Van Brienenoord-bridge. We assume that the arrival intensity of barges is the same every day and every time unit of the day.

8.4.3 Terminal characteristics

In this section we describe the characteristics of terminals. We stress that the numbers we provide in this section are *our* estimation and might deviate from practice. Table 8.3 presents all the terminal characteristics we use in the base case. We describe the columns of the Tables 8.3 successively.

Table 8.3 first denotes for every terminal, the region where the terminal is located. This is needed for the calculation of the time window of a barge. We use the following abbreviations: *city* for the city terminals, *botlek* for the botlek terminals, and *mv* for the terminals located at the Maasvlakte. We refer to Figure 2.2 for a picture of the port with the respective regions.

With respect to the terminal capacity we distinguish three types of quays, namely i) dedicated barge, ii) barge and sea vessel, iii) dedicated sea vessel quays. For convenience we call the barge and sea vessel quays also *duo quays*. A quay is a combination of resources that are all necessary to handle a barge, namely a berth, crane(s), and a team (see Section 5.4.1). The capacity of a quay varies depending on the kind of vessel that is processed. We assume that both sea vessels and barges are processed at one quay, but in case of a sea vessel the terminal deploys, e.g., four cranes and for a barge one crane.

To determine how much capacity of the duo quays is available for barges, we performed some calculations based on the total number of containers transshipped by the ECT Delta and APM terminals and the total number of dedicated sea vessel quays of these terminals. We derived that approximately 50% of the duo quay capacity is needed to process sea vessels, assuming a time to move a container of three minutes and a utilization degree of these quays of 85%. However, this is a rough estimate which can vary per terminal and per period.

Terminal	Region	#quays barges - duo - sea vessel	Opening times	Avg call size	St. dev. call size	Util. degree
C1	city	1-0-0	0-24h	30	5	50%
C2	city	1-0-0	8-16h	8	4	50%
C3	city	1-0-0	8-16h	8	4	50%
C4	city	1-0-0	8.5-15h	8	4	50%
C5	city	1-0-0	8 -15h	8	4	50%
C6	city	1-0-0	8-16h	8	4	50%
C7	city	1-0-0	8-16h	15	3	60%
C8	city	0-1-0	8.5-16h	15	3	60%
C9	city	1-0-0	8.5-15h	15	4	60%
C10	city	1-0-0	8.5-15h	15	4	60%
C11	city	0-1-0	8.5-15h	15	4	60%
C12	city	1-1-0	6-21h	15	4	60%
C13	city	1-0-0	7-21h	8	4	70%
C14	city	0-2-0	7.5-23h	8	4	70%
C15	city	0-5-0	7.5-16h	8	4	70%
C16	city	2-0-0	8.5-15h	8	4	70%
C17	city	0-1-0	8.5-15h	8	4	70%
C18	city	1-0-0	7-21h	8	4	70%
C19	city	0-2-0	7.5-22h	15	3	80%
C20	city	1-0-0	7.5-22h	8	4	80%
C21	city	1-2-0	0-24h	23	6	85%
B1	botlek	1-0-0	8-17h	8	4	50%
B2	botlek	1-0-0	8-16h	15	4	60%
B3	botlek	0-1-0	8.5-16h	8	4	70%
B4	botlek	0-1-0	8.5-15h	23	3	70%
B5	botlek	2-0-0	6-21h	8	4	80%
M1	mv	0-2-0	6-24h	15	4	60%
M2	mv	1-0-0	0-24h	8	2	60%
M3	mv	1-3-1	0-24h	30	5	85%
M4	mv	0-1-3	0-24h	30	5	85%
M5	mv	0-2-1	0-24h	38	7	85%
M6	mv	0-1-2	0-24h	23	7	85%
M7	mv	3-0-0	0-24h	45	5	85%

Table 8.3: Terminals and their attributes

The presented opening times are the opening times of terminals at January 1., 2007. We assume that these opening times are the same every day in a week. In the absence of data about the distribution of the call sizes at terminals, we assume a beta-distribution with a minimum of 1 and a maximum of 60 containers. We assume that the handling time of barges is equal to three minutes per container move, assuming that a crane can perform 20 moves per hour. In practice the handling of barges is done at different speeds, varying from 2-20 (or more) moves per hour. This has a significant impact on the time a barge sojourns at the terminal.

We estimated the utilization degrees given in Table 8.3. We remark that individual terminals might calculate the utilization degree in a different way. We adopt the definition as given in Section 4.5.2. In our model, as in reality, we use sailing times between terminals which are different for upstream and downstream movements.

Sea vessel characteristics

With respect to sea vessel characteristics we have analyzed data of more than 23,000 container sea vessels that visited the major container terminals in the Port of Rotterdam in the period 2005-2007. The data concerns the arrival time of a vessel, the terminal it visited, and the sojourn time at the terminal. We have aggregated the arrivals of all sea vessels and analyzed the interarrival times for this aggregated set of terminals and the sojourn at terminals. We assume that the sojourn time of a sea vessel at the terminal is equal to its processing time. Figure 8.1 depicts a histogram of the interarrival times and Figure 8.2 a histogram of the processing times.

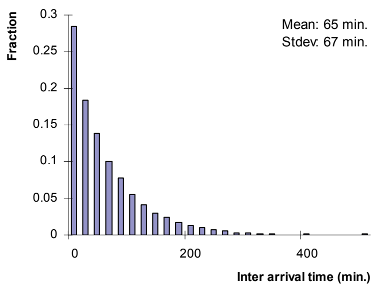


Figure 8.1: Histogram of the interarrival times aggregated for the main terminals in the port of Rotterdam

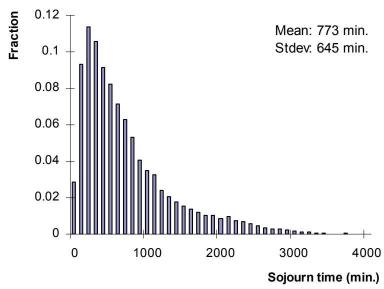


Figure 8.2: Histogram of the sojourn times aggregated for the main terminals in the port of Rotterdam

Considering Figure 8.1 we think that an exponential distribution fits well the interarrival times of container ships at port level. These findings are in line with Kuo et al. (2006). They have done a case study into interarrival times of container ships in international ports in Taiwan. They analyzed the arrivals of container sea vessels on port, terminal, and single berth level. They found that an Erlang K (with k=1) distribution fits well the interarrival times of container ships at port and terminal level. However, at single berth level the interarrival times can better be described by an Erlang K (with k = 1 or 2) distribution. The latter is also interesting in our case. If we assume Poisson arrivals of sea vessels at a single quay then a next sea vessel B might arrive during the processing of sea vessel A. Sea vessel B has to wait at sea until the processing of vessel A is completed, which means that the resulting interarrival times of the vessels at quay level are not exponentially distributed anymore. Moreover, if we assume Poisson arrivals for sea vessels and we delay the arrival of a sea vessel if it arrives during the processing of another vessel, it might happen that a quay is occupied for many days up to a week. Interviews with experts revealed that terminals are occasionally occupied by sea vessels for more than three to four days in a row. It happens sometimes in busy periods.

Poisson arrivals at a single quay are also not very likely since terminal operators can plan sea vessels over the available quays. The actions of the terminal operator therefore undermine the Poisson arrival process. The more quays that are operated by the same terminal operator, the more flexible this operator can plan the sea vessels. Moreover, the way terminal operators plan sea vessels differs per terminal. We therefore adopt the following strategy to get a realistic distribution of sea vessels over the quays of terminals. At departure of a sea vessel we determine the arrival time of the next sea vessel drawn from an exponential distribution. The mean of the distribution is equal to the mean interarrival time of the sea vessels minus the average processing time of the sea vessels. To give an illustration. Suppose we expect 20 sea vessel arrivals in 100 days, then the mean interarrival time is equal to 100/20=5 days. If the average processing time of a sea vessel is one day, then the mean of our new distribution is equal to four days. i.e., if the processing of one sea vessel is completed, then the time until the next sea vessel arrival is exponentially distributed with a mean of 4 days. In this way we prevent clustering of sea vessels at a quay for an unrealistic amount of time, but it still can happen that sea vessels arrive soon after another.

We prefer to use a theoretical distribution for the processing times of sea vessels. The reasons why are that: i) we use data that is aggregated for all terminals in the port, ii) the data seems to fit well with known distributions, and iii) by using an empirical distribution we pretend a precision which does not correspond with other parameters in our model. We therefore compared the sojourn time data of the sea vessels with theoretical distributions, such as log-normal, beta, and gamma, to determine a suitable distribution to approximate the processing time of sea vessels. We have chosen to apply a beta-distribution for two reasons. First, the beta distribution fits the data quite well except for larger values (see Figure 8.3 for a Q-Q plot). Second, there is a clear upper bound

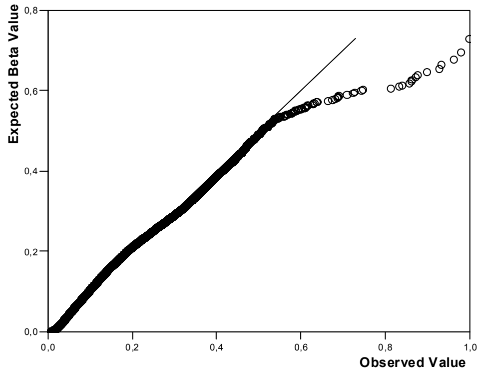


Figure 8.3: A Q-Q-plot of the processing times aggregated for all terminals compared with the beta distribution with α_1 = 1.14 and α_2 = 8.3

to the number of containers sea vessels can load and unload due to capacity restrictions. Distributions with an infinitely long tail (such as the log normal distribution) have to be cut-off to prevent call sizes that exceed the sea vessel capacity. Moreover, it is well-known that a lot of stochastic models are not very sensitive for more than the first two moments of the underlying probability distribution (see, e.g., Tijms, 2003), unless one is very interested in the tails of the distributions.

The total number of sea vessels is derived from the available capacity for sea vessels and the desired utilization degree of terminals. We refer to Section 4.5 for a detailed description of how a scenario is created.

8.5 Description of the base case and the scenarios

In the experiments we consider a basic scenario (called base case) and two other scenarios (Scenarios 1 and 2), which are variations on the base case. The base case is used to obtain insight in the performance of the Multi-Agent based control (developed in the previous chapters) in the Port of Rotterdam. The two scenarios are used for sensitivity analysis to investigate the effects of certain factors on the performance of barges and terminals. These scenarios are based on ideas that are studied in practice (see also, Section 2.5.3).

We simulate the Multi-Agent system, we developed in Chapter 6, to evaluate its performance in the base case and the two scenarios. We consider three levels of information exchange, namely i) no information, ii) yes/no, and iii) service-time profiles. In the remainder of the chapter we refer to the ‘no information’ level of information exchange as 'No Info (appoint)', since no information is exchanged and barges make appointments with terminals. Besides, we simulate the Multi-Agent model presented in Section 7.2 with decision rule 1. We refer to this as 'No Info (FCFS)', since no information is exchanged and terminals process barges first-come first-served (FCFS). 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.

In the base case and the scenarios, we determine the time window of a barge analogously to Section 4.6.2. 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 based on the base case. We do not consider closing times on containers. The objective of the barges is to minimize the sojourn time in the port.

In the next sections we describe the base case and the two scenarios.

8.5.1 Base case

The base case is constructed based on the experimental settings described in Section 8.4. We make the following choices. We make no distinction in the average number of calls in a rotation for barges with different hinterland destinations. The reason is that we do not have enough evidence to justify this.

In our simulation we omit the dedicated sea vessel quays since they do not affect the processing of barges. With respect to the duo quays (barge and sea vessel) we assume that 50% of the quay capacity is available for sea vessels with a utilization degree of 85%. We assume that the opening times of terminals, mentioned in Table 8.3 do not hold for sea vessels, i.e., sea vessels can be processed 24h a day. The processing time of a sea vessel is beta distributed with a mean processing time of 770 minutes, a standard deviation of 645 minutes, a minimum processing time of zero, and a maximum processing time of 6400 minutes.

The number of barges is determined analogously to Section 4.5. For the base case it means that a number of about 48 barges visit the port daily. The average utilization degree of the barges is 80% (assuming an average barge capacity of 120 TEU). The total number of container moves per year is about 2 million, 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.

For the simulation of the service-time profiles we apply slack policy 1 of Table 8.4. In addition to the Multi-Agent control we also evaluate the performance of

Utilization degree ρ of a terminal	**Policy 1**	**Policy 2**
$\rho \leq 50\%$	0	30
$50\% < \rho \leq 75\%$	30	60
$\rho > 75\%$	60	90

Table 8.4: Different slack policies. Slack denoted in minutes for different utilization degrees of terminals

the off-line benchmark. We use the off-line benchmark as described in Section 4.4. For 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.

8.5.2 Scenario 1

In both practice and theory, ideas have been mentioned to reduce the number of terminals a barge visits. An example is the introduction of a large hub, far in the hinterland, e.g., in Duisburg (De Binnenvaartkrant, 2008). The idea is that containers are put unsorted on large barges and are sailed to the hinterland where the containers are sorted and shipped to their hinterland destination, either by truck or barge. The expected effect of this development is a reduction of the average number of calls in a rotation and an increase of the average call sizes at terminals (to maintain a certain utilization degree of barges).

In Scenario 1 we therefore consider the effect of reducing the average number of calls in a rotation and increasing the call sizes at terminals. In the next sections we give our motivation for choosing this scenario and we describe the way we model the scenario.

Motivation for choosing Scenario 1

The reason why we consider Scenario 1 is that we expect that a reduction of the average number of calls in a rotation, not necessarily results in a reduction of the total waiting time in a rotation. Indeed, we expect that waiting time at a busy terminal can not be used anymore to visit other terminals. Moreover, the corresponding increase of call sizes might lead to an increase of waiting times at terminals and can reduce the possibility to process barges in between, e.g., planned sea vessels. Finally, the performance of sophisticated methods (such as issuing service-time profiles) might be less valuable when the number of sequences in which terminals are visited is limited to, e.g., at most 3! possible rotations.

Description of the scenario

Scenario 1 actually consists of two scenarios, namely Scenario 1a and 1b. In both Scenarios 1a and 1b we reduce the average number of calls in a rotation and we increase the average call size at the terminals to maintain an 80%

utilization degree of the barges. These two adjustments imply that a barge spends less time on mooring during its stay in the port, since it visits only a few terminals to handle the same amount of containers. This is also attractive for terminals, as they can use the time saved with mooring to transship containers. Regarding the modeling of Scenario 1 we now have two options: i) keeping the utilization degree of terminals the same as in the base case (resulting in a higher throughput of containers), or ii) keeping the throughput of containers the same as in the base case (resulting in a reduced utilization degree of terminals). We investigate both options in Scenarios 1a and 1b respectively. Let us take a closer look at the parameters of the scenarios.

The number of calls in a rotation, both in Scenario 1a and Scenario 1b, is triangularly distributed with a minimum of one, a maximum of three, and a mode equal to two. To maintain an 80% utilization degree of the barges we increased the average and standard deviation of the call size at every terminal with a factor four.

In Scenario 1a we keep the utilization degree of the terminals the same as in the base case. This results in about 57 barge arrivals per day. The total number of container moves per year is about 2.5 million, about 1.2 million more than reported by the Port of Rotterdam. The reason we can handle about 500,000 moves more than in the base case is that mooring time now is used for the transshipment of containers.

In Scenario 1b we keep the throughput the same as in the base case. We therefore reduce the average utilization degree of terminals with 15%. The resulting number of barge arrivals per day is about 45. The total number of container moves per year is similar to the base case.

The aim of both scenarios is to get insight in the effect of a reduced number of terminal visits and increased call sizes, on the sojourn time of barges in the port and on the performance of different control methods.

For the simulation of the service-time profiles we apply both slack policies from Table 8.4 for Scenario 1a, and slack policy 1 for Scenario 1b.

8.5.3 Scenario 2

The second scenario is based on another idea, namely to realize a hinterland hub about 50 kilometers from the port (a so-called *transferium*) to transfer containers from truck to barge and vice versa. The aim is to relieve the congested roads leading to the sea terminals and to perform logistic activities in less congested areas (Konings, 2007; De Binnenvaartkrant, 2008). The expected effect of such a transferium is an increased number of barges in the port. For terminals this implies that truck handling operations will reduce and that barge handling operations will increase.

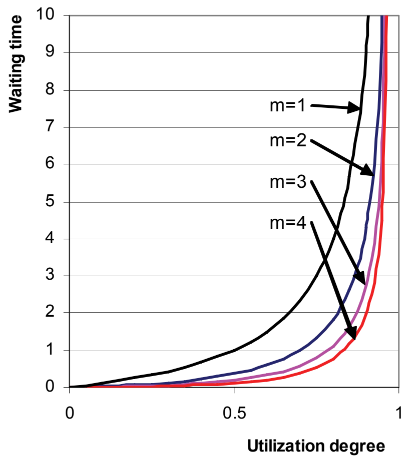


Figure 8.4: Example of the relation between the utilization degree of a terminal and the average waiting time (in time units). The example is based on an $M/M/m$ queue, with $m \in \{1, 2, 3, 4\}$

In the next sections we give our motivation for choosing this scenario and we describe they way we model the scenario.

Motivation for choosing Scenario 2

The reason why we choose this scenario is that barges already face long waiting times at terminals today. This is due to both a poor alignment of activities and a high utilization degree of terminals. If terminals have to process additional barges, the utilization of the quays will even grow. The question is what the effects of this growth are on the sojourn time of barges in the port.

Based on insights from queueing theory we expect that this extra flow of barges may result in a significant increase of waiting time at terminals. We illustrate this with an example. In Figure 8.4 we depict the relationship between the utilization degree of a terminal (with $m \in \{1, 2, 3, 4\}$ quays) and the average waiting time at this terminal. The picture is based on the assumption that barges are processed first-come first-served, arrive according to a Poisson arrival process, and have exponentially distributed processing times. This is a so-called $M/M/m$ queue (see, e.g., Gross and Harris, 1998). It is clear that an increased utilization of the terminal corresponds with growing waiting times, especially when the terminal has a limited number of quays. Moreover, Hopp and Spearman (2000) argue that the waiting time increases even faster if the variability in the arrival process or the processing times of barges increases.

Description of the scenario

In Scenario 2 we increase the utilization degree of terminals with 20%, compared to the base case, with a maximum of 95%. All other settings are similar to the base case, including the average call size. The aim is to investigate the effect of an increased utilization of the terminals on the sojourn times of barges in the port and the waiting times at terminals. We evaluate the performance of different levels of information exchange. In this scenario the number of barges arriving daily in the port is about 59 (an increase of 23% compared to the base case). The total number of container moves per year is about 2.4 million, about 1.1 million more than reported by the Port of Rotterdam. The reason we can handle about 400,000 moves more than in the base case is due to the increased utilization of terminals.

For the simulation of the service-time profiles we apply slack policy 1 from Table 8.4.

8.6 Results and insights

In this section we describe and analyze the results of the simulations and the off-line benchmark for the base case and the two scenarios. In Section 8.6.1 we describe the results regarding the base case. Section 8.6.2 and Section 8.6.3 describe the results of scenario 1 and 2 respectively. Finally, we compare in Section 8.6.4 the performance of the Multi-Agent system for the base case and the different scenarios.

8.6.1 Base case

In this section we look at the results of the base case. Figures 8.5 and 8.6 depict the average project tardiness and project lateness of different Multi-Agent controls. The results show that service-time profiles perform significantly better than the other Multi-Agent based controls. We also find that 'No Info (FCFS)' outperforms 'No Info (appoint)' as we would expect based on the results in Section 7.2. The results of 'No Info (FCFS)' might even be improved if more sophisticated decision rules are applied.

The average sojourn time of barges with a rotation length of seven terminals is about 36 hours (in case of service-time profiles). This consists of about 9 hours handling, 9 hours sailing, and 18 hours waiting. The waiting time boils down to about 2.6 hours per terminal on average. The sojourn time of a barge in case of 'No Info (appoint)' is about 69 hours, consisting of 7 hours sailing, 9 hours handling, and 54 hours 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. In Section 8.6.4 we discuss this in more detail.

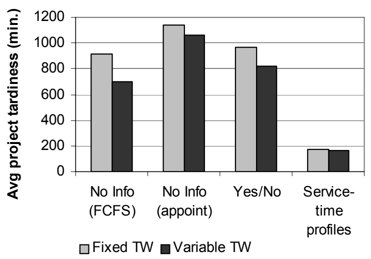


Figure 8.5: Average project tardiness for different Multi-Agent controls in the base case

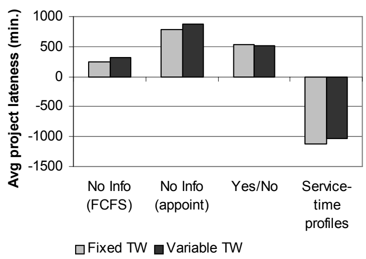


Figure 8.6: Average project lateness for different Multi-Agent controls in the base case

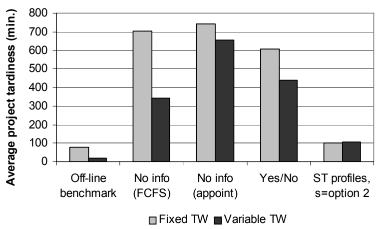


Figure 8.7: Average project tardiness for different Multi-Agent controls compared with the off-line benchmark. Results for the base case

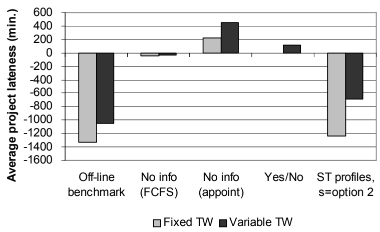


Figure 8.8: Average project lateness for different Multi-Agent controls compared with the off-line benchmark. Results for the base case

If we make a comparison with the off-line benchmark we get the results presented in Figures 8.7 and 8.8. We find that the Multi-Agent system with service-time profiles performs well compared to the off-line benchmark. Moreover, we find the same patterns as we see in Figures 8.5 and 8.6. To give an impression of the computational burden: every bar representing the off-line benchmark in the Figures 8.7 and 8.8 account for about 240 hours of computation time.

In Figure 8.9 we depict the arrival patterns of barges during the day for different terminals. In Figure 8.9 we see, e.g., that terminal C19 (closed from 10 p.m. to 7.30 a.m.) clearly has more barge arrivals during the day than in the night. Terminal C21, however, is opened 24h 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 24h 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 24h terminals during the night.

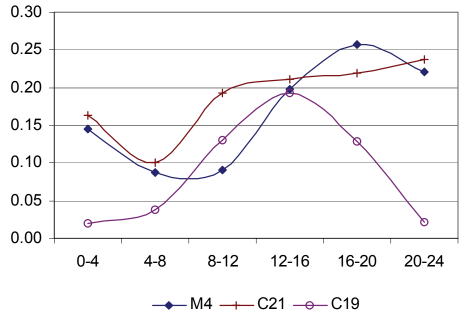


Figure 8.9: Fraction of barges arriving at terminals M4, C21, and C19 during certain time intervals of the day

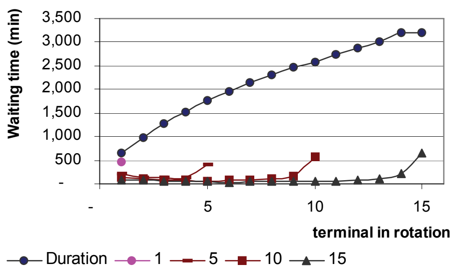


Figure 8.10: The avg waiting time at a terminal measured at the time a barge plans its rotation, as function of the position of the terminal in a rotation with a specific length (1, 5, 10, and 15). Results for the base case and service-time profiles

Now that we have seen the arrival pattern of barges at different terminals, let us consider the waiting times at terminals when barges plan their rotation. We are interested in the correlation between the waiting times at terminals and the order in which a barge visits the terminals concerned. Consider a barge with a specific rotation length. In order to plan a rotation, the barge requests waiting profiles at the terminals that have to be visited. At the time these waiting profiles are issued to the barge, we (as researchers) collect these waiting profiles and extract from these the waiting time at each terminal at that very moment. Then we consider the rotation a barge has planned based on these waiting profiles and we link the observed waiting time at each terminal to the position of that terminal in the rotation. We average the observed waiting time at the n^{th} terminal in the rotation of all barges with the same rotation length. In Figure 8.10 we depict the average waiting time at the n^{th} terminal in the rotation, at the time a barge plans its rotation. For comparison, we also depict the average duration of a rotation for different rotation lengths. Note that the x-axis for the average duration denotes the rotation length, whereas for the other lines the x-axis denotes the n^{th} terminal visited in the rotation. The results indicate that terminals with long waiting times -as perceived at the very moment of planning- are usually visited at the end of the rotation. In general, however, most terminals do not have long waiting times at the time a barge plans its rotation, which means that they can be visited rather easily in the short run.

8.6.2 Scenario 1

In Scenario 1a we reduced the average and maximum length of the rotation compared to the base case. The utilization degree of terminals in this scenario is the same as in the base case. In Figures 8.11 and 8.12 we depict the average project tardiness and project lateness for different Multi-Agent controls. The results reveal that the service-time profiles perform significantly better than the other controls, however, the relative difference with other controls is much smaller than in the base case. This result indicates that the added value of service-time profiles is less when barges have shorter average rotation lengths. If we vary the slack policy in the service-time profile communication, we find that slack policy 2 (Table 8.4) leads to a slight improvement of the results compared to slack policy 1, but the difference is small.

In Scenario 1b we also reduced the average and maximum length of the rotation compared to the base case. However, we maintain the same throughput as in the base case by reducing the average utilization degree of terminals. In Figures 8.13 and 8.14 we compare the average project tardiness and project lateness in Scenario 1a and 1b, in case we use service-time profiles. It is clear that the reduced utilization degree of terminals leads to a significant improvement of the average project tardiness and project lateness of barges. This is in accordance with results in Chapter 5.

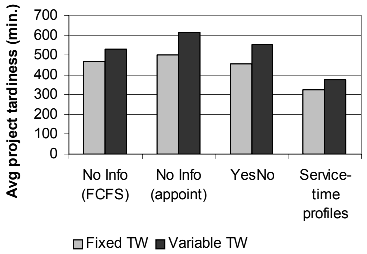


Figure 8.11: Average project tardiness for different Multi-Agent controls in Scenario 1a

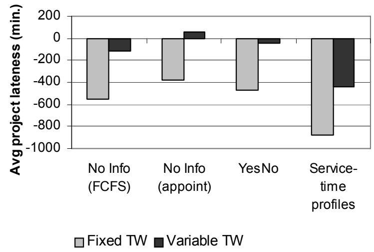


Figure 8.12: Average project lateness for different Multi-Agent controls in Scenario 1a

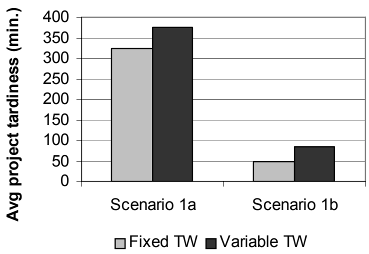


Figure 8.13: Average project tardiness in Scenario 1a and 1b, specified for fixed and variable time windows

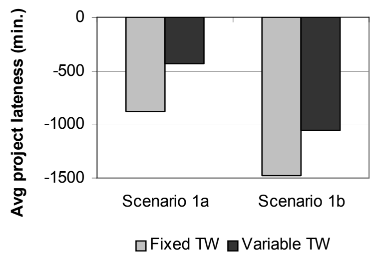


Figure 8.14: Average project lateness in Scenario 1a and 1b, specified for fixed and variable time windows

Note that in Scenario 1b we realize the same throughput as in the base case. This indicates that mooring time (which is part of the utilization of terminals in our model) leads to a significant waste of capacity at terminals.

8.6.3 Scenario 2

In Scenario 2 we increased the utilization degree of terminals with 20%, with a maximum of 95%. In Figures 8.15 and 8.16 we depict the average project tardiness and project lateness for different Multi-Agent controls. We do not depict the results for the yes/no information exchange, since the simulation effort was too extensive and we expect that the results are in the same order of magnitude of figures we have presented before.

The results indicate that the average project tardiness has increased significantly compared to the base case. This means that an increase in the utilization degree of terminals leads to a more than proportional increase in the sojourn time of barges due to waiting time at terminals. Also now we see that service-time profiles outperform the other controls.

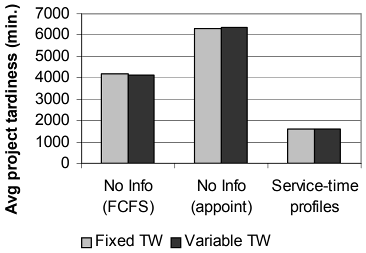


Figure 8.15: The average project tardiness for different Multi-Agent controls in Scenario 2

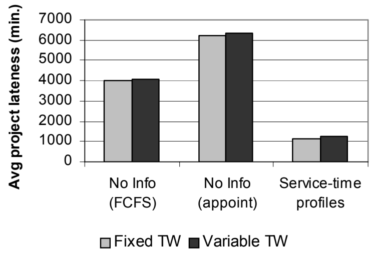


Figure 8.16: The average project lateness for different Multi-Agent controls in Scenario 2

8.6.4 Comparison of the base case and the scenarios

In this section we take a closer look at the results and make a comparison between the base case and the scenarios. We start with a comparison of the sojourn time of an average rotation in the base case and each of the scenarios. In Figure 8.17 we depict the share of waiting, sailing, and handling time in the rotation. Above each of the bars in Figure 8.17 we denote the duration of the rotation in hours.

From Figure 8.17 we can derive several insights. First, the difference between the average rotation length in Scenario 1a and the base case is only due to sailing time. This means that a reduction of the average rotation length not necessarily results in a reduction of the barge sojourn time. However, when we reduce the utilization degree of terminals such that we get the same throughput in the base case, we find that the average sojourn time reduces significantly, i.e., Scenario 1b. This is mainly due to a reduction of average waiting time. Second, an increase of the average terminal utilization (Scenario 2) leads to a significant increase of waiting time in the rotation. The share of the other components (handling and sailing time) in Scenario 2 is more or less equal to the base case.

In Figure 8.18 we depict the average barge's sojourn time for different rotation lengths in the base case and each of the scenarios. The figure shows that the results found in Figure 8.17 are consistent also for other rotation lengths. In Figure 8.19 we show the duration of a rotation in the base case for different rotation lengths and different Multi-Agent controls. Also here we find similar results as in previous chapters, namely that the average port sojourn time of barges decreases in the length of the rotation, in case barges and terminals make appointments. This relation is linear if we process barge FCFS.

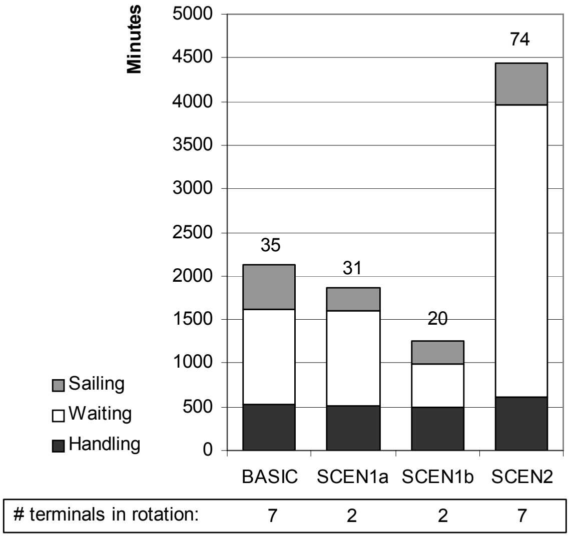


Figure 8.17: The duration of an average rotation in the base case and the scenarios. The duration of the rotation is specified for sailing, handling, and waiting time

8.7 Discussion of the results

As part of the validation of our model we discuss the average sojourn time of a barge in the base case. An average barge in the base case has a sojourn time of about 36 hours in case we apply a Multi-Agent system with service-time profiles. Whether this is realistic is hard to say. Consider the following numbers.

- Connekt (2003) reports an average port sojourn time of 22.5 hours, consisting of 7.5 hours handling and 15 hours for sailing and waiting. Although not explicitly mentioned, we assume that these numbers are based on the study B-RIL in 1998 (RIL Foundation, 1998).
- Konings (2007) reports sojourn times varying from 30-36h. During these 30-36h about 150 TEU are loaded and unloaded on average, with a total handling time of about 12h. These numbers are probably based on a study in 2003.
- In 2007 several press releases appeared reporting increasing waiting times varying from 24 to 72 hours, mainly due to congestion at the major terminals (see, e.g., Nieuwsblad Transport, 2007a).
- The CBRB (an organization for employers and entrepreneurs in logistics

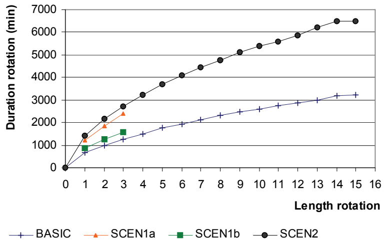


Figure 8.18: The duration of a rotation in the base case and the scenarios for different rotation lengths

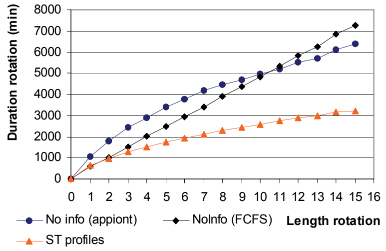


Figure 8.19: The duration of the rotation in the base case, specified for different Multi-Agent controls

and inland navigation, cf. Section 1.1.2) reports since 2007, in cooperation with several barge operators, the so-called *haven verblijfindex* (the port sojourn index). The index denotes the average amount of time a container barge sojourned in the port per move, i.e., per container transshipment. The average 'haven verblijfindex' in the first half year of 2008 was about 8 minutes. If we assume an average barge capacity of 120 TEU and a barge utilization degree of 80%, then an average barge transships in the port 192 TEU, i.e., about 116 containers. This means that the sojourn time of an average barge is about 928 minutes (116 containers * 8 minutes), which is about 15.5 hours.

The different sources of information result in different values for the average port sojourn time of a barge. Considering the numbers we just mentioned, we think that a sojourn time of 36 hours in the base case (for a barge with a rotation length of seven terminals) might be relatively high compared to the current situation. Especially when we take into account that in practice the alignment of activities is not done efficiently. This can imply several things. First, it could be that the terminal utilization is not as high as we assume it is for all terminals. Lower utilization degrees lead to shorter waiting times and shorter sojourn times. Second, it is likely that barges on average are utilized less than 80%. This means that they have fewer container transshipments which impacts the handling and waiting time at terminals. Third, in practice terminals process barges both according to appointments and FCFS. This hybrid system probably functions better than 'No Info (appoint)' as reported in Section 8.6.1.

8.8 Conclusions

In the previous chapters we evaluated the performance of our Multi-Agent systems for fictitious data sets. In this chapter we considered the Port of Rotterdam to get insight in the performance of our Multi-Agent system in a realistic setting.

We faced several difficulties in making a realistic model of the port. The first difficulty is the availability and reliability of data. The second difficulty is the degree of complexity and detail we can include in our model. The third difficulty the period of validity of a realistic model. These difficulties imply that i) it is hard to validate the model, ii) certain *less realistic* assumptions influence the results, and iii) a detailed model has limited predictive value. Nevertheless, we think that we managed to develop a realistic model of the Port of Rotterdam and provide valuable insights in the performance of our Multi-Agent system. However, we did not aim to quantify the impact of implementing a Multi-Agent system on, e.g., the operations of terminals and barges or the economical impact on the port. Instead, we focused on the relative performance of our Multi-Agent system in different situations and compared with the off-line benchmark, and we analyzed which factors impact the performance of the Multi-Agent system in general.

In this chapter we made a realistic model of the Port of Rotterdam for the time frame 2006-2007. We called this model the base case. In addition, we developed two scenarios (based on the base case) by which we performed sensitivity analysis. In the scenarios we varied factors of which we think (based on results in Chapter 5 and Chapter 6), that they have a major impact on the performance of terminals and barges.

The first scenario (Scenario 1) is based on an idea to ship containers with large barges to a transferium far in the hinterland (say Duisburg), where the containers can be distributed to their respective hinterland destinations. The effect on the barge handling process will be a reduced average rotation length of barges and increased call sizes at terminals. The second scenario (Scenario 2) is based on an idea to build a transferium close to the port, to transfer containers from truck to barge and vice versa. The aim is to relieve the congested roads leading to the sea terminals. The expected effect is an increasing number of barge handling activities at the terminals. We described the parameters of the base case and the scenarios, and the data we used. We evaluated the performance of the Multi-Agent systems developed in Chapter 6 and Section 7.2 and the off-line benchmark for different the situations.

The results for the base case indicate that a Multi-Agent system with service-time profiles performs well compared to the off-line benchmark, and outperforms the other Multi-Agent controls. Since it is not clear which way barges are planned currently (probably a combination of appointments and FCFS processing), we cannot say what the expected improvement is of a Multi-Agent system compared to the current situation in terms of waiting times at terminals or the port sojourn of barges. The comparison with the off-line benchmark indicates that a Multi-Agent system is promising in the port.

The idea to build a large transferium far in the hinterland can be interesting (Scenario 1), especially when the average rotation lengths of barges decrease. Terminals then waste less capacity during the mooring of barges, and the throughput of the port can be increased without an increase of terminal utilization. To reduce the average rotation length, it will be necessary to organize the barge container transport differently, such that barges only collect containers for a limited number of sea terminals. The question is how easily this can be realized in the current market.

Regarding the idea to build a transferium close the port (Scenario 2), we expect that the increase in barge movements will lead to a significant increase of waiting at the terminal, unless terminals increase the capacity available for barges. A similar concern we raise regarding the 2nd Maasvlakte. If terminals do no reserve enough capacity for barges and aim for a reasonable utilization degree (say $\leq 85\%$), then the waiting times of barges might be significant. Increased waiting times worsen the competitive position of the barge modality and will also determine the success of a transferium. We wonder whether this effect is currently taken into account.

It is hard to give an indication of the economical benefits of a Multi-Agent system based on our results, but we can think of several direct and indirect benefits. If terminals and barges make reliable appointments, then barge operators do not need to stow their barges very flexibly anymore. This allows for an increased utilization of barges. Moreover, shorter port sojourn times allow for more round trips to the hinterland with the same barge. Terminal operators, on the other hand, can probably realize a reduction of the operational costs, since quays can be used more efficiently and quay operations can be better aligned with other activities at the terminal. More indirect benefits for both terminals and barges are an improved reliability of barge container transportation, shorter transit times, and lower costs. For the Port of Rotterdam this is an important competitive advantage. However, also from a societal perspective, a modal shift in favor of barges is attractive when this leads to a less congested road infrastructure and less environmental damage.

It would be interesting to perform our study again, when better data about the Port of Rotterdam becomes available in the future. Another option is to perform a field test to see how our model works out in practice and which benefits can be observed. In the next chapter we already make a first step towards implementation in practice by developing a game, as a means to communicate our model in practice.

Chapter 9

The use of a management game

9.1 Introduction

In Chapter 3 we argue that in the design of a Multi-Agent system not only optimization but also acceptance is important. Suppose we are able to design a Multi-Agent system that optimizes the operations of each of the actors. However, the system requires that actors have to participate on conditions they do not accept. Then, although the system is very efficient, it will probably never be implemented and therefore certainly not be effective.

We think that acceptance of a Multi-Agent system is determined by several aspects. We mention the following. First, in the design of the Multi-Agent system we already have to take into account specific business constraints. These business constraints define the set of feasible solutions (see Chapter 3). Second, actors might be reluctant to adopt a solution which functioning they do not fully understand or trust. Third, in our problem there are possibly more than 40 independent actors involved. To make the system successful, a critical mass of actors needs to be in favor of the solution. If not, it will be difficult to get the system running successfully. Suppose only a few terminals and barge operators adopt the Multi-Agent system (less than the critical mass), then a fraction of the activities can be planned by means of the system and the rest must be done separately. A Multi-Agent solution in this situation is probably not efficient.

To facilitate the acceptance process of our Multi-Agent system we have chosen to develop a game in which actors can experience different alternative solutions. In this chapter[1] we describe the game and our first experiences in workshops with practitioners and students.

[1] This chapter is based on the paper: A.M. Douma, J. van Hillegersberg, and P.C. Schuur (2008). Using a management game to exemplify a Multi-Agent approach for the barge

The outline of this chapter is as follows. In Section 9.2 we describe related literature and explain our aims with developing a game. In Section 9.3 we provide a description of the game. Section 9.4 describes our first experiences in different workshops we organized and in Section 9.5 we draw some conclusions and we mention directions for further research.

9.2 Why developing a game

In this section we explain why we develop a game to communicate our research to practitioners. In Section 9.2.1 we describe relevant literature on the use and design of serious games. Section 9.2.2 describes the aims we try to realize by means of a management game.

9.2.1 Literature review

Games and simulations are strongly related in the field of serious gaming. Both provide an environment that facilitates learning or the acquisition of skills (Angelides and Paul, 1993). Simulations mimic the behavior of a set of variables of a 'real' situation that are considered to be important, whereas games stimulate competition between participants within a predefined set of rules (Ellington et al., 1982; Ryan, 2000). Angelides and Paul (1999) state that gaming-simulation is a sequential decision making exercise through which players can experience the consequences of their decisions rapidly in an artificial environment containing some characteristics of a real situation. The aim is to enhance a comprehensive understanding of a complex system and to develop learning skills.

Although a wide range of literature has appeared on teaching effectiveness of simulations and experiential exercises, there is hardly any objective evidence to conclude that simulations and experiential exercises indeed result in learning (Anderson et al., 1998; Gosen and Washbush, 2004). It is unclear whether the perceived learning of players also means actual learning (see for further reading Gosen and Washbush (2004)). On the other hand there are many authors claiming that games do contribute, e.g., to increase the understanding of complex situations. Wenzler and Chartier (1999) state that the properties of the parts can only be understood in the dynamics of the whole. Games and simulations are a mechanism to show the big picture (Gestalt understanding) in a condensed period of time. Reducing the learning period of people from real time to simulated time, allows for steeper learning curves. Hoogewegen et al. (2006) give an example of how games can contribute in understanding dynamic business networks and the effect of different strategies. Another interesting contribution comes from Barreteau et al. (2001), stating that games are

rotation and quay scheduling problem in the Port of Rotterdam, *in* T. Blecker, W. Kersten, and C. Gertz (eds.), *Management in Logistics Networks and Nodes*, Erich Schmidt Verlag, Berlin.

good at explaining the content of Multi-Agent systems. Especially for negotiation support purposes it might be necessary to give the actors insight in the functioning of the system and discuss whether the model's assumptions match their own representation of the system dynamics and whether agents have a right range of possible actions.

Ryan (2000) claims that changing management practices require learning skills that can be met through games. He states that problems and issues become increasingly interrelated and that, as more people become involved in decision making, priorities become less clear and implementation gets more difficult. His experience is that simulation games are extremely useful in developing an appreciation of systems thinking, i.e., seeing the consequences as the product of the interaction of the parts. Ryan (2000) states that simulation games create common experience among participants, which they can refer to discussing the system concepts (see also Wenzler and Chartier, 1999; Le Bars and Le Grusse, 2008). Le Bars and Le Grusse (2008) share their experience with a simulation game and a decision support system for (collective) decision making among several actors. They show that this combination improves discussions between stakeholders and facilitates the emergence of acceptable solutions. They mention that acceptability of the solution in this case is more important than the optimal solution. A game-like setting has also been applied in an earlier study of our problem (Moonen et al., 2007). They report that through a game the participants start to realize the seriousness of the problem and the potential benefits of an agent based system.

The brief literature overview shows that games are considered to have potential to increase the understanding of complex systems and support the discussion between stakeholders based on common experiences. In this chapter we describe a game and our first experiences with the use of the game as a means to explain the functioning of a Multi-Agent system.

9.2.2 The aims of the game

The aims of the game we developed are fourfold. The first aim is to support the transition of our solution to people in practice. In fact, we can also explain by means of a presentation the concept of waiting profiles, the interaction of different distributed decision makers, and the way the Multi-Agent system functions. However, we expect that experiencing the solution helps people to understand the working of the system, to see which information agents can use for their decision, and to imagine the way agents reason. This contributes, as we expect, to a better judgment of, e.g., the possible sensitivity of exchanged information.

The second aim is to use the game as practical validation of our solution. Are people, e.g., able to make better (or more efficient) decisions if they are provided with more information? How do people use the provided information and what does it mean for the decisions they make?

The third aim is to evaluate the design of the user interface and the way information is presented to the player. Finally, we aim to initiate a pointed discussion between actors (barge and terminal operators). Through the game they can experience the effect of different interaction protocols and discuss whether certain solutions are acceptable or not.

9.3 Game description

In this section we describe the game by addressing the game setting and typology, the task and the aim of the player, the course of the game, different scenarios we can play, the game architecture, the user interface, the specific choices we made to get a balance between complexity and simplicity, and the performance indicators.

9.3.1 Game setting and typology

To simplify the game setting we take the Multi-Agent system and the port settings of Chapter 5 as starting point. The game setting is a port with a variable number of terminals. In the port barges arrive at specific points in time to visit a number (not necessarily all) of the terminals to load and unload containers. The boats arrive in the port via a single waterway.

We position our game in the following way along the dimensions of the typology of Angelides and Paul (1999). First, we build a role-playing game, i.e., we focus on a particular position within a system, namely the barge operator. Second, we build a specific game, i.e., industry specific. Third, we build a functional game focusing on the planning discipline within (terminal and) barge operator companies.

9.3.2 Task and aim of the player

A player in the game is considered to be a barge operator (of one barge). The task of the player is to determine a rotation, i.e., a sequence in which the barge visits the terminals. The aim is to leave the port as soon as possible. To do so, the player can communicate with the corresponding terminals to retrieve information about convenient handling times. Multiple players plan their barges simultaneously, thus influencing each others possibilities for specific handling times.

9.3.3 Course of the game

At the start of the game all barges are positioned at the entrance of the port. Every player gets an assignment, i.e., a number of terminals he has to visit. Players get different but equivalent assignments. The game leader gives the players, at the start of the game, a notification that they can start to plan their rotation. From that moment on (game) time proceeds with a certain (configurable) speed.

To plan a rotation barges can propose specific time slots to terminals. Terminals can accept or reject a request depending on whether the terminal is available to process the barge. How terminals reply to barges depends on the scenario played. We discuss this in the next section.

After the player has determined a first terminal to head to, his barge starts to sail to this terminal, say terminal A. If the player decides to head to terminal B instead of A, the barge changes direction and sails to terminal B. If the barge arrives at the terminal prior to the planned time slot, it queues and waits for processing. Once the barge is processed it heads to the next planned terminal and queues on arrival, unless the player has decided in the mean time to sail to another terminal.

After the barge has been processed at all the assigned terminals, it heads back to the entrance of the port. When the barge arrives there, the player has completed his tasks and his arrival time is logged. The player which barge returns first at the entrance of the port is winner.

During the game the player gets a map of the port, including the position of his own barge, the barges of other players, and all the terminals. In addition to the barges of players, we can also introduce dummy barges. Dummy barges are barges that are operated by the computer. Dummy barges limit the possibilities for players to plan specific rotations. In this way we can make the game setting more complex, since the actions of dummy barges also affect the actions of the players.

9.3.4 Different scenarios

In the game players play a living agent. One of the aims of the game is to let players experience different communication protocols. We consider the following four scenarios, varying along two dimensions. First, how barges are processed at terminals, either first-come first-served or based on appointments. Second, whether players get insight in the queue lengths of a terminal (waiting information).

Scenario 1: First-come first-served, no waiting information
If barges are processed first-come first-served, the player in fact only determines the sequence in which terminals are visited. The player has no information

about the availability of the terminals. No appointments with terminals are made and barges are processed in the order they arrive at the terminal.

Scenario 2: First-come first-served, waiting information
This scenario is similar to scenario 1, but now players get insight in the queue length on arrival at the terminal.

Scenario 3: Appointments, no waiting information
If terminals process barges based on appointments, players really have to make appointments with terminals. The player can send a request for a time slot to a terminal and the terminal replies with yes (the time slot turns green) or no (the time slot turns red) if the time slot is available or not, respectively. If a barge arrives later than a planned time slot, the time slot is cancelled and a new appointment has to be made before the terminal will process the barge. Players get no waiting information. This could mean that the player has to communicate repeatedly with terminals to determine convenient handling times.

Scenario 4: Appointments, waiting information
This scenario is similar to scenario 3, but now players get waiting profiles from the terminals they have to visit. This information gives the player insight in the occupation of the terminals during the day. The waiting profiles are updated dynamically during the game as the result of actions of other players.

If terminals process barges based on appointments (scenarios 3 and 4), they apply the following policy:

- If a terminal agrees with a request for a certain start slot, the barge is added to the schedule of the terminal. This means that all barges are processed in the sequence they are scheduled (unless a barge is not present).
- If a barge is in the schedule of a terminal but does not arrive in time, it is removed from the schedule and has to make a new appointment. The player gets a notification.
- If the terminal is idle in the next time slot, it considers its queue and starts the next planned barge which is waiting in the queue.

Terminals reply to barges with a certain (configurable) random delay (several seconds). In this way we make it less interesting to find a convenient time slot just by sending many requests. The processing of requests constitutes a burden for the terminals and in practice the terminal operator might need some time to send a reply.

In the game we can play one of the four scenarios mentioned, or play a combination of scenarios. We can let one part of the group play scenario 1 and the other part scenario 2, or one part scenario 3 and the other part scenario 4.

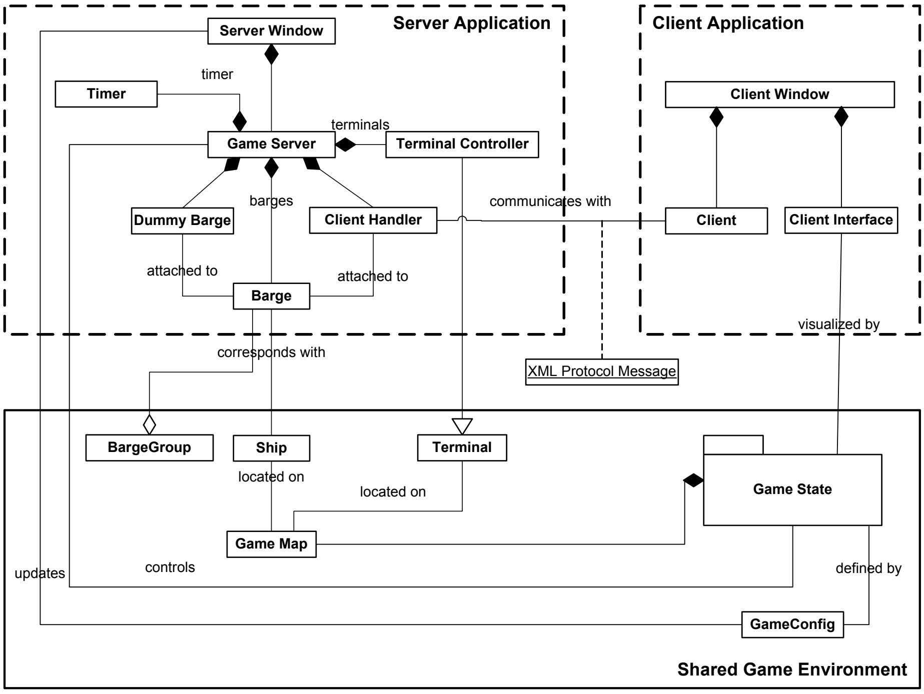


Figure 9.1: UML static structure diagram of the game

9.3.5 Game architecture

The game architecture is depicted in a UML static structure diagram in Figure 9.1. The game consists of three parts; the server application, the client application, and a shared game environment.

The client application is a client window, consisting of a client class and a client interface class. The client interface class takes care of the graphical representation of the game state on the screen of the player. The client class handles all actions of the player and the communication with the server.

The server application is a game server consisting of different classes, namely a server window, a timer, a terminal controller, a client handler, and a dummy barge class. The server window class handles all the actions of the game leader regarding the initialization of the game, interventions in the game, and depicts detailed information about the state of the game. The terminal controller is a child of the terminal class and maintains the schedules of the terminals and the processing of barges. The client handler maintains the communication with the clients. The barge class concerns the barges of all the players. The dummy barge class is meant to control the dummy barges. The game server takes care

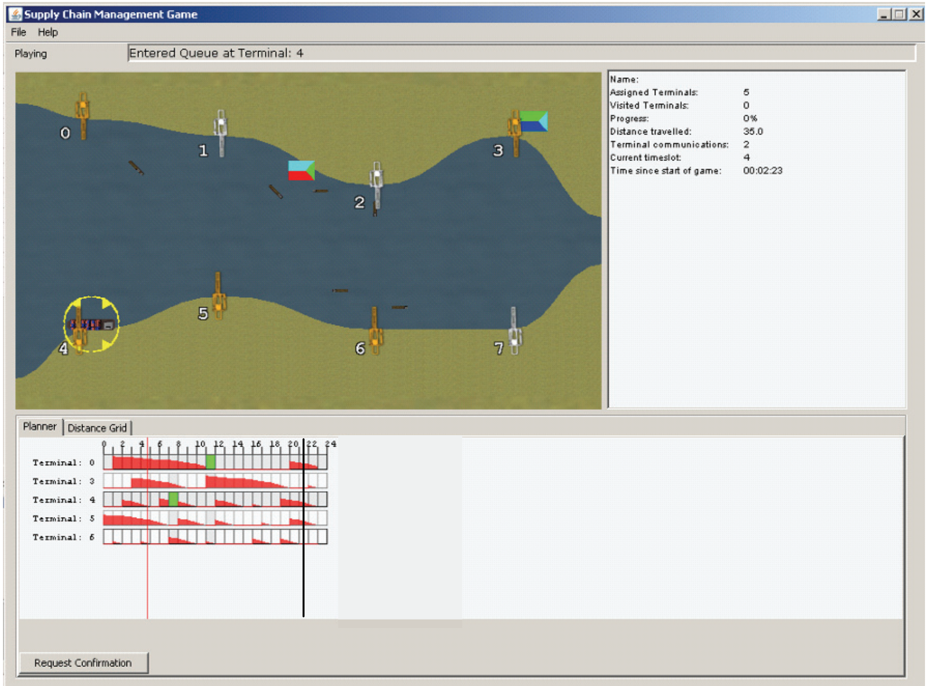


Figure 9.2: Screen shot of the user interface

of the communication between the classes and controls the game state.

The shared game environment package consists of several classes, namely a barge group, ship, terminal, game map, game state, and game configuration class. The game configuration class is updated on initialization of the server.

9.3.6 User interface

The user interface consists of three parts (see Figure 9.2). The upper left part is a graphical representation of the game state, depicting the location of all the barges and terminals. The upper right part of the screen presents the performance of the barge. The lower part is reserved to plan a rotation. In this part of the screen a time line is given for every terminal the player has to visit. This time line is divided in time slots. The player can select a time slot by clicking on the specific time slot. By pushing the button on the left bottom corner of the screen, the selected time slots are communicated with the terminals. As long as the player has not pushed this button, the plan is just a proposal made by the player but not communicated with the server (and the terminals).

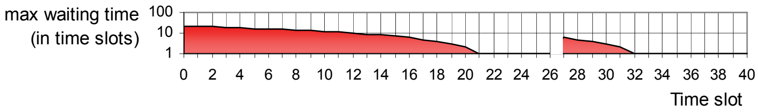


Figure 9.3: Example of a waiting profile depicted in the time bar of a terminal. The maximum waiting time is depicted on a logarithmic scale

The red bar in the lower part of the screen denotes the progress of time. If the processing of a barge is finished, the corresponding time slot turns blue. Refused time slots turn red and approved time slots green. Proposed (and not communicated) time slots are depicted as grey.

In the time bars we can depict waiting profiles as shown in Figure 9.3. Waiting profiles are depicted on a logarithmic scale and show the number of time slots a barge has to wait before it can be processed.

9.3.7 Specific game choices

We have made several choices to create a game that is not too complicated to understand and not too simple to make it of no value for the players (see also Barreteau et al. (2001) for similar considerations).

Every player in the game plans a single barge. Terminals are controlled by the computer, to be sure to have a timely and equal response to players. The time horizon is divided in discrete time slots. The processing of a barge at every terminal takes one time slot. Sailing times between terminals are determined based on the shortest path between two terminals, taking into account the course of the shore line. The sailing speed is equal for all barges. Every barge arrives and leaves the port at the same point.

The length of the simulated horizon is configurable. Terminals only have one quay and one place where barges can queue. Terminals do not process sea vessels, but can be closed for barges for a random time. During the game the player can see his position in the port, the position of the terminals, and the other players.

The following parameters can be configured on initialization of the game.

- The layout of the network
- The total number of terminals
- Length of the planning horizon
- The number of time slots in the planning horizon
- The number of players
- The number of terminals the players have to visit
- Specific user id's

- Speed of the simulation
- Sailing speed of the barges
- Delay in the response of the terminal after a request of the barge
- The number of dummy barges and the corresponding arrival times in the port
- Which scenario is played (scenario 1, 2, 3, or 4, or certain combinations)

The user id's are denoted by real flags which are handed out at the start of the game. In this way players can see which flag (and barge) corresponds with which player.

On initialization of the game we can select the fraction of time slots that are randomly closed at terminals. In this way we can influence the available time slots and thus the difficulty to find an optimal (or even feasible) rotation.

In the waiting profile scenario we omit the slack, which plays an important role in Chapters 5 and 6. The result is that waiting profiles reduces to time slot information. The reason we have chosen to omit the slack is to make the waiting profiles not too complicated for players. Waiting profiles without slack already give an impression how waiting information can be used by players and how competitively sensitive it possibly is.

9.3.8 Evaluation of the game

To evaluate the performance of players in different scenarios we can both use objective and subjective measures. The objective measures are the number of times communication with terminals was necessary to plan a rotation, and the sojourn time of the barge in the port. The subjective measures regard the perceived usefulness of the waiting profiles, the perceived ease of use, the attitude towards adoption, and the quality of the concept. We also ask participants to reflect on the perceived usefulness and the ease of use of the game itself and the user interface.

9.4 First experiences

In this section we describe our first experiences with the game. We have played it three times with groups of practitioners, academics, and students. The group of participants in the first workshop was quite diverse and consisted of barge and terminal operators, but also consultants, and scientists. The group in the second workshop consisted of Master students in Industrial Engineering, Business Information Technology, and Computer Science, and of Ph.D. students. The group in the third workshop consisted mostly of practitioners and academics. We describe our experiences with the three groups successively.

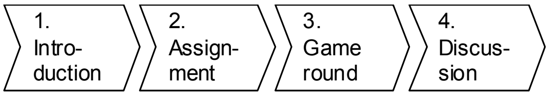


Figure 9.4: Steps in workshop 1

9.4.1 The first workshop

We discuss first the setup of the workshop and then the workshop experiences.

Setup of the workshop

The setup of the first workshop is depicted in Figure 9.4. The workshop started with a presentation introducing the problem and the solution we developed. We especially focused on the agent's communication protocol and the information exchanged. After the introduction we gave all the participants an exercise on paper with the assignment to plan the shortest rotation along eight terminals (see Appendix B). For every terminal we provided a waiting profile, indicating when a barge can be processed. We also gave the sailing times between terminals. The participants got a few minutes to do the assignment. This turned out to be quite difficult and since it took them about 5 minutes to find a solution, they realized that in practice waiting profiles already could have changed. This makes the problem even harder.

The assignment serves as introduction to the game. The aim of the game was to give the players a feeling of the dynamics in the planning problem. There was not time enough to play several rounds. After the game we briefly discussed the experiences with the participants. In the workshop about 18 people participated on 10 computers (clients).

Experiences

The participants reacted enthusiastically on the game. None of the participants had difficulty to operate the user interface or to interpret the information on the interface. For participants it became clear that the actions of other players influenced directly the possibilities they had to plan their rotation. After playing the game we got several responses of practitioners. Some people, e.g., still wondered why a central solution is not possible in our problem. We also got detailed feedback from people working for a barge operator. They first indicated that our game omits several practical aspects that complicate the problem, e.g., the stowage plan of barges which restricts the sequence in which terminals can be visited. They were, however, enthusiastic about our solution, but expected that it can be hard to implement the Multi-Agent system. The reason is that a Multi-Agent system requires different behavior from the actors, e.g., to make reliable appointments or to reveal specific information. The question is whether actors are willing to adopt the Multi-Agent based solution.

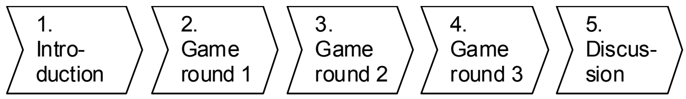


Figure 9.5: Steps in workshop 2

They expect that if it is possible to implement the system, significant benefits can be expected for their operations, such as an increased vessel utilization and more round trips from barges to and from the hinterland.

9.4.2 The second workshop

We discuss first the setup of the workshop and then the workshop experiences.

Setup of the workshop

The second workshop (with students) was set up differently from the workshop with practitioners to focus mainly on the game (see Figure 9.5). We started the workshop with a brief introduction of the problem and the game. Then we played the game in three rounds. The first round was a test round. The second and third round were serious rounds. In all rounds, players had to make appointments with terminals. In round 2, half of the players got waiting profiles, whereas the other half lacked this information. In round 3, we reversed this so that all players experienced both the value of waiting profiles and having no waiting profiles.

After playing the game we started a discussion related to the concepts we tried to explain and the game itself. In the workshop about 10 people participate on 10 computers (clients).

Experiences

The experience with the student group was that they were able to operate the user interface easily and had no difficulty to interpret the information presented. Students could easily work with waiting profiles. They indicated that waiting profile information made the planning of a rotation easier, especially when the utilization of the network increases. Analysis of the behavior of the players in different rounds revealed that players adopt different strategies to make plans for their barge. Some students make a lot of proposals and changes in their plan, whereas others plan once and update only a few times. From the game rounds it also became clear that if the network is lowly utilized, it does not really matter how much information someone has at his disposal. This is in accordance with findings in our simulation study. The higher the utilization degree of the network and the more terminals a player has to visit, the larger the benefits of having waiting profiles.

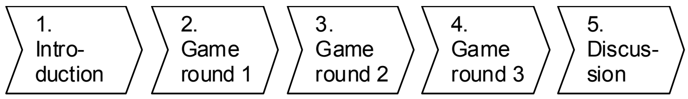


Figure 9.6: Steps in workshop 3

The students indicate that the game was fun to play and increased their understanding of the dynamics in planning problems among different (self-interested) actors. All students perceived the game as useful for their understanding and the explanation of the developed concepts. They had some difficulty to see the task of the agent. They did not directly realize that they actually played a living agent.

9.4.3 The third workshop

We discuss first the setup of the workshop and then the workshop experiences.

Setup of the workshop

The third workshop (with practitioners and academics) was started with a brief introduction (see Figure 9.6). In this introduction we explained the problem, the game (the task of the player, when someone is winner etcetera), and the scenario we were going to play in game round 1 (namely scenario 2 in Section 9.3.4). The first game round was also meant to give users the possibility to get some experience with playing the game. In the second and third round we played, after a brief introduction, scenarios 3 and 4 respectively.

After playing the three game rounds we closed the workshop with a brief discussion of the experiences. We played the game with about 20 players on 18 computers (clients).

Experiences

During the workshop we were struck by some annoying bugs. The bugs sometimes prevented a player to complete his rotation. Despite that, all players played (part of) the three scenarios. Most of the participants could intuitively work with the game interface after our brief introduction. However, some bugs resulted in questions why certain things did not happen. It indicates that the expectations of the players are in accordance with the supposed functioning of the game.

After playing the three rounds, the players indicated that scenario 2 was rather easy to play compared to the other scenarios. The reason why is that players do not have much information at their disposal and simply sail along the terminals where they are processed FCFS. The lack (or limited availability) of information about queues at the terminals means that this information cannot be taken into account in the decision, which makes the decision easier in that respect. Scenario 3 was the most difficult to play according to the participants. The players that planned a rotation soon after the game started were mostly successful in getting a convenient slot at a terminal. However, players that were a bit later faced a lot of occupied slots and repeatedly had to send requests to terminals. They considered this as rather inconvenient. Scenario 4 was more easy to play than scenario 3, although people still had difficulty to find a good rotation. The reason why is that waiting profiles change repeatedly because of the actions of other players. Moreover, the fact that players had relatively much information about the availability of terminals, made it more difficult to decide quickly on a convenient rotation in scenario 2, where this information was lacking.

After playing the three game rounds, we discussed the experiences of the game with the players. From the discussion is became clear the players not always understand the role of an agent in our game. They did not realize that they were playing a living agent. One of the participants, employed by one of the major containers terminals in Rotterdam, explained that his terminal already makes appointments with barges. An appointment means that a barge has to be present at the terminal before a certain time and then receives a guaranteed latest departure time from the terminal. These appointments, however, are mostly cyclic appointments, e.g., every week or every couple of days. They would be interested in a system in which similar appointments can be made more dynamically. The participant also suggested to add some slack to the appointments, to create some planning flexibility. The Multi-Agent system we propose in Chapter 6 has incorporated already these suggestions.

9.5 Conclusions and further research

In this chapter we presented a multi-player game of a Multi-Agent system for the barge handling problem. In the game, players act as living agents and they have to solve the problems that agents have to solve in the Multi-Agent system. In this way they can experience the different solutions we developed and create an understanding of the functioning of the system. The aim of the game is to facilitate the acceptance process of the Multi-Agent system.

The first experiences with the game are encouraging and promising. In three workshops participants indicate that the game increased their understanding of our solution. In the game they experience the dynamics in the problem and the complexity of aligning activities between multiple self-interested parties. Through the game we were able to start pointed discussions with participants about, e.g., their attitude towards the solution, the difficulties they expect in the implementation, the extent to which waiting profiles reveal sensitive information, whether the solution is acceptable, etcetera.

Although the first experiences with the game are promising, we cannot yet conclude that all our aims can be achieved. To get more experience we plan to organize yet another workshop with practitioners, especially barge and terminal operators. With respect to the further development of the game, we aim to introduce also the notion of agents. An interesting question is how to give players an impression of the task of an agent and how to design the communication between an agent and a player. The game can function in this way as prototyping technique for the development of the user interface of the Multi-Agent system. Another interesting option is the possibility for players to increase or decrease the sailing speed of their barges. An increase of the sailing speed of a barge with a few kilometers per hour, can double the fuel consumption. The corresponding fuel costs can then be taken into account in the determination of the winner. In a later stage we aim to let the terminal also be played by a player instead of the computer. This gives terminal operators the option to reflect on the way they plan their quays and the issues a terminal operator agent should take into account when it responds to barges.

Chapter 10

Summary, conclusions, and further research

10.1 Summary and conclusions

In this thesis we considered the barge handling problem, concerning the alignment of container barge and terminal operations in a port. Barges are used to transport containers from the port to the hinterland, and vice versa. Every time a barge arrives in the port, it visits several terminals to load and unload containers. The sequence in which a barge visits these terminals depends, among others, on the availability of terminals. The availability of terminals, in turn, is depending on the other barges and the sea vessels that have to be processed. Throughout our thesis we used the Port of Rotterdam as our case of reference, although our model is applicable to general multi-terminal, multi-barge settings.

The barge handling problem is not easy to solve due to several complicating factors mentioned in Section 1.1.3. Previous attempts to provide a solution to the barge handling problem made clear that a centralized solution is not acceptable for the actors concerned, and that a decentralized planning approach may be one of the few ways to solve the problem. The problem is highly relevant in practice, e.g., in the Port of Rotterdam. The inefficiencies, resulting from a poor alignment of barge and terminal operations, lead to significant (in)direct costs and also affect the attractiveness of the Port of Rotterdam as a node in global container transportation chains. We formulated the goal of our research as follows:

The aim of this study is to develop and evaluate an efficient and effective distributed planning system for the barge handling problem -concerning the alignment of container barge and terminal operations in a port-, and to gain insight in the way the proposed system functions.

In this thesis we developed a Multi-Agent system based on service-time profiles. The general conclusion we draw is that our Multi-Agent system is a promising approach for tackling the barge handling problem and that it may lead to a significant improvement in practice. In the following sections we discuss this conclusion and the insights we obtained by answering the research questions presented in Section 1.4.

10.1.1 Research Question 1

The first research question is: *What is the role of barge hinterland container transportation in The Netherlands and as part of worldwide container transportation?*

Over the last decade, container flows have increased rapidly worldwide. The Port of Rotterdam plays a major role in global container transportation. In 2007, the port was the sixth largest in the world and the largest in Europe with respect to the number of transshipped containers. The hinterland transportation of containers related to the Port of Rotterdam is performed by barge, rail, and truck. In 2007, about 36% of all containers going to or coming from the hinterland were shipped by barge and this fraction will probably grow in the coming years. The reasons why are: i) other modalities (road and rail) get congested or have reached the limits of their capacity, whereas barge container transportation has still room for growth, ii) the increasing attention for environmental issues and the rising oil prices increase the attractiveness of barge container transportation, and iii) the development and exploitation of the 2^{nd} Maasvlakte will lead to additional container volumes, since the Port of Rotterdam has made agreements with the terminal operators exploiting the 2^{nd} Maasvlakte about the maximum fraction of containers that are transported by truck.

Nowadays, several developments take place in global container transportation. Liner shipping companies play a central role in these developments. They try to improve their performance by choosing ports that allow for low cost, reliable, and fast container transshipment. In addition, they increasingly try to get control over the hinterland transportation chain. Also the barge container sector is rapidly developing. Several developments take place or are expected to take place, such as strategic alliances, increase in scale, and the automation of container handling. In addition, new hinterland transportation concepts are considered, such as the development of large container transferia outside the port.

Low cost, fast, and reliable hinterland connections are of vital importance for a port such as the Port of Rotterdam. However, increasing container volumes lead to problems such as long waiting times of barges at terminals, and require the development of new concepts to structurally improve barge hinterland container transportation.

10.1.2 Research Question 2

The second research question is: *What are key performance indicators of barge operators, terminal operators, and the Port of Rotterdam?*

The key performance indicator of a barge operator is to minimize possible delays in the sailing schedule of the barge. The sailing schedule determines the time at which a barge is planned to be in the port and at its hinterland destinations. The key performance indicator of a terminal operator is to maximize the utilization of his resources. The key performance indicator of the port is to minimize the congestion in the port. In our model we fix the capacity and utilization degree of terminals and we therefore focus on barge related KPIs, which in turn reflect the degree of congestion in the port. We operationalize the mentioned KPIs by measuring the fraction of barges leaving the port late and the (average) amount of time barges leave the port late (or early).

10.1.3 Research Question 3

The third research question is: *What is an efficient and effective Multi-Agent system for the barge handling problem?*

The design of a Multi-Agent system is generally about the strategy of players and the interaction protocol. Both are important to gain *acceptance* and to facilitate the *optimization* of the operations of the actors concerned. In a previous study (Approach 1) a Multi-Agent system was proposed, consisting of two types of agents, namely barge operator agents and terminal operator agents. In this Multi-Agent system, barge and terminal operator agents align their operations once every day, at a fixed time, for the next 24h. To determine a rotation, barge operator agents can ask terminal operator agents repeatedly whether a certain time slot is convenient, and terminal operator agents reply with 'yes' or 'no'. We show that this Multi-Agent system has room for improvement, both with respect to optimization and acceptance. We therefore propose our own Multi-Agent system. In our Multi-Agent system we use the same types of agents as in Approach 1, but we apply a different interaction protocol and agent strategies. We assume that the strategy of an agent is to behave opportunistically and to make the best decision for its principal. We propose a new interaction protocol and introduce the concept of *waiting profile*. A waiting profile expresses the guaranteed maximum waiting time until the start of service given a certain arrival time during a certain time period. Waiting profiles facilitate an efficient communication between barge and terminal operators and allow for optimization of the operations of each of the actors concerned. The interaction protocol we propose is based on negotiation and consists of two phases. In the first phase, terminal operators issue waiting profiles to a barge. A barge operator agent determines its best rotation, by using the information provided in the waiting profiles. In the second phase of the negotiation an appointment is made. An appointment means that a barge operator promises the terminal operator to arrive prior to a latest arrival time. The terminal operator in turn promises the barge operator a guaranteed maximum waiting time until the start of service, provided that the barge has arrived prior to its latest arrival time.

The Multi-Agent system we propose allows, in contrast with Approach 1, for real-time alignment of barge and terminal operations. Moreover, it gives barge and terminal operators the possibility to optimize their operations with only limited communication. Chapter 3 explains our Multi-Agent system in detail, including its strengths and weaknesses. In Chapter 6 we extend the concept of waiting profiles to service-time profiles.

10.1.4 Research Question 4

The fourth research question is: *How to evaluate the performance of our Multi-Agent system?*

We evaluate the performance of our Multi-Agent system by means of simulation. In addition to waiting profiles, we also consider other levels of information exchange resembling, e.g., the interaction protocol applied in Approach 1. We compare the results with an off-line benchmark. Thus, we obtained insight in the performance of a distributed planning system compared to a central planning system. In Chapter 4 we discuss successively our conceptual simulation model, the off-line benchmark, the model parameters, the data sets used for simulation, and the data sources.

10.1.5 Research Question 5

The fifth research question is: *How does our Multi-Agent system perform in various port settings?*

This research question is answered in the Chapters 5 and 6. In Chapter 5 we focus on simplified port settings to get a basic understanding of the performance of our Multi-Agent system. In Chapter 6 we consider more realistic port settings. Let us describe the contribution of each chapter and the conclusions we can draw.

In Chapter 5 we specify the Multi-Agent system proposed in Chapter 3. We model the barge and terminal operator agent and explain in detail how waiting profiles can be constructed. We propose to add slack to the waiting profiles and we show how this enhances the planning flexibility of terminals.

Based on the simulation results we draw the following general conclusions. Our Multi-Agent system clearly outperforms more restricted levels of information exchange. Moreover, our Multi-Agent system performs well compared to the off-line benchmark, especially when we take into account that the off-line benchmark has a priori information about the planning horizon. In-depth analysis of the waiting times in the system revealed an interesting result, which might seem counterintuitive at first sight. We observed that, when the utilization degree of the port is high (>75%), then the average total waiting time in a rotation of a barge does not necessarily increase if a barge visits more terminals. The reason why is that a barge plans to visit a bottleneck terminal at the end of its rotation. During the waiting time of this terminal it visits other terminals in the port, thus using waiting time for sailing and handling. For other insights we obtained we refer to Chapter 5.

In Chapter 6 we consider more realistic (although still fictitious) port settings, including terminals with restricted opening times, sea vessels, unbalanced networks, and closing times of containers. We extend the Multi-Agent system developed in Chapter 5 by introducing the notion of service-time profiles, which is an extension of waiting profiles, to deal with time-dependent service-times due to the restricted opening times of terminals.

Simulation results show again that our Multi-Agent system (using service-time profiles) outperforms more restricted levels of information exchange. Comparing the simulation results with the off-line benchmark, learns us that our Multi-Agent system can deal well with restricted opening times, unbalanced networks, and sea vessels. However, in case containers have closing times, our Multi-Agent system has difficulty to provide solutions of similar quality as the off-line benchmark.

The experimental results in the Chapters 5 and 6 indicate that our Multi-Agent system might be promising in practice.

10.1.6 Research Question 6

The sixth research question is: *What are relevant extensions to the model?*

Based on the results obtained in the Chapters 5 and 6 we decided to consider two extensions to our Multi-Agent model, namely i) different degrees of cooperativeness of a terminal, and ii) disturbances and uncertainty. We discuss the extensions successively.

In the Chapters 5 and 6 we assumed that terminals are cooperative in the sense that they share the required information and are willing to make appointments. However, what will happen if terminal operators are not so cooperative? To investigate that, we introduce the following, so-called, degree of cooperativeness of terminals:

1. *Fully cooperative*: a terminal issues waiting profiles and keeps the appointments made with barges (the situation in the Chapters 5 and 6).

2. *Partly cooperative*: a terminal issues waiting profiles, but processes barges first-come first-served (FCFS).

3. *Lowly cooperative*: a terminal gives (limited) insight in the current state of the barge queue at the terminal, and processes barges FCFS.

We analyze the different degrees of cooperativeness based on the simplified (fictitious) port settings of Chapter 5. We conclude that the lowly cooperative situation performs surprisingly well compared to the fully cooperative situation. The results can probably be improved when barges share or make more sophisticated use of available information. This means that lowly cooperative might be a reasonable alternative, if terminals are not willing to be fully cooperative. However, we expect that the performance in the lowly cooperative case worsens in more realistic port settings (including sea vessels and restricted opening times of terminals), when information about future activities of the terminals becomes important.

The second extension is a preliminary exploration of the effects of disturbances and uncertainties on the operations of barges and terminals. In Chapter 7 we explain different uncertainties in the daily operations of barges and terminals and we make a distinction between minor and major disturbances. In case of minor disturbances, we suggest that terminal and barge operators create robust quays schedules and rotations. We mention three forms of robustness (stability, flexibility, and nervousness avoidance) and we explain how these forms of robustness can be achieved. We explain why the appointment structure implemented in our Multi-Agent system dampens the effects of, especially, minor disturbances. To deal with major disturbances we suggest intensified communication and cooperation among actors during a disturbance, to minimize the disrupting effects on all parts of the system. The port authority could play a central role in that. Simulation of a water blockage indicates that a disturbance can have a major impact on the system, especially when terminals are highly utilized.

10.1.7 Research Question 7

The seventh research question is: *How does our Multi-Agent system perform in the Port of Rotterdam?*

For this research question we focused on the Port of Rotterdam. We created a base case, which is a realistic model of the Port of Rotterdam for the time frame 2006-2007. In addition, in order to perform a sensitivity analysis, we considered two scenarios. These scenarios are based on ideas that are studied in practice. The first scenario (Scenario 1) is based on an idea to use large barges to ship containers to a transferium far in the hinterland (say Duisburg), where the containers can be distributed to their respective hinterland destinations. This will probably lead to shorter rotation lengths of barges and larger call sizes at terminals. The second scenario (Scenario 2) is based on an idea to build a transferium close to the port, to transfer containers from truck to barge and vice versa. The aim is to relieve the congested roads leading to the sea terminals. Scenario 2 will probably lead to a higher utilization of the terminals, since there will be additional barge movements compared to the current situation.

The results for the base case indicate that a Multi-Agent system with service-time profiles performs well compared to the off-line benchmark, and outperforms the other Multi-Agent controls. The comparison with the off-line benchmark suggests that our Multi-Agent system is promising in the port.

In Scenario 1 we simulated the effect of shorter rotation lengths of barges. The results indicate that this leads to a higher throughput of the port, without an increase in the utilization of terminals. If the throughput of the port stays the same as in the base case, then shorter rotation lengths will lead to a reduced utilization degree of terminals and shorter sojourn times of barges. In Scenario 2 we simulated the effect of an increased utilization degree of terminals. The results show that, besides an increased throughput, a more than proportional increase in waiting times of barges can be observed. We wonder whether this effect is currently taken into account.

10.1.8 Research Question 8

The eighth research question is: *How can we effectively communicate our solution to practice?*

As we mentioned earlier, the design of a Multi-Agent system is both about optimization and acceptance. To communicate our Multi-Agent system to practice, we developed in Chapter 9 a multi-player game resembling our Multi-Agent system. Through the game we aim to facilitate the acceptance process of the Multi-Agent system. In the game, players act as living agents and they have to solve the problems that agents have to solve in the Multi-Agent system. In this way, they can experience different solutions and create an understanding of the functioning of the system.

The first experiences with the game are encouraging and promising. In three workshops participants indicated that the game increased their understanding of our solution. In the game they experienced the dynamics in the problem and the complexity of aligning activities between multiple self-interested parties. Through the game we were able to start pointed discussions with participants about their attitude towards the solution, the difficulties they expect in the implementation, the extent to which waiting profiles reveal sensitive information, whether the solution is acceptable, etcetera.

Although the first experiences with the game are promising, we cannot yet conclude that our game is an effective means to facilitate the acceptance process of our Multi-Agent system. To get more experience, we plan to organize other workshops, preferably with barge and terminal operators. Through that we hope to get insight in the extent to which a game contributes to the understanding of our Multi-Agent system, provides a tool for effective communication of our solution to practice, and can be used as prototyping technique for the further development of our Multi-Agent system.

10.2 Further research

We distinguish three areas for further research: model improvements, model extensions, and implementation aspects.

10.2.1 Model improvements

As topics for further research we mention the following two model improvements, namely: the personalization of the barge and terminal operator agent and a detailed simulation of several practical situations. We describe these improvements successively.

Personalization of the barge and terminal operator agent
Throughout our thesis we assume that all barge operators (and terminal operators) have identical intelligence and are only concerned about their *key* performance indicator. In practice, barge operators (and terminal operators) might have all kinds of preferences when planning their rotations or quays. To make our Multi-Agent system ready for implementation in practice, we think it is necessary to give barge and terminal operators the possibility to declare their preferences to their agents. For barge operators it might, e.g., be interesting to have waiting time in a rotation clustered at specific points to have time for shopping or other activities. For terminal operators it might be interesting to be able to deal with different priorities of containers (and the associated barges), to take into account the reputation of a barge, and to apply preferred strategies to deal with unexpected events or to recover from disturbances.

Detailed simulation of several practical situations
In our thesis we simulated a realistic situation of the Port of Rotterdam. However, the level of abstraction in our model did not allow us to include all kinds of specific characteristics in the port. Besides, we struggled with the absence of reliable historical information about the barge handling problem in the Port of Rotterdam. Before implementing our Multi-Agent system, it would be interesting to perform another simulation study based on historical information or future scenarios, such as the 2^{nd} Maasvlakte, to answer the following research questions: what is a critical mass of actors to guarantee a successful implementation of the system, how to deal with disturbances, how to treat actors that

frustrate the Multi-Agent system, and how to prevent the system from collapsing because of the behavior of specific actors or because of certain (disturbing) events.

10.2.2 Model extensions

We mention three interesting model extensions, namely: dealing with disturbances, extending the interaction protocol, and extending the system boundaries.

Dealing with disturbances
In Chapter 7 we briefly touched upon the notions of disturbances and uncertainty. We recommend to extend our model to deal properly with these two notions. The following research questions need to be carefully assessed: i) how can barge and terminal operators deal with disturbances and uncertainty, ii) what are the risks of misuse of mechanisms to deal with uncertainty, and iii) how will actors respond to strategic behavior of other actors.

Extending the interaction protocol
In our Multi-Agent system we did not yet include the option for barge operator agents to communicate among each other. It would be interesting to extend our interaction protocol to enable barges to, e.g., exchange appointments or to share relevant information.

Extending the system boundaries
In a future study it would be interesting to extend the system boundaries beyond the port by including parts of the hinterland. By doing so, we can integrate in an early stage (when the barge is still in the hinterland) the stowage of the barge and the planning of the rotation in the port. Moreover, barge shippers can adjust their sailing speed in the hinterland when they know early at what time they are expected at the first terminal in the port. In this way they can save fuel and lower their operational costs.

10.2.3 Implementation aspects

In this section we mention some issues regarding the implementation of our Multi-Agent system. We mention some necessary conditions for implementation and discuss the question who implements and manages the Multi-Agent system. Finally, we discuss the usefulness of our management game as prototyping technique.

Necessary conditions for implementation
Before implementing our Multi-Agent system several conditions need to be fulfilled to make the implementation successful and the operation manageable. We mention a proper IT-infrastructure, enabling all actors in the port to communicate with each other. In addition, it must be possible to register unambiguously

and objectively the performance of barges and terminals. We expect that both conditions can be fulfilled in the near future.

Who implements and manages the Multi-Agent system?
When actors in the Port of Rotterdam agree on implementation of our Multi-Agent system, the question is: who is going to implement and manage the Multi-Agent system? One might argue that no platform operator is necessary, since the system can be implemented in a distributed fashion. However, there are several issues that need to be addressed, regarding the responsibility for the design and use of the Multi-Agent system. We mention the following: who makes agreements with actors about the functioning and use of the Multi-Agent system, who is responsible for the functioning of the Multi-Agent system, who will take care that actors live up to the agreements made, who is punishing actors that exhibit undesirable behavior, and who takes the lead if the system is down due to a general breakdown?

Use of the management game
The management game we developed in Chapter 9 can be used as prototyping technique prior to implementation of our Multi-Agent system. Through the game several issues can be assessed. First, do people understand the functioning of our Multi-Agent system and are they willing to participate? Second, how should the user interface be designed, such that actors can work effectively with the system? Third, how should the communication between the principal and the agent be designed and what kind of preferences do actors want their agent to take into account? Finally, the game can be used to simulate several practical situations to see how the system functions and how players make their decisions.

Before implementation
We can imagine that, when our Multi-Agent system is about to be implemented, not all actors in the port are willing to participate. This may result in a hybrid setting, where part of the terminal and barge operators participate and make appointments through our Multi-Agent system and others make appointments as they do it nowadays. Such a hybrid setting can be complicated, since a barge operator, e.g., may not have all the necessary information at his disposal at the time he has to plan a rotation. Although a field test would be desirable before implementing the system, we expect that a field test with less than the critical mass might be problematic due to the hybrid setting that will arise.

This thesis provides a comprehensive study to the barge handling problem and proposes a solution in the form of a Multi-Agent system. Although the theoretical results are promising, the way towards implementation might be long. Nevertheless, we think that advanced planning techniques, such as the ones we discuss in our thesis, are essential for a port, such as the Port of Rotterdam, to maintain or even improve its competitive position in the world.

Bibliography

Ahn, B. and Shin, J. (1991). Vehicle-routeing with time windows and time-varying congestion, *The Journal of the Operational Research Society* **42**(5): 393–400.

Anderson, P., Cannon, H., Malik, D. and Thavikulwat, P. (1998). Games as instruments of assessment: a framework for evaluation, *Developments in business simulation and experiential learning* **25**: 31–39.

Angelides, M. and Paul, R. (1993). Developing and intelligent tutoring system for a business simulation game, *Simulation Practice and Theory* **1**(3): 109–135.

Angelides, M. and Paul, R. (1999). A methodology for specific, total enterprise, role-playing, intelligent gaming-simulation environment development, *Decision Support Systems* **25**: 89–108.

Arbib, C. and Rossi, F. (2000). Optimal resource assignment through negotiation in a multi-agent manufacturing system, *IIE Transactions* **32**: 963–974.

Asdemir, K., Jacob, V. and Krishnan, R. (2008). Dynamic pricing of multiple home delivery options, *European Journal of Operational Research* . Forthcoming: doi:10.1016/j.ejor.2008.03.005.

Barreteau, O., Bousquet, F. and Attonaty, J.-M. (2001). Role playing games for opening the black box of multi-agent systems: method and lessons of its application to senegal river valley irrigated systems, *Journal of artificial societies and social simulation* **4**(2).

Böcker, J., Lind, J. and Zirkler, B. (2001). Using a multi-agent approach to optimise the train coupling and sharing system, *European Journal of Operational Research* **131**(2): 242–252.

Bürckert, H., Fund, P. and Vierke, G. (2000). Holonic transport scheduling with Teletruck, *Applied Artificial Intelligence* **14**(7): 697–725.

BvB (2007). *The power of inland navigation*, Bureau Voorlichting Binnenvaart.

Camerer, C. (1997). Progress in behavorial game theory, *The Journal of Economic Perspectives* **11**(4): 167–188.

Campbell, A. and Savelsbergh, M. (2005). Decision support for consumer direct grocery initiatives, *Transportation Science* **39**: 313–327.

Campbell, A. and Savelsbergh, M. (2006). Incentive schemes for attended home deliveries, *Transportation Science* **40**(3): 327–341.

CBRB (2003). Basisdocument containerbinnenvaart, *Technical report*, Centraal Bureau Rijn en Binnenvaart.

CEMT (1992). ECMT resolution 92/2 on new classification of inland waterways.

Christiansen, M. and Fagerholt, K. (2002). Robust ship scheduling with multiple time windows, *Naval Research Logistics* **49**(6): 611–625.

Christiansen, M., Fagerholt, K. and Ronen, D. (2004). Ship routing and scheduling: Status and perspectives, *Transportation Science* **38**(1): 1–18.

Connekt (2003). Eindrapport approach, *Technical report*, Connekt, Delft. Report is in Dutch.

Cordeau, J., Laporte, G., Legato, P. and Moccia, L. (2005). Models and tabu search heuristics for the berh-allocation problem, *Transportation Science* **39**(4): 526–538.

Davidsson, P., Henesey, L., Ramstedt, L., Törnquist, J. and Wernstedt, F. (2005). An analysis of agent-based approaches to transport logistics, *Transportation Research Part C* **13**(4): 255–271.

De Binnenvaartkrant (2008). Bakkenshuttle naar duisburg, *De Binnenvaartkrant* **5**.

De Souza, G., Beresford, A. and Pettit, S. (2003). Liner shipping companies and terminal operators: internationalisation or globalisation?, *Maritime Economics and Logistics* **5**: 393–412.

Decker, K. and Li, J. (2000). Coordinating mutually exclusive resources using GPGP, *Autonomous Agents and Multi-Agent Systems* **3**(2): 133–157.

Demeulemeester, E. and Herroelen, W. (1992). A branch-and-bound procedure for the multiple resource-constrained project scheduling problem, *Management Science* **38**(12): 1803–1818.

Demeulemeester, E. and Herroelen, W. (2002). *Project Scheduling: A Research Handbook*, Kluwer Academic Publishers.

Dewan, P. and Joshi, S. (2002). Auction-based distributed scheduling in a dynamic job shop environment, *International Journal of Production Research* **40**(5): 1173–1191.

Durfee, E. and Lesser, V. (1989). Negotiating task decomposition and allocation using partial global planning, *in* L. Gasser and M. Huhns (eds), *Distributed artificial intelligence: vol. 2*, Morgan Kaufmann Publishers Inc., San Francisco, CA, pp. 229–243.

Ellington, H., Addenall, E. and Percival, F. (1982). *A Handbook of Game Design*, Kogan Page, London.

Ertogral, K. and Wu, S. (2000). Auction-theoretic coordination of production planning in the supply chain, *IIE Transactions* **32**(10): 931–940.

ESPO (2007). *Annual Report 2006-2007*, European Sea Ports Organisation.

Falk, A. and Fischbacher, U. (2006). A theory of reciprocity, *Games and Economic Behavior* **54**: 293–315.

Figliozzi, M. (2004). *Performance and Analysis of Spot Truck-Load Procurement Markets Using Sequential Auctions*, PhD thesis, School of Engineering, University of Maryland.

Figliozzi, M., Mahmassani, H. and Jaillet, P. (2006). Quantifying opportunity costs in sequential transportation auctions for truckload acquisition, *Transportation Research Record* **1964**: 247–252.

Fleischmann, B., Gietz, M. and Gnutzmann, S. (2004). Time-varying travel times in vehicle routing, *Transportation Science* **38**(2): 160–173.

Franck, B. and Bunel, J. (1991). Contestability, competition and regulation: The case of liner shipping, *International Journal of Industrial Organization* **9**(1): 141–159.

Franz, T., Voss, S. and Rölke, H. (2007). Market-mechanisms for integrated container terminal management, *Proceedings of the 6th international conference on computer and IT applications in the maritime industry*, pp. 234–248.

Gemeente Rotterdam (2004). *Havenplan 2020: Ruimte voor kwaliteit.*

Gintis, H. (2000). *Game theory evolving. A problem-centered introduction to modeling strategic interaction*, Princeton University Press.

Glaister, K. W. and Buckley, P. J. (1996). Strategic motives for international alliance formation, *Journal of Management Studies* **33**(3): p301 – 332.

Gosen, J. and Washbush, J. (2004). A review of scholarship on assessing experiential learning effectiveness, *Simulation & Gaming* **35**(2): 270–293.

Gross, D. and Harris, C. (1998). *Fundamentals of Queueing Theory*, 3 edn, Wiley, John & Sons, Incorporated.

Haralambides, H. (2002). Competition, excess capacity, and the pricing of port infrastructure, *International Journal of Maritime Economics* **4**: 323–347.

Hardin, G. (1968). The tragedy of the commons, *Science* **162**.

Henesey, L. (2006). *Multi-Agent Container Terminal Management*, PhD thesis, Blekinge Institute of Technology.

Henesey, L., Davidsson, P. and Persson, J. (2006). *Agent based simulation architecture for evaluating operational policies in transshipping containers*, Lecture Notes in Computer Science, pp. 73–85.

Herroelen, W. and Leus, R. (2005). Project scheduling under uncertainty, survey and research potentials, *European Journal of Operational Research* **165**(2): 289–306.

Hoogewegen, M., Van Liere, D., Vervest, P., Hagdorn, L. and De Lepper, I. (2006). Strategizing for mass customization by playing the business network game, *Decision Support Systems* **42**: 1402–1412.

Hopp, W. and Spearman, M. (2000). *Factory Physics*, 3 edn, McGraw Hill, New York, NY.

Huhns, M. and Stephens, L. (1999). *Multi Agent Systems: A modern approach to distributed artificial intelligence*, 1 edn, MIT Press, chapter Multi Agent Systems and Societies of Agents, pp. 79–102.

Ichoua, S., Gendreau, M. and Potvin, J. (2006). Exploiting knowledge about future demands for real-time vehicle dispatching, *Transportation Science* **40**(2): 211–225.

Imai, A., Nishimura, E., Hattori, M. and Papadimitriou, S. (2007). Berth allocation at indented berths for mega-containerships, *European Journal of Operational Research* **179**: 579–593.

Imai, A., Nishimura, E. and Papadimitriou, S. (2001). The dynamic berth allocation problem for a container port, *Transportation research part B* **35**(4): 401–417.

Jacobs, P. (2005). *The DSOL simulation suite. Enabling multi-formalism siimulation in a distributed context.*, PhD thesis, Delft University of Technology.

Jennings, N., Sycara, K. and Wooldridge, M. (1998). A roadmap of agent research and development, *Autonomous Agents and Multi-Agent Systems* **1**(1): 7–38.

Jensen, M. (2001). *Robust and flexible scheduling with evolutionary computation*, PhD thesis, University of Aarhus.

Kolisch, R. (1996). Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation, *European Journal of Operational Research* **90**: 320–333.

Kolisch, R. and Drexl, A. (1996). Adaptive search for solving hard project scheduling problems, *Naval Research Logistics* **43**(1): 23–40.

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

Kozlak, J., Créput, J., Hilaire, V. and Koukam, A. (2004). Multi-agent environment for dynamic transport planning and scheduling, *in* M. Bubak, G. van Albada, P. Sloot and J. Dongarra (eds), *International Conference on Computational Science (ICCS 2004)*, Vol. 3038 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 638–645.

Kraus, S. (1997). Negotiation and cooperation in multi agent environments, *Artificial Intelligence* **94**: 79–97.

Kumar, K. and Van Dissel, H. (1996). Sustainable collaboration: Managing conflict and cooperation in interorganizational systems, *MIS Quartely* **20**: 279–300.

Kumar, P., Kalwani, M. and Dada, M. (1997). The impact of waiting time guarantees on customers' waiting experiences, *Marketing Science* **16**(4): 295–314.

Kuo, T., Huang, W., Wu, S. and Cheng, P. (2006). A case study of interarrival time distributions of containers ships, *Journal of Marine Science and Technology* **14**(3): 155–164.

Law, A. and Kelton, W. (2000). *Simulation Modeling and Analysis*, Vol. 3, McGraw-Hill, Singapore.

Le Bars, M. and Le Grusse, P. (2008). Use of a decision support system and a simulation game to help collective decision-making in water management, *Computers and electronics in agriculture* **62**: 182–189.

Leon, V., Wu, S. and Storer, R. (1994). Robustness measures and robust scheduling for job shops, *IIE Transactions* **26**(5): 32–43.

Leus, R. (2003). *The generation of stable project plans - complexity and exact algorithms*, Ph.d., Katholieke universiteit Leuven.

Lloyd's List (2007). Benelux port delays costing cargo owners; congestion at antwerp and rotterdam worsens, *Lloyd's List* . August 17., 2007.

Lokuge, P. and Alahakoon, D. (2007). Improving the adaptability in automated vessel scheduling in container ports using intelligent software agents, *European Journal of Operational Research* **177**: 1985–2015.

Mailath, G. and Samuelson, L. (2006). *Repeated Games and Reputations: Long-Run Relationships*, Oxford University Press.

Malandraki, C. and Dial, R. (1996). A restricted dynamic programming heuristic algorithm for the time dependent traveling salesman problem, *European Journal of Operational Research* **90**: 45–55.

Mehta, S. and Uzsoy, R. (1998). Predictable scheduling of a job shop subject to breakdowns, *IEEE Transactions on Robotics and Automation* **14**(3): 365–378.

Melis, M., Miller, I., Kentrop, M., Van Eck, B., Leenaarts, M., Schut, M. and Treur, J. (2003). Distributed rotation planning for container barges in the port of rotterdam, *in* T. Verduijn and B. van de Loo (eds), *Intelligent Logistics Concepts*, Eburon Publishers, pp. 101–116.

Mes, M. (2008). *Sequential Auctions for Full Truckload Allocation*, PhD thesis, School of Management and Governance, University of Twente, The Netherlands.

Mes, M., Van der Heijden, M. and Schuur, P. (2008). Dynamic threshold policy for delaying and breaking commitments in transportation auctions, *Transportation Research Part C* **Forthcoming - doi:10.1016/j.trc.2008.03.001**.

Mes, M., Van der Heijden, M. and Van Harten, A. (2007). Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems, *European Journal of Operation Research* **181**(1): 59–75.

Mes, M., Van der Heijden, M. and Van Hillegersberg, J. (2008). Design choices for agent-based control of AGVS in the dough making process, *Decision Support Systems* **44**(4): 983–999.

Moonen, H. and Van de Rakt, B. (2005). Approach 2 - building the vision, *Technical report*.

Moonen, H., Van de Rakt, B., Miller, I., Van Nunen, J. and Van Hillegersberg, J. (2007). Agent technology supports inter-organizational planning in the port, *in* R. de Koster and W. Delfmann (eds), *Managing Supply Chains - Challenges and Opportunities*, Copenhagen Business School Press, Copenhagen.

Nieuwsblad Transport (2007a). Contargo voert congestietoeslag in. July 16, 2007.

Nieuwsblad Transport (2007b). Stremming rijn zorgt voor lange 'file'. March 27, 2007.

Notteboom, T. (2002). The interdependence between liner shipping networks and intermodal networks, *Proceedings of the IAME, Panama*.

Notteboom, T. (2004). Container shipping and ports: An overview, *Review of Network Economics* **3**(2).

Notteboom, T. (2007). Strategic challenges to container ports in a changing market environment, *Research in Transportation Economics* **17**: 29–52.

Paessens, H. (1988). The savings algorithm for the vehicle routing problem, *European Journal of Operational Research* **34**: 336–344.

Park, Y. and Kim, K. (2003). A scheduling method for berth and quay cranes, *OR Spectrum* **25**: 1–23.

Paulussen, T., Jennings, N., Decker, K. and Heinzl, A. (2003). Distributed patient scheduling in hospitals, *Proceedings 18th international joint conference on AI*.

Pinedo, M. and Chao, X. (1999). *Operations Scheduling. With applications in manufacturing and services*, Irwin/McGraw-Hill.

Qinghe, H., Kumar, A. and Shuang, Z. (2001). A bidding decision model in multiagent supply chain planning, *International Journal of Production Research* **39**(15): 3291–3301.

RIL Foundation (1998). Afhandeling containerbinnenvaart rotterdam, *Technical report*, B-RIL, CBRB, and TNO-Inro.

RIL Foundation (2000). Functioneel ontwerp bargeplanning.nl, *Technical report*.

Ronen, D. (1983). Cargo ships routing and scheduling: Survey of models and problems, *European Journal of Operational Research* **12**: 119–126.

Ross, S. (2003). *Introduction To Probability Models*, 8 edn, Academic Press.

Ryan, T. (2000). The role of simulation gaming in policy-making, *Systems research and behavorial science* **17**: 359–364.

Ryoo, D. and Thanopoulou, H. (1999). Liner alliances in the globalisation era: A strategic tool for asian container carriers, *Maritime Policy and Management* **26**(4): 349–367.

Sandholm, T. (1999). *Multiagent Systems, A Modern Approach to Distributed Artifical Intelligence*, 1 edn, MIT press, chapter Distributed Rational Decision Making, pp. 201–258.

Schotanus, F. (2007). *Horizontal Cooperative Purchasing*, PhD thesis, Beta Research School.

Schut, M., Kentrop, M., Leenaarts, M., Melis, M. and Miller, I. (2004). Approach: Decentralised rotation planning for container, *Proceedings of the 16th European Conference on Artificial Intelligence*, Prestigious Applications Intelligent Systems, Valencia.

Schutten, J. (1996). List scheduling revisited, *Operations Research Letters* **18**: 167–170.

Schutten, J. (1998). Practical job shop scheduling, *Annals of Operations Research* **83**: 161–178.

Sevaux, M. and Sörensen, K. (2002). A genetic algorithm for robust schedules, *Proceedings of 8th International Workshop on Project Management and Scheduling, PMS'2002*, pp. 330–333.

Shen, W. and Norrie, D. (1999). Agent-based systems for intelligent manufacturing: A state-of-the-art survey, *Knowledge and Information Systems, an International Journal* **1**(2): 129–156.

Sinclair, M. and Van Dyk, E. (1987). Combined routeing and scheduling for the transportation of containerized cargo, *The Journal of the Operational Research Society* **38**(6): 487–498.

Slack, B., Comtois, C. and Sletmo, G. (1996). Shipping lines as agents of change in the port industry, *Maritime Policy & Management* **23**(3): 289–300.

Smith, R. (1980). The contract net protocol: High-level communication and control in a distributed problem solver, *IEEE Transactions on Computers* **C-29**(12): 1104–1113.

Stahlbock, R. and Voß, S. (2008). Operations research at container terminals: a literature update, *OR Spectrum* **30**: 1–52.

Steenken, D., Voß, S. and Stahlbock, R. (2004). Container terminal operation and operators research - a classification and literature review, *OR Spectrum* **26**: 3–49.

't Hoen, P. and La Poutré, J. (2004). A decommitment strategy in a competitive multi-agent transportation setting, *in* P. Faratin, D. Parkes and J. Rodriquez-Aguilar (eds), *Agent Mediated Electronic Commerce V (AMEC-V)*, Vol. 3048 of *Lecture Notes in Artificial Intelligence LNAI*, Springer-Verlag, pp. 56–72.

't Hoen, P. and La Poutré, J. (2006). Repeated auctions with complementarities, *Agent Mediated Electronic Commerce VII (AMEC-VII*, Vol. 3937 of *Springer Lecture Notes in Artificial Intelligence (LNAI)*, Springer-Verlag, pp. 16–29.

Thangiah, S., Shmygelska, O. and Mennell, W. (2001). An agent architecture for vehicle routing problems, *Proceedings of the 2001 ACM symposium on Applied computing (SAC '01)*, ACM, New York, NY.

Thurston, T. and Hu, H. (2002). Distributed agent architecture for port automation, *Proceedings of the 26th Annual International Computer Software and Applications Conference*.

Tijms, H. (2003). *A First Course in Stochastic Models*, 2 edn, John Wiley & Sons.

Van der Horst, M. and De Langen, P. (2008). Coordination in hinterland transport chains: A major challenge for the seaport community, *Maritime Economics and Logistics* **10**: 108–129.

Van Dyke Parunak, H. (1999). Industrial and practical applications of DAI, *Multiagent systems: a modern approach to distributed artificial intelligence*, The MIT Press, Cambridge, MA, pp. 377–421.

Van Groningen, L. (2006). *Performance measurement in barge planning*, Master's thesis, RSM Erasmus University.

Vermeulen, I., Bohte, S. and Somefun, K. (2007). Multi-agent pareto appointment exchanging in hospital patient scheduling, *Service Oriented Computing and Applications* **1**(3): 185–196.

Vis, I. and De Koster, R. (2003). Transshipment of containers at a container terminal: an overview, *European journal of Operations Research* **147**: 1–16.

Visser, J., Konings, R., Pielage, B. and Wiegmans, B. (2007). A new hinterland transport concept for the port of rotterdam: organisational and/or technological challenges?, *Proceedings of the Transportation Research Forum*, North Dakota State University.

Welch, P. (1983). *The Computer Performance Modeling Handbook*, Academic Press, New York, chapter The Statistical Analysis of Simulation Results.

Wenzler, I. and Chartier, D. (1999). Why do we bother with games and simulations: an organizational learning perspective, *Simulation Gaming* **30**(3): 375–384.

Whitt, W. (1999). Improving service by informing customers about anticipated delays, *Management science* **45**(2): 192–207.

Wooldridge, M. (2005). *An Introduction to Multi Agent Systems*, 4th edn, John Wiley & Sons.

Wooldridge, M. and Jennings, N. (1995). Intelligent agents: Theory and practice, *The Knowledge Engineering Review* **10**(2): 115–152.

Wullink, G. (2005). *Resource Loading under Uncertainty*, PhD thesis, Beta - Research School for Operations Management and Logistics.

Zhu, K., Ludema, M. and Van der Heijden, R. (2000). Air cargo transport by multi-agent based planning, *Proceedings of the Thirty-third Annual Hawaii International Conference on Systems Sciences*, Maui, Hawaii.

Zlotkin, G. and Rosenschein, J. (1996). Mechanism design for automated negotiation, and its application to task oriented domains, *Artificial Intelligence* **86**: 195–244.

Appendices

Appendix A

To evaluate the performance of the DP-heuristic of Malandraki and Dial (1996), we performed several off-line experiments. For the experiments we created several instances, which are solved using the DP-heuristic of Malandraki and Dial (1996), a branch-and-bound algorithm, and the savings-based algorithms with a 2-opt procedure as described in Section 5.2.2. Every experiment is a combination of the following variables. First, the number of regions (1, 2, or 3). Second, the number of terminals per region (varying from 3-12) with a maximum of 12 terminals for the whole network. Third, randomly generated waiting profiles for varying average waiting times.

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

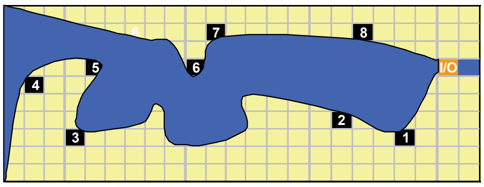
The waiting profiles are created as follows. We first compute the number of times the waiting time is zero during a certain planning horizon. We do this by multiplying a sufficient long planning horizon with different values of $k \in \{0.05, 0.025, 0.017, 0.0125\}$, where k is the probability that the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

Appendix B

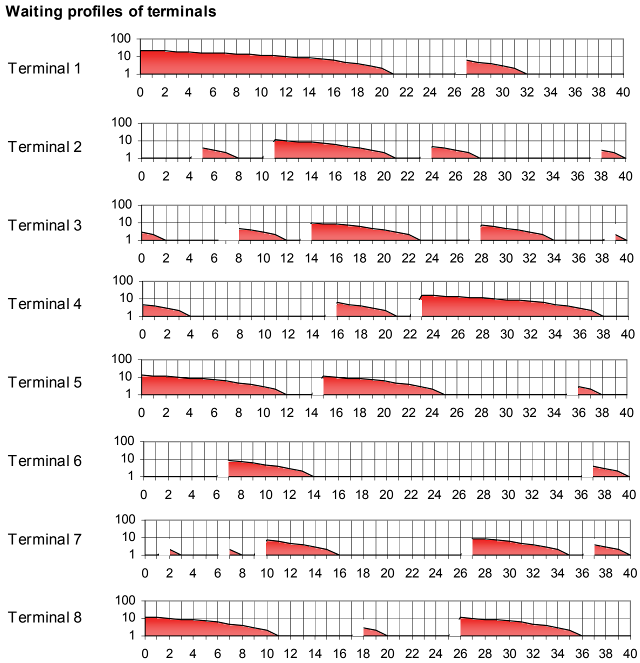
In Figure B.1 we depict the assignment (including a solution) we handed out to participants during the first workshop on the Management Game (as described in Section 9.4). The assignment is to plan the shortest rotation along all the eight terminals, starting and ending in I/O.

Distances in time slots

	1	2	3	4	5	6	7	8	IO
1	0	1	7	8	6	4	5	2	2
2	1	0	6	7	5	3	3	2	2
3	7	6	0	4	2	3	4	6	7
4	8	7	4	0	2	4	5	7	8
5	6	5	2	2	0	2	3	5	6
6	4	3	3	4	2	0	1	4	4
7	5	3	4	5	3	1	0	3	4
8	2	2	6	7	5	4	3	0	2
IO	2	2	7	8	6	4	4	2	0



The processing time at each terminal is one time slot
One box in the time bar is equal to one time slot



The duration of the optimal rotation is 39 time slots (going from I/O point to I/O point, via all terminals). There are in total 45 optimal solutions. An optimal solution is I/O-7-6-5-4-3-2-1-8-I/O

Figure B.1: The assignment given to the participants during the first workshop, described in Section 9.4

Index

Samenvatting

In dit proefschrift onderzoeken we het zogenaamde *'barge handling problem'* dat gaat over de afhandeling van container binnenvaartschepen bij terminals in een haven. Hierbij hebben we te maken met concurrerende partijen die kunnen profiteren van samenwerking, maar daartoe alleen bereid zijn onder specifieke voorwaarden. In dit onderzoek nemen we de haven van Rotterdam als inspiratiebron, hoewel onze oplossing niet tot deze haven beperkt is. Het probleem dat we beschouwen is als volgt.

Voor het vervoer van containers van de haven van Rotterdam naar het achterland (en vice versa) worden binnenvaartschepen (Engels: barges) ingezet. Wanneer een binnenvaartschip vanuit het achterland in de haven van Rotterdam arriveert, moeten gemiddeld ongeveer zeven terminals bezocht worden om containers te lossen en te laden. Een barge operator (het bedrijf dat het binnenvaartschip heeft ingehuurd) wil graag dat het binnenvaartschip deze terminals in een zodanige volgorde bezoekt, dat het binnen een bepaalde tijd de haven kan verlaten. Om dit te realiseren maakt de barge operator afspraken met de desbetreffende terminal operators. Een terminal operator is het bedrijf dat een terminal exploiteert. Het maken van afspraken gaat veelal via telefoon, fax of e-mail. Het komt echter vaak voor dat afspraken niet nagekomen (kunnen) worden. Dit heeft verschillende redenen, zoals het feit dat er soms dubbele afspraken gemaakt worden, maar ook dat een verstoring bij één terminal zich snel kan verspreiden door de hele haven. Het gevolg is dat afspraken erg onbetrouwbaar zijn.

De onbetrouwbaarheid in de afspraken uit zich in onzekere wachttijden voor binnenvaartschepen, onzekere verblijftijden in de haven en onzekere aankomsttijden bij terminals. Dit leidt tot veel problemen in de afstemming van de activiteiten van binnenvaartschepen en terminals, en zorgt ervoor dat deze niet optimaal benut worden.

Het probleem is urgent geworden in de afgelopen jaren als gevolg van de sterke toename van het containerverkeer. In het verleden is geprobeerd één (centrale) partij te installeren die de activiteiten van alle terminals en binnenvaartschepen coördineert. Dit bleek geen acceptabele oplossing te zijn voor de partijen in de haven, aangezien ze hierdoor hun autonomie over de eigen operaties zouden moeten opgeven en bovendien concurrentiegevoelige informatie zouden moeten delen.

In een eerdere studie (genaamd APPROACH 1) is daarom geprobeerd de afstemming van activiteiten van binnenvaartschepen en terminals te regelen door middel van een decentraal systeem, in dit geval een Multi-Agent systeem. In een decentraal systeem wordt de coördinatie van activiteiten gerealiseerd door onderlinge communicatie tussen partijen die zelfstandig beslissingen nemen. Er is niet één centrale partij die beslist wie wat wanneer doet. In een Multi-Agent systeem worden deze interacties geautomatiseerd. Een agent is een stuk software dat handelt in opdracht van zijn principaal en in staat is zelfstandig beslissingen te nemen, te onderhandelen met andere agenten en te reageren of anticiperen op gebeurtenissen. De onderhandeling tussen de agenten gaat via een bepaald interactieprotocol en de beslissingen die genomen worden gelden als bindend in de praktijk. Het voordeel is dat partijen autonoom blijven, dat de communicatie wordt geautomatiseerd en dat niet alle informatie hoeft te worden gedeeld met derden.

In de APPROACH 1 studie is aangetoond dat een Multi-Agent systeem een oplossing kan bieden voor de betrokken partijen. De voorgestelde oplossing stelt barge en terminal operators in staat eenmaal per dag hun activiteiten af te stemmen voor de komende 24 uur. De resulterende planning wordt echter niet meer herzien tijdens de uitvoering. Bovendien is het systeem niet gericht op het *optimaliseren* van de planning van elk van de partijen, maar vooral op het creëren van *uitvoerbare* planningen.

Op basis van de resultaten van de APPROACH 1 studie kwam de vraag naar voren de mogelijkheden van decentrale planning voor het 'barge handling problem' verder te verkennen. Ons onderzoek maakt deel uit van deze verkenning. We hebben ons daarbij met name gericht op de ontwikkeling van een Multi-Agent systeem dat mogelijkheden biedt voor real-time planning en optimalisatie van de operaties van de betrokken partijen. Het doel laat zich als volgt formuleren:

Het doel van het onderzoek is het ontwikkelen van een efficiënt en effectief gedistribueerd planningssysteem voor het 'barge handling problem' en inzicht te krijgen in de manier waarop het voorgestelde systeem werkt.

Met efficiënt bedoelen we de mate waarin barge en terminal operators hun doelstellingen kunnen realiseren, gegeven dat beslissingen worden genomen in real-time en met beperkte communicatie. Met effectief bedoelen we dat het systeem ook daadwerkelijk in de praktijk geïmplementeerd kan worden. Als het systeem dat we voorstellen, hoewel efficiënt, niet acceptabel is voor de betrokken partijen, dan wordt het zeker niet effectief.

In dit proefschrift ontwikkelen we een Multi-Agent systeem gebaseerd op zogenaamde servicetijdprofielen. We komen hier later op terug. Om de prestatie van het Multi-Agent systeem te evalueren voeren we simulatiestudies uit. Hierdoor krijgen we inzicht in de prestatie van het systeem als geheel als gevolg van de interacties van de barge en terminal operators. Daarnaast vergelijken we de prestatie met die van een centraal planningssysteem. Tenslotte hebben we een game ontwikkeld om na te gaan hoe onze oplossing in de praktijk wordt ontvangen. De algemene conclusie is dat ons Multi-Agent systeem een veelbelovende benadering is voor het oplossen van het barge handling problem en dat het kan leiden tot een significante verbetering in de praktijk.

Laten we nu dit proefschrift meer in detail beschrijven. Om het onderzoeksdoel te bereiken hebben we een aantal meer specifieke doelen geformuleerd. Elk van deze doelen is uitgewerkt in één of meerdere hoofdstukken. Hieronder geven we een samenvatting van elk van deze hoofdstukken.

In hoofdstuk 2 beschrijven we de rol die de binnenvaart vervult in het nationale en internationale containervervoer en de ontwikkelingen die daar op dit moment gaande zijn. Daarnaast beschrijven we de Approach 1 studie en een studie naar de prestatie-indicatoren van terminal en barge operators.

In hoofdstuk 3 gaan we in op het ontwerp van ons (conceptuele) Multi-Agent systeem. We laten zien dat we ons bij het ontwerp van een Multi-Agent systeem moeten richten op zowel optimalisatie als acceptatie. Optimalisatie betreft het verbeteren van de operaties van elk van de betrokken partijen. Acceptatie gaat over de vraag of het voorgestelde systeem uiteindelijk acceptabel is voor de gebruikers en daarom geïmplementeerd kan worden. We werken dit verder uit door een lijst op te stellen met gewenste criteria waaraan een Multi-Agent systeem moet voldoen. We analyseren de oplossing die is voorgesteld in de Approach 1 studie en laten zien dat deze oplossing kan worden verbeterd. We stellen daarom ons eigen Multi-Agent systeem voor. Belangrijk hierbij is de informatie die wordt uitgewisseld en de afspraken die worden gemaakt. We stellen voor om informatie uit te wisselen in de vorm van een wachtprofiel. Een wachtprofiel wordt uitgegeven door een terminal en laat zien wat de gegarandeerde maximale wachttijd is tot het moment van afhandeling, gegeven een bepaalde aankomsttijd in een bepaalde tijdsperiode. Deze wachtprofielen kunnen door de barge operator gebruikt worden om een optimale rotatie te bepalen. Een rotatie is een volgorde van terminal bezoeken. Nadat een barge operator een optimale rotatie heeft bepaald, maakt hij afspraken met alle betrokken terminal operators. Een afspraak houdt in dat een barge operator belooft vóór een bepaalde tijd bij de terminal te arriveren. De terminal operator belooft, als de barge inderdaad op tijd arriveert, een maximale wachttijd tot het moment waarop zijn afhandeling wordt gestart.

We evalueren de prestatie van ons Multi-Agent systeem door middel van simulatie. Daarnaast maken we een vergelijking met een zogenaamde off-line benchmark. Deze off-line benchmark bootst de situatie na waarin alles centraal gepland wordt. We maken daarbij gebruik van bestaande optimalisatiealgoritmes en we veronderstellen dat daarbij alle informatie over de gehele planningshorizon bekend is op het moment van planning. In werkelijkheid (evenals in de simulatie) komt informatie beschikbaar door de tijd heen. We bestuderen in hoofdstuk 5 vereenvoudigde en in hoofdstuk 6 realistischer havensituaties. De manier waarop we ons Multi-Agent systeem evalueren, beschrijven we in hoofdstuk 4. We beschrijven daarin achtereenvolgens het conceptuele simulatiemodel, de off-line benchmark, de modelparameters en de data en de datasets die we gebruiken.

Hoofdstuk 5 beschrijft de intelligentie van de barge en terminal operators en geeft aan hoe de wachtprofielen kunnen worden geconstrueerd. Uit de experimenten die we gedaan hebben trekken we de volgende conclusies. Ons Multi-Agent systeem (met wachtprofielen) presteert goed in vergelijking met de off-line benchmark, vooral wanneer we ons realiseren dat de off-line benchmark alle informatie over de planningshorizon heeft op het moment van planning. Een gedetailleerde analyse van de wachttijden in het model leverde een verrassend en een, op het eerste gezicht, niet intuïtief resultaat op. We namen waar dat bij hogere bezettingsgraden van de terminals (meer dan 75%) de gemiddelde totale wachttijd in de rotatie van een binnenvaartschip afneemt als het schip meer terminals aandoet. De reden is dat het schip een afspraak maakt met een bottleneck-terminal en in de wachttijd van deze terminal alle andere terminals in zijn rotatie bezoekt. Op deze manier wordt wachttijd bij de bottleneck-terminal gebruikt voor het varen naar en afhandeling bij andere terminals.

In hoofdstuk 6 bekijken we realistischer havensituaties, waaronder beperkte openingstijden van terminals en zeeschepen. We breiden het wachtprofielconcept uit tot een servicetijdprofiel om te kunnen omgaan met tijdsafhankelijke afhandeltijden die het gevolg zijn van beperkte openingstijden van terminals. De simulatieresultaten bevestigen het beeld dat naar voren kwam in hoofdstuk 5. Dit houdt in dat ons systeem ook goed werkt in realistischer havensituaties.

In hoofdstuk 7 onderzoeken we twee uitbreidingen van ons model, namelijk i) de mate waarin terminals coöperatief zijn en ii) verstoringen en onzekerheid in de dagelijkse operaties. We zullen deze kort beschrijven. In hoofdstuk 5 en 6 hebben we verondersteld dat terminals coöperatief zijn in die zin, dat ze bereid zijn de noodzakelijke informatie te delen en afspraken te maken. Stel echter dat terminals niet bereid zijn afspraken te maken of bepaalde informatie te delen, dat wil zeggen, minder coöperatief zijn. In hoofdstuk 7 simuleren we verschillende maten waarin terminals coöperatief zijn voor de vereenvoudigde havensituaties uit hoofdstuk 5. Een conclusie die we trekken is dat als terminal en barge operators geen betrouwbare afspraken kunnen (of willen) maken, dat ze dan vaak beter af zijn door helemaal geen afspraken te maken. Het maken van betrouwbare afspraken heeft echter tal van voordelen, aangezien het meer zekerheid geeft in de operaties van barge operators en terminal operators. Deze conclusie is dus gebaseerd op vereenvoudigde havensituaties. Of het ook geldt in realistischer havensituaties is onderwerp van toekomstig onderzoek.

De tweede uitbreiding betreft de verkenning van de impact van onzekerheid en verstoringen op de operaties van barge en terminal operators. We beschrijven verschillende verstorende factoren in de praktijk, maken onderscheid tussen kleine en grote verstoringen en we beschrijven hoe terminal en barge operators in de planning met deze verstoringen rekening kunnen houden, zodat het effect van een verstoring wordt geminimaliseerd. Ook doen we een voorstel hoe er omgegaan kan worden met verstoringen die een significant deel van het systeem beïnvloeden. Simulatie van een blokkade van een waterweg laat zien dat verstoringen een grote impact kunnen hebben op het systeem, vooral wanneer terminals hoog bezet zijn.

Naast fictieve havensituaties hebben we ook gekeken naar de haven van Rotterdam. We beschrijven dit in hoofdstuk 8. Het bleek, om verschillende redenen, lastig te zijn betrouwbare en volledige informatie te verkrijgen van de tijdsperiode die wij hebben beschouwd (2006-2007). We hebben op basis van de beschikbare informatie een model gemaakt (de zogenaamde basissituatie). Daarnaast hebben we twee scenario's beschouwd die gebaseerd zijn op ideeën die bestudeerd worden in de praktijk. Het eerste scenario is een idee om grote hoeveelheden containers ongesorteerd naar een transferium in het achterland te varen (bijvoorbeeld Duisburg) en daar verder te distribueren. Dit leidt naar verwachting tot minder terminalbezoeken per rotatie. Het tweede scenario is gebaseerd op het idee om een transferium dichtbij de haven te realiseren, met als doel de wegen te ontlasten door containers van vrachtwagens over te slaan op binnenvaart (en vice versa). We verwachten dat dit tot een hogere kadebezetting leidt van de terminals. De resultaten laten allereerst zien dat ons Multi-Agent systeem met servicetijdprofielen goed presteert in vergelijking met de off-line benchmark en een goede oplossing kan bieden in de praktijk. Verder blijkt dat het eerste scenario tot een hogere doorzet kan leiden in de haven zonder een stijging van de bezettingsgraad van terminals. De reden is dat schepen minder terminals aandoen en de tijd voor het aan- en afmeren gebruiken voor overslag. Als de bezettingsgraad van terminals gelijk blijft als in de basissituatie (wat inhoud dat terminals meer containers overslaan) dan is de verblijftijd van binnenvaartschepen in scenario 1 vrijwel gelijk aan die in de basissituatie. De resultaten van de simulatie van het tweede scenario laten zien dat een toename van het aantal binnenvaartschepen zal leiden tot een meer dan lineaire toename van de wachttijd bij terminals. We betwijfelen of dit effect tegenwoordig voldoende wordt meegenomen in de praktijkstudies.

Ten slotte beschrijven we in hoofdstuk 9 een game die we gemaakt hebben om ons model inzichtelijk te maken voor derden. In de game is een speler de barge operator van een binnenvaartschip en moet hij afspraken maken met bepaalde terminal operators. Deze terminal operators worden gespeeld door de computer. Het doel van de game is dat spelers zelf kunnen ervaren hoe het voorgestelde Multi-Agent systeem werkt en hoe de beslissingen van andere spelers de eigen mogelijkheden beïnvloeden. Door middel van de game zijn we in staat gerichte discussies te genereren over onze oplossing, in hoeverre deze acceptabel is, hoe spelers erop reageren en dergelijke. De ervaringen die we opgedaan hebben in een aantal workshops zijn veelbelovend.

Dit proefschrift beschrijft een uitvoerige studie naar het barge handling problem en stelt een oplossing voor in de vorm van een Multi-Agent systeem. Hoewel de theoretische resultaten veelbelovend zijn, kan de weg naar implementatie wel eens lang zijn. Toch denken we dat geavanceerde planningstechnieken, zoals degene die we in dit proefschrift bespreken, essentieel zijn voor een haven als Rotterdam, om haar concurrentiepositie in de wereld te handhaven of zelfs te verbeteren.

About the author

Albert Douma was born on April 9, 1980, in Achtkarspelen, the Netherlands. In 1998 he completed the secondary school at the Greijdanus college in Zwolle. In the same year he enrolled in the 'Business Information Technology' program at the University of Twente. A year later he also joined the 'Industrial Engineering and Management' program at the same university. He specialized in logistics and e-commerce. He did his graduation assignment at B.V. Wavin KLS in Hardenberg, under supervision of dr. E.W. Hans, ir. W. Bandsma, and dr. P.A.T. van Eck. In January 2004 he obtained his MSc (ir.) degrees in Business Information Technology and Industrial Engineering and Management after completing his Master's thesis, entitled 'Mogelijkheden tot structurele verbetering van de concurrentiepositie van Wavin KLS. Afstudeer onderzoek naar aanpassing van de productiebesturing en de informatievoorziening'.

In 2004 he started as a Ph.D. student in the group Operational Methods for Production and Logistics at the School of Management and Technology at the University of Twente. His Ph.D. research was part of the project 'Distributed Planning of Freight Transport Networks using Multi-Agent Technology', funded by the BSIK program Transumo and resulted in this thesis. Since September 2008 he is employed as researcher at the University of Twente.