Coordinated Planning of Inland Vessels for Large Seaports

Shijie Li

Coordinated Planning of Inland Vessels for Large Seaports

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op dinsdag 13 december 2016 om 12.30 uur
door

Shijie LI

Master of Science in Control Theory and Control Engineering,
Harbin Engineering University
geboren te Jingmen, Hubei, China.

This dissertation has been approved by the promotors:
Promotor: Prof. dr. ir. G. Lodewijks
Copromotor: Dr. R. R. Negenborn

Composition of the doctoral committee:

Rector Magnificus	chairperson
Prof. dr. ir. G. Lodewijks	promotor
Dr. R. R. Negenborn	co-promotor

Independent members:

Prof. dr. ir. A. Verbraeck	Delft University of Technology
Prof. dr. C. Witteveen	Delft University of Technology
Prof. dr. R. D. van der Mei	Vrije Universiteit Amsterdam/CWI
Prof. dr. I. F. A. Vis	University of Groningen
Prof. ir. T. Vellinga	Delft University of Technology

The research described in this dissertation is supported by the China Scholarship Council under Grant 201206680009.

TRAIL Thesis Series T2016/23, the Netherlands TRAIL Research School
P.O. Box 5017
2600 GA Delft, the Netherlands
Phone: +31 (0) 15 278 6046
Email: info@rsTRAIL.nl

Published and distributed by: Shijie Li
E-mail: lishijie_123@outlook.com

ISBN 978-90-5584-216-2

Printed in the Netherlands

"No man ever steps in the same river twice, for it is not the same river and he is not the same man."

—Heraclitus (535–475 BC)

To my parents and my fiancé Jialun Liu

Preface

It was October 2012 when I first set foot on Dutch soil and started my PhD study. The most impressive moments that I have had in those initial days were the heavy rains in which I was biking without a waterproof jacket. When I look back at my four years in the Netherlands at the end of the PhD journey, I realize that those days were half sunny and half rainy, just like the weather here. It is such a wonderfully bittersweet experience. Now I would like to express my sincere thanks to the people that are involved in such an important journey of my life.

First of all, I would like to thank my promotor Prof. Gabriël Lodewijks for providing me the opportunity to work in the Section of Transport Engineering and Logistics at the Department of Maritime and Transport Technology. Although he has a very busy schedule, he always gave valuable comments and suggestions on my research. I really appreciate his help in speeding up the procedure required for the defence. Meanwhile, I would like to thank my daily supervisor Dr. Rudy R. Negenborn. Dear Rudy, thank you for your patient supervision and assistance. You always use positive words and encourage me to try new ways when I met with problems in my research. You also showed me how to be a good supervisor in our experiences of supervising master students and how to network with people in joining conferences. Those experiences are quite valuable to me. I am also grateful that you always read every paper that I wrote thoroughly and give detailed feedback to me, and correct every grammar mistake that I have made.

In addition, I would also like to thank Dr. Thomas Léauté for his constructive suggestions and help on distributed constraint optimization and the FRODO2 toolbox. Those suggestions are very helpful when I was still a newbie to the DCOP world.

Secondly, I gratefully acknowledge the grant from the China Scholarship Council. I am also thankful to Harbin Engineering University, China and my master project supervisors Prof. Guoqing Xia and Dr. Yuanhui Wang for their support and recommendation. Thirdly, I would like to thank all colleagues in the Section of Transport Engineering and Logistics, as well as the other friends in Delft. Special thanks to my lunch-mates and travel-mates, Qu and Xiao, Wenhua for the happy times that we had together. I wish you all the best in your future accomplishments. Many thanks also goes to the secretaries: Jozephina, Patty, and Dineke for always being kind and helpful to me.

Many thanks to all the friends in ISC Delft, especially the choir. We had such a lot of enjoyable times playing music and singing songs. Thanks for having me as the piano player in the choir. I also sincerely thank my friends that are not in the Netherlands for their support and encouragement through e-mails, telephones and Wechat. Finally, many thanks to my parents, Xiwei Li and Shunxian Wang for their unconditional support and love. Thank you for making me who I am today. Meanwhile, a lot of thanks to my beloved fiancé, Jialun Liu

for his care and love, my life is never boring with you by my side.

Shijie Li,
Delft, August, 2016.

Contents

Chapter 1

Introduction

1.1 Development of transport over water

For centuries, transport systems have been developed for moving cargo and passengers from one location to another. Transport systems used to be based on roads in the beginning. Later on, alternative modes started being developed and used, including transport over water, over rail, and through the air [224]. To integrate the advantages of different transport modes while reducing potential disadvantages, intermodal transport chains were created by combining them. With the increasing cargo flows, road congestion, and the pressure towards less emissions, the concept of intermodal transport is stimulated [211]. The EU has promoted the provision of the Trans European Network (TEN) and the implementation of support policies for intermodal transport in order to address environmental problems [281].

With the developments in Information and Communication Technology (ICT) in recent years, a new concept has been proposed: synchromodal transport. This type of transport is closely related to intermodal transport. The main difference exists in the fact that the logistic service provider (LSP) of synchromodal transport deploys different transport modes flexibly and dynamically based on real-time information, traffic conditions, and resource availability, while the LSP of intermodal transport deploys different transport modes in advance [211]. Consequently, synchromodal transport is able to ensure more sustainable operations, better use of resources and infrastructure, and provide higher cost efficiency [304]. To promote synchromodal transport, it is important to improve the flexibility and reliability of different transport modes, so that they can be better integrated into the sychromodal logistic chain.

Compared with other transport modes, transport over water ensures higher level of safety, less CO_2 emission per ton, and has the capability of handling large volume of cargoes without congestion [99]. With the trend towards less-polluting and sustainable transport solutions, the European Commission aims to strengthen the competitive position of transport over water, especially inland waterway transport, and to facilitate its integration into synchromodal logistic chain [98–100]. Using the potential of inland waterway transport could significantly contribute to achieve the "EU2020" Strategy and the EU transport policy targets of the European Commission [100]. In addition, to alleviate the congestion on roads and railways, as well as reduce pollutant emissions, the Dutch government also aims for an

Figure 1.1: A typical inland container vessel sailing in the port of Rotterdam (source: Port of Rotterdam Authority).

increase in the proportion of transport over water [157].

Seaports are crucial nodes in international trade and transport [282]. Some of the cargoes arriving at seaports are transshipped to other ports, while others are transported to inland destinations [281]. Large seaports usually consist of multiple terminals serving container vessels, railways, and other forms of hinterland transportation. In the port of Antwerp, for example, nearly 40% of transport to and from the port is by inland container vessels and every week around 925 inland vessels moor in the port in 2014 [245]. In the port of Rotterdam in 2015, 7,386,528 containers were handled [247], and 30% of them used inland waterway transport [281]. In addition, 10,613 sea-going vessels and 77,000 freight inland vessels have moored in the port in 2015 for transporting cargo [248]. Figure 1.1 shows a typical inland container vessel sailing in the port of Rotterdam. Figure 1.3 shows the average number of vessels that arrive at and departure from the port of Rotterdam during a typical day; 23.65% of them are cargo vessels. To improve handling of current and future container flows, the Port of Rotterdam aims to raise the use of waterborne transport to have the largest modal share over the next 20 years [246], and the Port of Antwerp aims to raise the share of container barge transport to 42% by 2020.

Over the last decade, the inter-port competition has moved to the competition between transport chains. Therefore, port authorities need to be more proactive in improving their hinterland strategies [133, 228, 281]. Efficient handling of inland container vessels in the port improves the performance of the hinterland service of the port, and makes it more attractive to port customers and encourages them to make more use of waterborne transport by inland vessels [89, 133, 158, 226]. To achieve that, intense collaboration and coordination between inland vessels and seaports are required. Moreover, with the extended use of ICT nowadays, vessels, container terminals, and port authorities are able to get more accurate information with respect to vessel positions, terminal equipment status, and port services in real-time. Consequently, there is an increasing need to investigate how to improve the hinterland services of large seaports based on the real-time information that is now available.

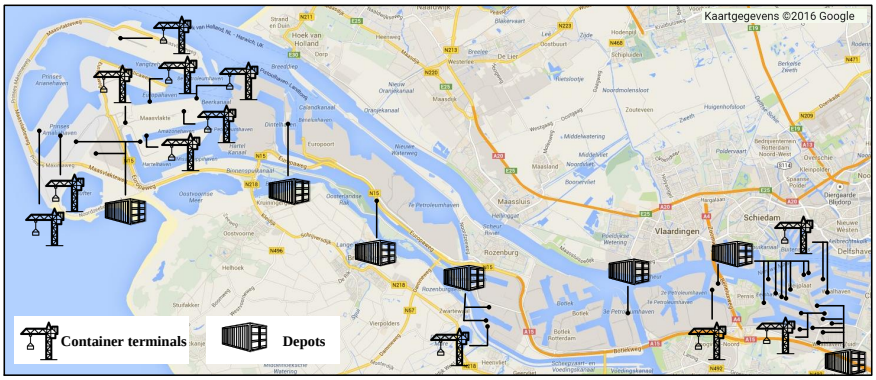

Figure 1.2: Container terminals and depots in the port of Rotterdam (source: Port of Rotterdam Authority).

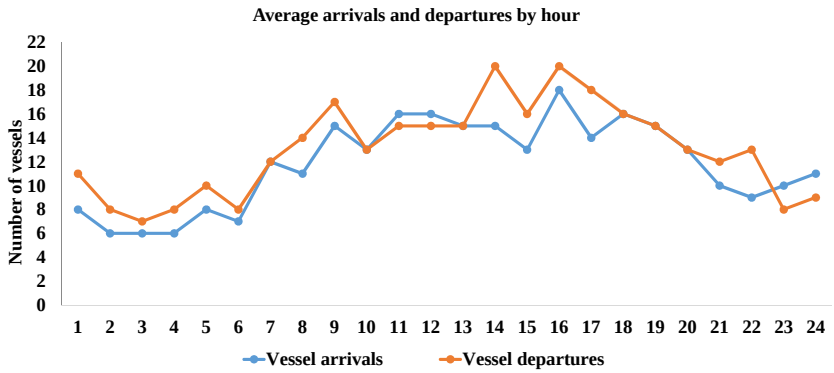

Figure 1.3: Average arrivals and departures in the port of Rotterdam during a typical day [248].

1.2 Development of ICT and optimization techniques

Recent developments in information and communication technology (ICT) provide real-time information exchange and visibility, as well as improved flexibility to react to unexpected changes that happen during the transport process [126]. These benefits could lead to improvements in the efficiency and security levels of transport over water.

Firstly, ICT systems facilitate the immediate availability of information. For example, Automatic Identification System (AIS) [125] is a system that is installed on many vessels that can automatically send and receive information on a vessel's name, position, speed and course. With its obligatory use on vessels, the level of information transparency and recognition would be increased, and thereby improve the safety level of the vessels and ensures smooth traffic management.

Secondly, ICT systems also improve the communication between operators involved in the process of transport over water. In the port of Rotterdam, any vessel in the nautical control area must listen to the correct Very High Frequency (VHF) sector channel [249]. Vessel operators can use the VHF channel to communicate two traffic centers in the port. With a traffic control system called Vessel Traffic Services (VTS) [249], the traffic centers provide continuous information about the current situation of the shipping traffic to vessel operators. In the port of Antwerp, a communication system called Barge Traffic Sytem (BTS) [245] has been implemented, through which vessel operators are able to give advance notice of their arrival at a terminal, request a time slot or pass on other information to a terminal operator. Based on the information received from barge operators, the terminal operator schedules the loading and unloading operations, and send the schedules back to barge operators through BTS.

Thirdly, ICT systems strengthen the collaboration between different operators in the container transport chain via increased information sharing. The port of Rotterdam has developed an on-line intermodal planner called InlandLinks [250] for shippers and logistics service providers, in order to help them planning the movements of containers.

Another major development in the ICT sector is the Internet of Things (IoT). In IoT, small electronic devices provide local intelligence to everyday physical things, and connect them to the cyberspace on the Internet [160]. Adopting IoT in transport over water can also lead to improvements both on the quality of transport services and innovation of infrastructure.

Firstly, with the real-time tracking of containers and vessels, terminal operators can make their schedules efficiently and flexibly. In addition, the re-planning of vessel schedules is made possible by the availability of real-time data, and the negative impacts of unexpected disturbances can be alleviated. Furthermore, it is easier to track the containers that are in transit. Consequently, the logistic service providers could make changes before the arrival of containers at their destination, which gives them flexibility in managing their supply chains. This would also promote better integration of the transport over water into the sychromodal transport chain. Secondly, IoT also contributes to the development of automated terminals, as the increased level of exchanged information improves the interactions between terminal infrastructures, and gives terminal equipment more information to use for their executions. A review of potential benefits of ICT in transport can be found in [126].

ICT innovations offer the possibility to coordinate the planning of transport over water, and many theoretical optimization techniques are also made available. Although there

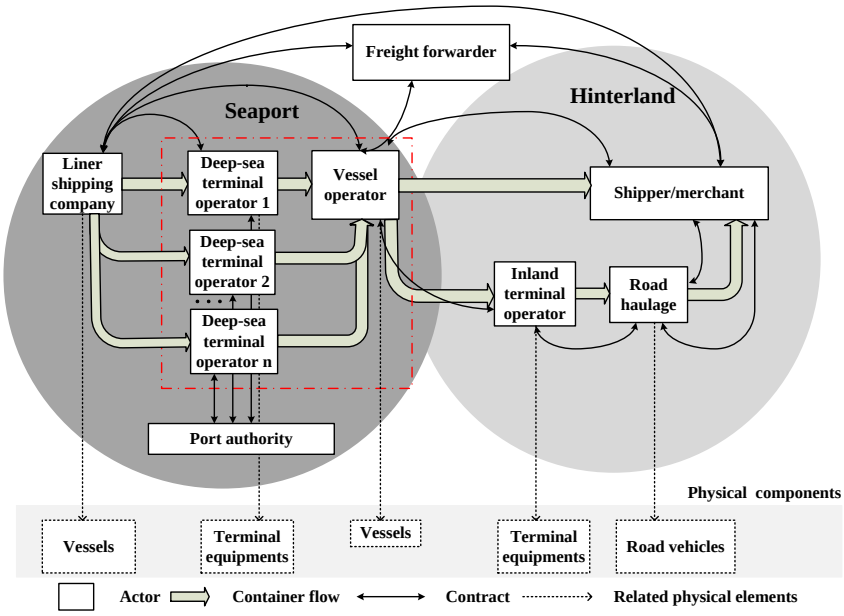

Figure 1.4: Actors and correlations in hinterland transport chain and seaport (adapted from [133]).

are thousands of journal articles addressing different optimization techniques for planning problems, they are distributed across different problem settings and research communities. The most common optimization techniques in literature include mathematical programming, constraint programming, continuous global optimization and heuristic methods [132]. These optimization techniques could be very useful in realizing the improved waterborne hinterland services. It is therefore crucial and challenging to determine how to make that link. A further discussion on the applicability of existing optimization techniques for planning problems of transport over water will be given in Chapter 2.

1.3 Problem statement

Figure 1.4 describes the actors and correlations between the hinterland transport chain and seaport. Terminal operators usually refer to the companies that operate terminals and offer services to the transshipment and temporary storage of containers. Vessel operators usually refers to the companies that schedule container transport to and from the hinterland. These companies usually do not own inland container vessels themselves, but contract the companies that own and manage vessels. A shipper/merchant refers to the organization that the containers will be transported to, or the organization that owns the containers, or the one that starts the container transport. A freight forwarder refers to the company that schedules container transport on behalf of the shipper. Port authority refers to the organization that leases sites to port-related business, whose responsibility includes the management of port infrastructure and other facilities in the port area, and ensure efficient and safe shipping

traffic.

A liner shipping company usually refers to a company whose core business concerns the organization of sea-going transport of containers, sometimes the shipping company also organizes the hinterland transport of containers. When the liner shipping company only organizes the sea-going transport of containers, and the shipper/merchant organizes the hinterland transport by himself, this situation is referred as merchant haulage. When the liner shipping company organizes both the hinterland and sea-going transport of containers, this situation is referred as carrier haulage. In addition, a small percentage can be classified as terminal haulage, where the terminals make the decisions about the hinterland transport.

The contracts that are required to transport containers depend on the shipper's choice of merchant haulage or carrier haulage. If the shipper chooses merchant haulage, then the hinterland transport of the containers has to be organized by himself. In this case, the shipper contracts directly inland container vessel operators or truck operators, and the liner shipping company. The shipper may also ask a freight forwarder to arrange the transport. If the liner shipper chooses carrier haulage, the shipper then contracts the liner shipping company, who arranges both the hinterland and ocean transport. Different contractual relations between the liner shipping company, the shipper/merchant, the terminal operators and the vessel operators are established, according to who organizes which part of the container transport.

The liner shipping companies always have contractual relations with the terminals, with which they make agreements about the transshipment of containers from a sea-going vessel to a successive hinterland transport modality (truck, train or inland container vessel) and vice versa. Meanwhile, the vessel operators are contracted by either the carriers (in carrier haulage) or by the shippers/merchants (in merchant haulage). There is no contractual relations between terminal operators and inland container vessel operators in both carrier and merchant haulage. This implies that inland container operators do not need to pay the terminal operator for the transshipment of containers, and both of them therefore cannot charge each other even if the agreements are not carried out satisfactory [89].

For years, inland container vessels have been facing time and efficiency constraints when calling at different terminals in seaports [157]. Two coordination problems exist in the planning of inland vessels in large seaports: firstly, the long stay in the port and secondly, the insufficient terminal and quay planning with respect to the sailing schedules of sea-going vessels and inland vessels [133].

Every time an inland container vessel enters the port, it visits multiple terminals spread over the port area. The sequence of how the inland vessel visits different terminals is defined as a rotation [89]. As an example, Figure 1.2 presents a map of the container terminals and depots in the port of Rotterdam. Since many inland container vessels may visit the same terminal, congestion and waiting times are inevitable [158]. Currently, inland container vessels in large seaports are planned bilaterally and usually on an ad hoc basis [226]. In practice, the inland vessel operator makes calls to the terminal operator and makes appointments on the agreed time window in which the vessel can be handled to load and unload containers [89, 133, 158, 226]. The process of phone calls back and forth takes up an unnecessarily large amount of the planners time [89, 133, 226]. In addition, when a delay at a terminal happens, the vessel's agreed time window at the next terminal will be missed. Vessel operators have to make allowance for such events by inserting large margins when planning their visits to terminals, otherwise the reliability of the transport service might be undermined [158]. Vessel operators try to plan efficient trips, but delays in the handling process means

that they may have change their schedules for terminal visits every now and then.

For example, in the port of Rotterdam, it is reported that in 59% of the barge visits the actual start of handling deviates more than 2 hours from the originally planned time windows, and the average time a vessel spends in the port varies from 21 hours for small vessels (ship length <85 m) up to 36 hours for large vessels (ship length >110 m) [226]. Moreover, the average waiting time of an inland container vessel for visiting a terminal is about 1 hour, but waiting times of up to a whole day are also possible [241].

On the other hand, as terminal operators have contractual relationships with the deep sea carriers, sea-going vessels currently have absolute priority over inland vessels at terminals, and inland vessels are scheduled after sea-going vessels have been handled [89, 133, 158, 226]. This can further increase the waiting time of inland vessels at terminals. Long waiting time implies loss of time and money, and could undermine the competitive position of a port's hinterland services. Insufficient planning at quays and terminals makes transport of inland vessels unreliable and unpredictable in the ports. To conclude, firstly, these two coordination problems make inland waterway transport costly for inland vessel operators, who can only generate income by sailing to the hinterland. Time savings in the port could therefore prove to be very valuable to vessel operators [158]. Secondly, the relatively long periods of time spent in the port have a negative influence on the turnaround times and the total cost of inland vessel services and thus undermine the competitiveness of inland waterway transport [158]. Thirdly, it also affects the quality of hinterland services of the a seaport, which could undermine its competitiveness. As the chance of queues increases, terminal equipment and inland vessels are not fully utilized.

Therefore, efficient handling of inland container vessels in the seaports is crucial to significantly reduce waiting time and turnaround times in the seaports and enable a higher capacity utilization of inland vessels, as well as improving the reliability and the efficiency of inland waterway transport from seaports to hinterland and vice versa.

In practice, inland vessel operators are in competitive positions and are unwilling to share their information with each other. Therefore, it is important to investigate in what way the inland vessel operators can be motivated to cooperate and share information with one another. For this, two levels of cooperativeness are identified, including partially-cooperative and fully-cooperative. Partially-cooperative means that vessel operators only share part of the information with respect to the arrival and departure time at different terminals. Fully-cooperative means that vessel operators are willing to share all information to get better rotations.

Meanwhile, it is also important to investigate how the ICT-enabled optimization techniques could benefit the coordination of vessels and planning of containers, and thereby facilitate flexible planning of transport over water, so that this transport mode can be better integrated into the synchromodal transport chain. Moreover, efficient handling of inland vessels could also contribute to the inter-terminal transport (ITT) in large seaports. ITT represents the movement of containers that are transferred between terminals within the port when they are transshipped between the same or different modes of transportation [95, 273, 307], and these containers are referred to as ITT containers in this thesis. By making use of the available space on inland vessels when they are transporting between terminals, the inland vessels can also be used to transport ITT containers, which could be a potential solution for alleviating the congestion of ITT on roads.

1.4 Research questions and approach

To facilitate efficient handling of inland container vessels in large seaports, the main research question addressed in this thesis is:

How can different coordination methods be used to improve the reliability and efficiency of inland container vessel transport in seaports?

Here, reliability is evaluated based on the deviation of the actually executed vessel schedules from the originally planned schedules, and efficiency is evaluated based on the total time that inland vessels spend in the port area. To address this main question, the following Key Research Questions need to be answered:

1. What performance indicators should be used to evaluate the reliability and efficiency of inland vessel transport in seaports?
2. To what extent can the inland container vessels be better coordinated considering different levels of cooperativeness?
3. How can the planning of inland container vessels contribute to inter-terminal transport (ITT) in large seaports?
4. How can the proposed coordination methods help practitioners making decisions?

The main physical elements that are considered in this thesis include terminals and inland vessels. In practice, there are vessel operators and terminal operators. Although one terminal operator can operate more than one terminal and one vessel operator can operate more than one vessel, without loss of generality, this thesis assumes that every terminal operator operates exactly for one terminal and one vessel operator operates exactly one vessel. The interactions between the operators are shown in Figure 1.5. Vessel operators communicate with terminal operators only to make appointments for planning loading and unloading operations and do not communicate with the other vessel operators.

To answer the research questions, this thesis proposes three conceptual frameworks to formulate the problem, as shown in Figures 1.6, 1.7 and 1.8. The concept of a multi-agent system (MAS) is adopted in designing the conceptual frameworks. There exist many definitions for describing agents; this thesis uses the definition proposed in [320], which considers an agent as a computer system that is capable of independent action on behalf of its user or owner, and considers a multi-agent system as consisting of a number of agents that interact with each other, typically via the exchange of messages. Consequently, the thesis assumes that for each physical element (a terminal or an inland vessel), there is a local agent that controls the physical element's operations and sends information to the other agents.

Based on the willingness of a vessel operator to share information with other vessel operators, two levels of cooperativeness are identified, including partially-cooperative (Figure 1.6 and Figure 1.7) and fully-cooperative (Figure 1.8). In addition, different coordination schemes are also considered, including single-level interaction (Figure 1.6) and multi-level interactions (Figure 1.7 and Figure 1.8). Single-level interaction refers to a situation in which there is no overall coordinator, each local agent communicates and sends its information in a distributed way. For multi-level interactions, there is a overall coordinator that sends and receives information from local agents, and searches for globally optimal solutions. Partially-cooperative planning with multi-level interactions means that each local agent firstly solves local optimization problem for each physical element, and sends partly

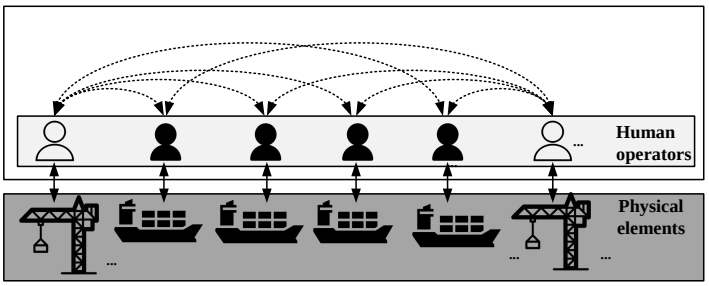

Figure 1.5: Interactions between vessel operators and terminal operators in practice.

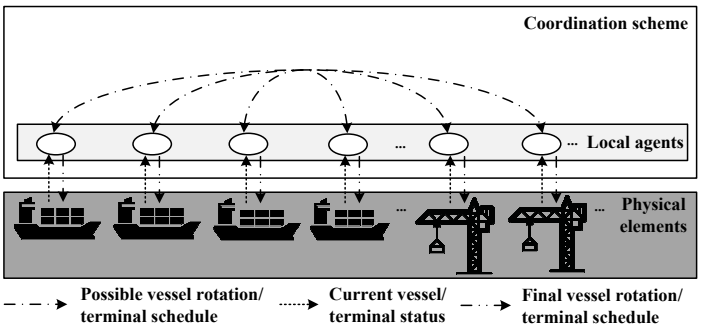

Figure 1.6: Partially-cooperative planning for single-level interaction.

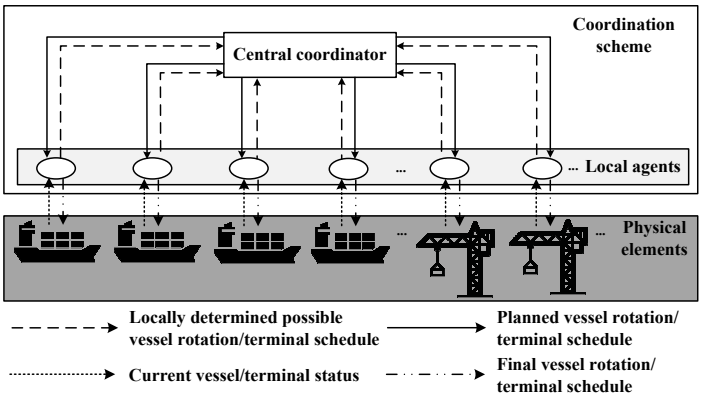

Figure 1.7: Partially-cooperative planning for multi-level interactions.

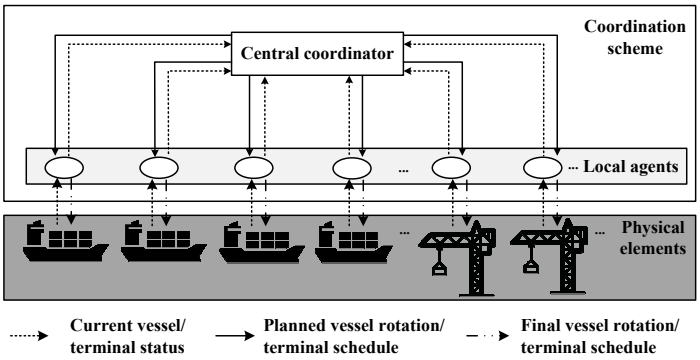

Figure 1.8: Fully-cooperative planning for multi-level interactions.

its information to the overall coordinator, after which the coordinator searches for globally optimal solutions. Fully-cooperative planning with multi-level interactions means that the local agents do not solve their local optimization problems, but instead send their information to the overall coordinator, after which the coordinator determines globally optimal solutions.

A distributed coordination scheme as shown in Figure 1.6 has the advantage that the vessel agents could only reveal information partly to the other agents, which ensures certain degrees of information privacy. However, it also has the disadvantage that the overall solving process could be slower than a centralized coordination scheme. This is because the coordination problem considered in this thesis requires a considerable amount of information exchange between terminal agents and vessel agents, and a distributed coordination scheme would involve a considerable amount of message exchange with the increase of problem sizes. This also implies that for large-scale problems a distributed scheme could cost a much longer time to find appropriate solutions. Therefore, exact and distributed constraint programming methods are proposed to solve the coordination problem mainly for small ports, with the framework shown in Figure 1.6.

For medium and large ports, the complexity of the problem increases substantially with the increase of vessels and terminals involved. Thus, it is difficult to solve the problem to optimality in a reasonable amount of time. Approximate methods are proposed to solve the problem formulated in Figures 1.7 and Figure 1.8. For Figure 1.7, solution methods that combine mathematical programming, constraint programming, and heuristic methods are proposed to solve the problem in medium ports. For Figure 1.8, a hybrid solution method that uses Benders decomposition and a large neighborhood search heuristic is proposed solve the problem in large ports.

In this way, optimization techniques from different research communities are investigated, including constraint programming, mathematical programming, and heuristic methods, to solve problems formulated with different conceptual frameworks.

1.5 Thesis outline

Figure 1.9 illustrates the relations among the chapters of this thesis, and an ordering in which the chapters can be read. To answer the research questions, the chapters are organized as follows:

- In **Chapter 2** a literature review on the operational planning problems in large seaports with respect to the handling of inland vessels is presented. The coordinated planning problem of inland vessels is compared with the other traditional planning problems, in which a literature review on the possible solutions methods is also given. In addition, the performance indicators and benchmark systems that are used later in this thesis for evaluating the proposed coordinated planning strategies are defined. This chapter answers Key Research Question 1.

- In **Chapter 3** a partially-cooperative planning strategy for single-level interaction in small ports is proposed. An exact approach based on distributed constraint optimization (DCOP) is studied. The problem is formulated as a DCOP by considering the vessels and terminals as agents, the constraints on vessels and terminals as individual

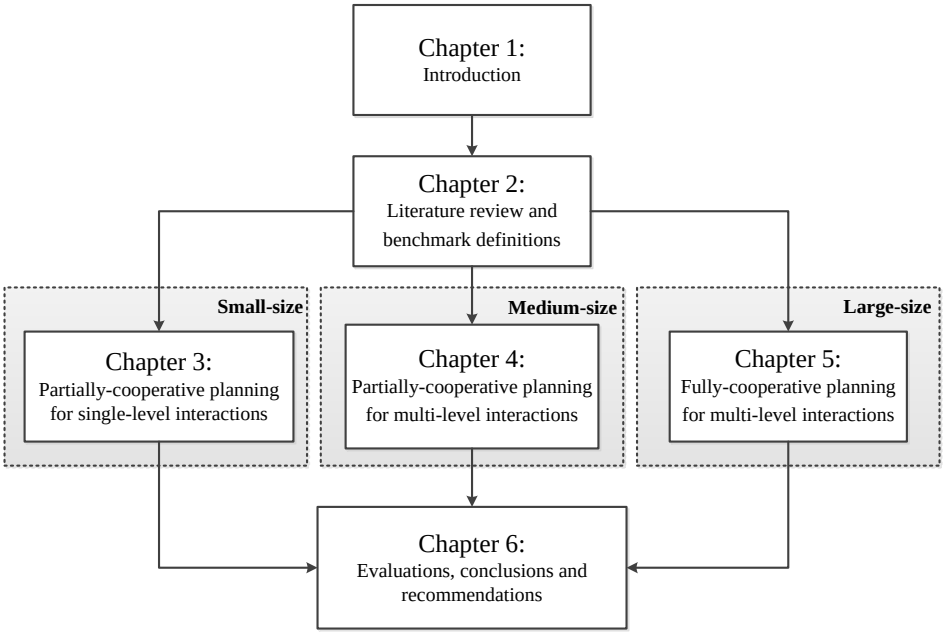

Figure 1.9: Road map. Arrows indicate read before relations.

utility functions of the corresponding agents, the constraints that involve variables from different agents are considered as inter-agent utility functions. The objective of formulated DCOP is to maximize the sum of values of the individual utility functions for each vessel, and the inter-agent utility functions among different vessels. Different utility values represent different preferences for visiting terminals at pacific time slots. Consequently, maximizing the sum of the utility functions means satisfying the preferences of all the vessels as much as possible. Two solution methods based on a single layer and a multi-layer structure are proposed. For each solution method, four different optimization algorithms are evaluated for solving DCOPs, aiming at studying how the algorithms perform with increasing problem sizes. Evaluation of the methods are based on the size and quantity of messages exchanged, computation time, and quality of solutions. This chapter answers partially the Key Research Question 2.

- In **Chapter 4** a partially-cooperative planning strategy for multi-level interaction in medium ports is proposed. A two-phase planning approach is proposed after taking into account optional inter-terminal containers and several practical constraints. In the first phase, a single vessel optimization problem is solved locally using MIP for each vessel, with the objective to minimize the total time spend for loading and unloading the required number of containers at each terminal in the port, while transporting as many inter-terminal containers as possible. As the optimal rotation plans obtained may conflict with each other, the objective of the second phase is to reduce conflicts and minimize the total time that inland vessels spend in the port. Three types of solution methods based on coordination rules, constraint-programming and large

neighbourhood search (LNS)-based heuristics are proposed to solve the coordination problem. This chapter answers partially the Key Research Question 2 and 3.

- In **Chapter 5**, a fully-cooperative planning strategy for multi-level interaction in large ports is proposed. The vessels shares all the information and cooperate with each other also to transport mandatory inter-terminal containers besides their own hinterland containers. A hybrid solution method based on logic-based Benders decomposition and LNS is proposed. The objective is to minimize the total time that the inland vessels spend in the port. Benders decomposition is used to split the problem into a rotation generation master problem and several rotation evaluation sub-problems, and LNS is introduced to solve the master problem for large problem instances. Possible disturbances that may happen in practice are also considered, including the failure of terminal equipment, and sudden closing of terminals due to extreme weather conditions. Whenever accidents happen, the vessels will be re-planned based on the up-to-date information. This chapter answers partially the Key Research Question 2 and 3.

- **Chapter 6** summarizes the results of this thesis and outlines directions for future research. This chapter answers the overall research questions.

Chapter 2

Literature study and benchmark definition

Chapter 1 has indicated that the objective of this thesis is to facilitate efficient handling of inland vessels in large seaports. To reach this goal, firstly, this chapter clarifies which level this thesis focuses on, as there exist different levels of planning problems in large seaports that are relevant to inland vessel transport. As the operational planning level involves more interactions among terminal operators and vessel operators, and also benefits more from the accessibility of real-time information exchange than the other levels, this thesis focuses on the operational planning problems that are relevant with the handling of inland vessels. Secondly, this chapter compares the coordination problem considered in this thesis with other traditional planning problems including vehicle routing problem and ship routing and scheduling problem. An analysis of the applicability of the existing solution methods from these traditional planning problems, to the problem considered in this thesis, is also given. Thirdly, to evaluate the proposed solution methods in Chapters 3, 4 and 5, this chapter defines key performance indicators and benchmark systems.

This chapter is organized as follows. Section 2.1 provides an overview of planning problems at different levels in ports. A literature review on the operational planning problems that are relevant to inland vessel transport is given in Section 2.2. Section 2.3 compares the coordination problem considered in this thesis with other similar traditional planning problems, including the vehicle routing problem and ship routing and scheduling problem. The applicability of the existing solution methods of these traditional planning problems to the coordination problem of this thesis is discussed in Section 2.4. Moreover, the key performance indicators and the benchmark systems that are used throughout this thesis are defined in Section 2.5 and Section 2.6.

2.1 Different levels of planning problems in ports

Based on [118], this chapter categorizes the planning problems that exist in large seaports according to the corresponding time horizons, into the following four levels:

- Strategic planning is the highest level of management and requires large capital in-

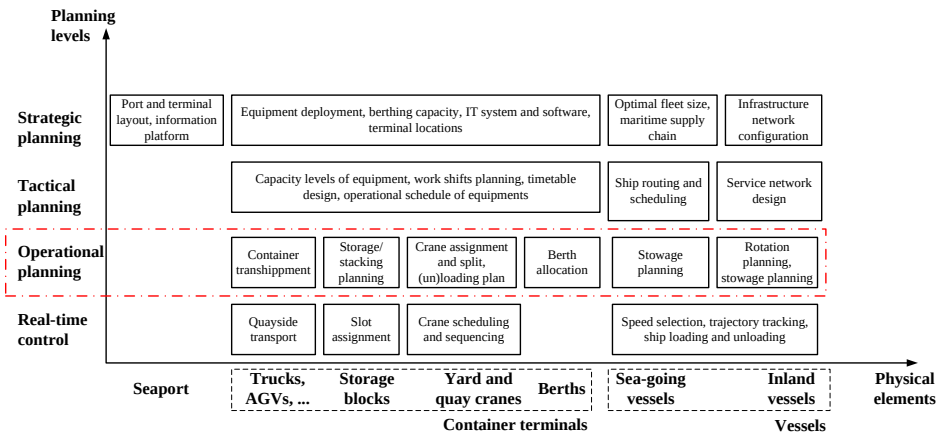

Figure 2.1: Planning problems categorized over different levels and involved physical elements.

vestment over long time horizons (years). Decisions at this planning level provide certain objectives and constraints for the operators in lower levels;

- Tactical planning ensures efficient and effective allocations of existing resources, and organize operations according to strategic objectives, in order to improve the performance of the whole system over a medium-term time horizon (several days to months). Decisions made at this level generally influence the activities made at the operational planning level and real-time control level;

- Operational planning concerns the short-term day-to-day operations. It defines what an operator is actually planning to do. It may be influenced by future changes in the transport system, for example, traffic conditions, or the new arrival of transportation requests. Decisions made at this level may have to be revised when actual conditions change, i.e., when unknown data becomes known expectedly;

- Real-time control reacts on discrepancies between planned and actual state of a physical system. Activities at this level depend on the decisions that are made at the higher levels and the availability of real-time information. Therefore, planning at this level relies on the information flow, for example, the vessel or container position, or the status of equipment such as quay cranes or yard trucks, to decide on the operations.

As this thesis focus on the waterborne transport in the port, the major physical elements that are involved include sea-going and inland vessels, container terminals, and the port itself. Figure 2.1 presents the different planning problems with respect to these major physical elements. Within a container terminal, four types of physical elements are involved in transporting the containers from/to the vessels, including the berths, yard and quay cranes, storage blocks, and vehicles for shore-to-stack/stack-to-shore transport (trucks, AGVs, ..., etc.).

The planning problems of a seaport mainly involve the strategic level, including the layout design of the port and terminals, as well as building port-wide information platform.

The planning problems of a container terminal exist at different levels. Strategical planning problems include the deployment of terminal equipment, development of terminal-wide IT systems and software, layout design of terminals, as well as deciding terminal locations; tactical planning problems include designing operational schedules and timetables for equipments, work shifts of human operators, as well as determining the capacity level of terminal equipment. Planning problems of the major physical elements within a container terminal exist in operational planning and real-time control levels. For shore-to-stack/stack-to-shore vehicles, the operational planning problems include the transshipment of containers, and real-time control concerns the tracking and scheduling of these vehicles. For storage blocks, the operational planning problems include the storage and stacking planning of containers, and real-time control concerns the real-time slot assignment of containers. For yard and quay cranes, the operational planning problems include the assignment and split of cranes and deciding the (un)loading plans for vessels, and real-time control includes the real-time sequencing and scheduling of cranes. For berths, the planning problem exists in operational planning level, which concerns the berth allocation of upcoming vessels.

For sea-going and inland vessels, real-time control problems are the same, including the speed selection, trajectory tacking, loading and unloading of vessels, while the planning problems at the other levels are different. On the strategic level, the planning problems of sea-going vessels concern determining the optimal fleet size and designing maritime supply chains, while the planning problem of inland vessels concerns infrastructure network configuration on inland waterways. On the tactical level, the planning problem of sea-going vessels concerns the routing and scheduling of vessels, while the planning problem of inland vessels includes the design of the intermodal service network. On the operational level, planning problem of sea-going vessels concerns the stowage planning on the vessel, while the problems of inland vessels include both the stowage planning and the rotation planning.

As indicated earlier in Chapter 1, this thesis aims to investigate how the real-time information exchange that is made possible by ICT could best benefit the coordination of inland vessels. Therefore, this thesis focuses on the operational planning level as this level involves more interactions between different vessel operators and it also benefits more from the accessibility of real-time information exchange than the other levels.

Therefore, a review of the operational planning problems in ports is given in the next section. For literature reviews on planning problems at the other levels, it is referred to [227, 228, 323] for strategic planning problems of ports. For strategic and tactical planning problems of container terminals, sea-going vessels, and inland vessels (barges), it is referred to [284–286, 314], [60–62], and [47, 314], respectively. Literature reviews on real-time control problems of sea-going vessels, container terminals, and inland vessels can be found in [60–62, 279], [1, 172, 240], and [13, 194, 220], respectively.

2.2 Operation sequences of inland vessels in a port

The relations between different actors that are involved in hinterland transport chain and seaport have already been introduced in Section 1.3. This section mainly focuses on the operations of inland container vessels within the port area.

The transport process of an inland vessel in a port consists of a sequence of subprocesses, including rotation planning and stowage planning of inland vessels, berth allocation,

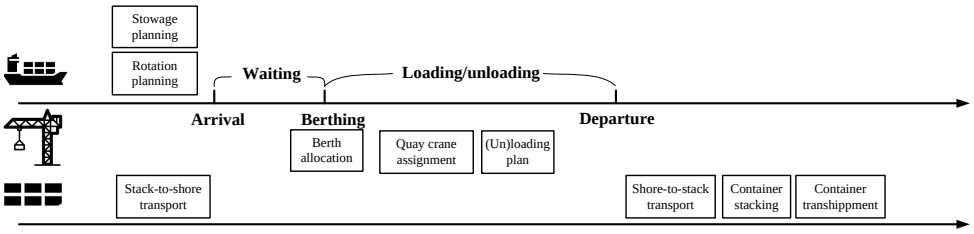

Figure 2.2: Relevant operational planning problems of the inland vessel operations in a seaport.

quay crane assignment and scheduling of container terminals, and container stacking and transshipment within the terminals. Figure 2.2 describes a vessel's time of stay at a terminal, which consists of waiting time, berthing time, loading and unloading time. For simplicity, the preparation time and maneuvering time are included in the loading and unloading time of the vessel. The total time that an inland vessel spends in the port area depends on the efficiency of how the above-mentioned subprocesses are planned and executed by the terminal operators and inland vessel operator. Therefore, in the following sections, these subprocesses are described in detail.

2.2.1 Vessel rotation planning

Before entering the port, an inland vessel operator makes appointments with terminal operators, normally one or two days in advance [89]. It is usually the vessel operator that initiates the communication with the terminal operators, to determine the most convenient time windows for handling the containers. The sequence of how the inland vessel visits different terminals is defined as a rotation [89]. Rotation planning of an inland vessel involves deciding on its optimal visiting sequence to multiple terminals, during the process of which terminal operators and the vessel operator communicate and cooperate with each other based on different interaction protocols and optimization techniques.

The publications that focus on the rotation planning problem of inland vessels are relatively scarce, and a literature review on all publications on VRPP is given as follows.

A multi-agent and distributed planning system is proposed in [209, 219, 276]. This system can construct efficient and realistic rotation plans and improve individual and joint plans of competitive parties with conflicting interests. However, the outcomes of this system sometimes contained rotations that included longer sailing times than needed. This type of rotations actually would not be allowed by a human planner [219]. This system is an off-line planning system, and mostly provides feasible solutions for vessels instead of optimal solutions. The authors also point out that a decentralized structure would be a more promising solution for the different parties that are involved [219].

Therefore, in [89] a vessel rotation planning problem (VRPP) is first proposed, in which the terminal and vessel operators cooperate with each other to obtain better alignment. A distributed multi-agent system that includes interaction protocols based on the waiting time profile and service time profile is proposed. The waiting time profile for every arrival moment gives out the maximum possible waiting time [86], and time-dependent service time profile gives out both gives out maximum possible waiting time and possible service time

[87] from terminal operators to vessel operators, respectively. Vessel operators can use these profiles to determine their rotations, and terminal operators can use these profiles to plan their quayside activities. To improve the applicability of this system, a simulation game is developed to communicate and help practitioners to understand the generated solution [88].

2.2.2 Stowage planning

Stowage planning concerns planning the positions of the containers on a vessel [221], which happens at the same time as rotation planning, since the sequence in which terminals are visited also influences the sequence in which containers are loaded and unloaded. The stowage planning consists of two phases [286]: in the first phase, the shipping lines design the stowage plan including the position for all containers on the vessel and all terminals to be visited; in the second phase, the terminal operators decide on the loading and unloading operations for the handling equipment such as the quay cranes and horizontal transport means.

This problem is important not only for shipping companies but also for terminals, as they are both involved in the loading, unloading, and storage of containers [217]. The stowage planning problem has been studied extensively in literature. Detailed literature reviews can be found in [217, 284, 286].

2.2.3 Berth allocation

After arrival at a terminal, the vessel moors at a specific berth. The berth is usually allocated to the upcoming vessel before its arrival by the corresponding operator at the terminal. Berth allocation refers to the decision process of assigning a berth position to a vessel. It depends on several factors, including the vessel's length, type of cargo, expected time of arrival, loading and unloading operations, the availability of quay cranes, and the berthing and crane requirements of other vessels which have already moored at the quay or are expected to arrive in a short time [151]. The berthing schedules of large sea-going vessels are known about one year in advance, which are sent from the shipping lines to the terminal operators by EDI (Electronic Data Interchange) [286], while the berthing schedules of inland vessels become known relatively closer to their arrival times.

Three types of berths are distinguished by [34, 139] based on their physical characteristics, including discrete, continuous and hybrid berths. In the discrete berth case, the quay is partitioned, and only one vessel can be served at each single berth at a time; a continuous berth refers to a berth in which vessels can berth at arbitrary positions within the boundaries of the quay; a hybrid berth refers to a berth in which the quay is partitioned, but vessels may share a berth or one vessel may occupy more than one berth.

Based on the arrival patterns of vessels, [34, 138] further categorized the berth allocation problem into four types: the static problem refers to the situation in which all vessels have arrived at the terminal and are waiting to be handled; the dynamic problem refers to the situation in which the vessels arrive at the terminal with individual but deterministic arrival times; the cyclic problem refers to the situation in which the vessels call at terminals repeatedly at fixed time intervals according to their liner schedules; the stochastic problem refers to the situation in which the arrival times of vessels are stochastic, either defined by continuous random distributions or by scenarios with discrete probability of occurrence.

The berth allocation problem has also been studied extensively in literature. For detailed literature reviews it is referred to [34, 35, 217, 284, 286].

2.2.4 Quay crane assignment and scheduling

After berthing, quay cranes are assigned to the vessel for loading and unloading operations. Quay crane assignment depends on the accessibility and availability of cranes at the berth. To load and unload a large container vessel, typically several quay cranes are required. Depending on the ship's size, it is common that three to five cranes operate at a sea-going vessel, and feeder ships are operated with one to two cranes [151]. In addition, the corresponding terminal operator also needs to decide on the loading and unloading sequences of containers.

The decisions made when solving the berth allocation problem and the quay crane assignment problem are highly interrelated [35], as they determine the time of a vessel's stay in the port together. Quay crane assignment and scheduling refers to the process of assigning quay cranes to a vessel, and determining the time and sequence of loading and unloading operations of a berthed vessel [285].

This chapter uses the classification scheme of quay crane assignment and scheduling problems in [34], which is based on the way the containers are grouped into crane tasks: a task in which all containers are within a certain area of vessel bays; a task in which all containers are within an individual bay; a task in which a set of containers are within a bay; a task that consists of a single container movement. For detailed reviews it is referred to [34, 35, 217, 284, 286].

2.2.5 Container stacking and transshipment

After the loading and unloading, the vessel either sails to the next terminal or leaves the port according to its planned rotation. The unloaded container from the vessel are transported to storage yards of terminals by transfer vehicles. The containers will then be stored temporarily. Depending on their destinations, they might be transshipped to another vessel or dispatched to the terminal gates in order to be transported by trucks or trains after having been inspected [48]. The main objective of a stacking strategy includes: efficient use of storage space; timely transportation from quay to stack and further destination and vice versa; avoidance of unproductive moves [74]. Therefore, the planning problems during this phase have less impact on the vessel's time of stay in the port area, and are more relevant to storage yard operations and management. More details can be found in [37, 48, 74, 171, 331].

To conclude, most of these operational planning problems with respect to inland vessels in the ports have been extensively investigated with numerous literature, except that there are relatively fewer publications on the vessel rotation planning problem. This implies that the VRPP requires more attention than the other operational planning problems from the perspective of ensure efficient handling of inland vessels. It also means that with fewer literature on VRPP, much more improvements could be made for solving this problem. Therefore, this thesis aims to improve the reliability and efficiency of inland vessel transport in the port by planning efficient rotations for inland vessels with different coordination methods.

2.3 Relations with traditional planning problems

As the VRPP has not be extensively studied in literature, there are few directly applicable solution methods. On the other hand, the VRPP shares several similarities with other traditional planning problems, such as vehicle routing problem and ship routing and scheduling problem. Therefore, this section first presents literature reviews on these problems and then compares the VRPP with the traditional planning problems .

2.3.1 Vehicle routing problem

The Vehicle Routing Problem (VRP) was first introduced by [71] as a generalization of the Traveling Salesman problem (TSP) presented by [108]. Here this chapter uses the definitions of [242]: the VRP is generally defined as a graph $\mathcal{G} = (\mathcal{V}, \varepsilon, \mathcal{C})$, where $\mathcal{V} = \{v_0, \ldots, v_n\}$ is a set of vertices; $\varepsilon = \{(v_i, v_j) | (v_i, v_j) \in \mathcal{V}^2, i \neq j\}$ is the arc set; and $\mathcal{C} = (c_{ij})_{(v_i, v_j) \in \varepsilon}$ is the cost matrix defined over ε, representing distances, travel times, or travel costs. Vertex v_0 is called the deport, while the remaining vertices in $\mathcal{V}$ represent the location of the customers (or requests) that need to be served. The VRP consists of finding a set of routes for K identical vehicles based at the depot, in which each of the vertices is visited exactly once, and the overall routing cost in minimized.

The VRP refers to a generic class of problem that involves the design of optimal routes for a set of vehicles, in order to serve a set of customers with different side constraints. Several variants of the VRP exist, depending on the type of cargoes that needs to be transported, the type of services required, as well as the types of customers and vehicles. Among these VRP variants, the most relevant variants are capacitated vehicle routing problem (CVRP), vehicle routing problem with time windows (VRPTW) and vehicle routing problem with pickup and delivery (VRPPD). The CVRP refers to the type of VRP in which each vehicle can perform at most one route and the total demand of the customers visited by a route cannot exceed the vehicle capacity [17, 20]. The VRPTW refers to the type of VRP in which each customer needs to be visited within a specified time window [17, 20]. The VRPPD refers to the type of VRP in which each transportation request is associated with an origin and a destination [29, 232]. Table 2.1 concludes the solutions methods that have been used to solve these VRP-variants in literature. As there are numerous papers on these problems, this table only includes the most original publications in major journals, conference papers and reports are therefore not included.

The VRPP shares several similarities with these VRP-variants. Firstly, they are all concerned about constructing routes for certain vehicles with the aim of minimizing particular objective functions. Secondly, the constraint that each vertex will be visited exactly once by all vehicles is also directly applicable to represent the constraint that ensures each terminal will be visited exactly once by each inland vessel. Thirdly, all these VRP-variants need to consider the capacity constraint of vessels/vehicles, and they all need to consider the time window during which the loading and unloading operations is available.

2.3.2 Ship routing and scheduling problem

The ship routing and scheduling problem (SPSP) is a generic class of problem that focuses on sea-going vessels, in which routing refers to deciding on the sequences of ports of call to

Table 2.1: Overview of the solution methods that have been used in relevant VRP-variants.

	Method type	Optimization algorithm
CVRP	Exact methods	Branch-and-cut [12, 16, 144, 175, 198, 222, 256]; branch-and-price [59, 272]; branch-and-cut-and-price [109, 197, 234, 235]; constraint programming [123].
	Approximate methods	Adaptive memory programming [293, 297, 298]; ant colony optimization [44, 45, 84, 170, 208, 260, 261, 291, 325]; column-and-cut [19]; graphic processing units-based [274]; genetic algorithm [5, 6, 15, 30, 142, 204, 212, 253]; GRASP [94, 202, 254, 287]; large neighborhood search [72, 264, 309]; neural networks [67, 116, 120, 214, 311]; particle swarm optimization [53, 203, 205]; route-first-cluster-second heuristic [25, 36, 104, 225]; simulated annealing [91, 92, 192, 229, 299, 301, 302, 312]; Tabu Search [23, 114, 192, 229, 258, 259, 296, 310, 321]; variable neighborhood search [55, 107, 164, 173].
VRPTW	Exact methods	Branch-and-bound [153]; branch-and-cut [24]; branch-and-price [69, 70, 154, 252, 271, 300]; branch-and-cut-and-price [11, 32, 78, 79, 143, 235]; constraint programming [8, 270]; dynamic programming [63, 140, 155, 156]; column-and-cut generation algorithm [19, 20].
	Approximate methods	Ant colony optimization [22, 83, 85, 103, 112, 295, 308, 324]; column generation [2, 56, 80, 90, 188, 270, 300]; genetic algorithm [7, 31, 57, 117, 136, 251, 294, 305, 306]; GRASP [41, 52, 159, 187, 189, 257]; Lagrangian relaxation-based heuristic [105, 123, 148, 150, 153]; large neighborhood search [252, 267]; particle swarm optimization [3, 26, 322, 330, 332]; simulated annealing [21, 58, 68, 73, 177, 193, 303, 305, 308]; Tabu Search [14, 39, 65, 93, 187, 251, 275, 292, 305]; variable neighborhood search [40, 41, 244].
VRPPD	Exact methods	Branch-and-bound [196]; branch-and-cut [66, 97, 147, 268, 289]; branch-and-price [75, 124]; branch-and-cut-and-price [18, 266, 290]; dynamic programming [96].
	Approximate methods	Ant colony system [51, 110, 111, 327, 328]; Benders decomposition [277]; column generation[10, 81, 127, 280]; constructive heuristic [33, 82, 190, 262, 263, 288]; GRASP [128, 255]; genetic algorithm [134, 145, 231, 316, 329]; large neighborhood search [28, 206, 255, 267]; local search heuristic [77, 122, 283, 326]; particle swarm optimization [4, 119]; simulated annealing [28, 76, 134, 149, 315]; Tabu search [46, 54, 122, 165, 218, 223]; variable-depth search [313]; variable neighborhood search [50, 213].

vessels, while scheduling is routing with time (or time windows) attached to the calls of the vessels in the ports [265].

There are three types of operational modes of sea-going vessels, including liner shipping, industrial shipping, and tramp shipping [265]. Therefore, SRSPs with different operational modes are also different. In liner shipping, the vessels have fixed routes according to a published schedule, with the aim to maximize profits. The SRSPs for liner shipping include network design on strategic level, fleet deployment on the tactical level [62]: network design consists of constructing routes and choosing which routes to serve, and a route concerns which ports to visit and in which sequence, how often to visit these ports, and the size and ship that should be used; fleet deployment refers to the assignment of ships to liner routes, and the planning horizon ranges from a shipping season to 6 months.

In industrial shipping, an industrial operator owns the cargo and controls the ships, with the aim to minimize the cost of delivering the cargoes [60]. For tramp shipping, the vessels follow the available cargoes (some of which may be optional), trying to maximize profit [60]. For industrial and tramp shipping, the SRSPs include fleet size composition on strategic level, cargo routing and scheduling on tactical level [62]: fleet size and composition concerns how to manage a fleet over time, and decides how many ships to buy and sell and the timing of these activities; the cargo routing and scheduling problem concerns the routing of a fleet of ships to serve a number of specified cargoes that are given as input to the planning process.

Other common SRSPs include the maritime inventory routing problem (MIR), sailing speed, bunkering and refueling operations, emission control, and offshore logistics [62]. MIR is a combination of inventory management and routing and scheduling of ships. The problem with respect to ship speed concerns taking into account sailing speed as decision variables when planning vessel routes, with the aim to maximize profits or minimize costs. Speed optimization concerns adjusting the vessel's speed to arrive at the berths within the allocated berthing time. Bunkering and refueling, emission control are also considered in designing routes or deciding ship speed in literature. Offshore logistics concerns the routing and scheduling of offshore supply vessels carrying products between onshore depots and offshore oil and gas installations.

The VRPP shares similarities with two sub-categories of ship routing and scheduling problem, including the cargo routing and scheduling problem and the fleet deployment. Firstly, the process of picking up and delivering containers from different ports is similar to the process of picking up and delivering containers from different terminals. Moreover, for both problems there are time windows in which the loading and unloading of cargo must start. From this perspective, some of the relevant time-window constraints might be applicable from SRSP to VRPP.

As the publications on SRSPs are largely distributed on different topics, it is difficult to summarize them using one table, it is referred to [60–62, 265] for explicit reviews. This chapter only summarizes the solution methods that have been applied in cargo routing and scheduling problem and the fleet deployment problem as follow:

- Exact methods: branch-and-price-and-cut algorithm [137]; commercial solvers [113, 195, 210].
- Approximate methods: multi-start heuristic [42, 101]; unified tabu search heuristic [161, 162]; variable neighborhood search [201], genetic algorithm [191]; column

Table 2.2: Comparison with traditional planning problems.

	VRPP	**VRP-variants**	**SRSP**
1. Involved elements	Inland vessels; terminals	Customers; depots; vehicles	Sea-vessels; terminals
2. Origins/destinations	No	Yes	Yes
3. Capacity of (un)loading locations	Yes	No	No
4. (Un)loading time windows	Already scheduled time window could be missed	Deadlines for scheduled time window must be met	No
5. Waiting time at (un)loading locations caused by other vehicles	Yes	No	No
6. Flexibility in changing visiting sequences on routes	Yes	Yes	No
7. Operator cooperativeness	Unwilling to cooperate, with different sometimes conflicting interests	Fully-cooperative	Fully-cooperative

generation [43, 152, 176]; large neighborhood search [141].

Most of the relevant publications on these two problems focus on developing new mathematical formulations, the publications on proposing new algorithms for solving these problems are relatively fewer than the publications on VRP-variants.

2.3.3 Comparison with traditional planning problems

While the VRPP have several similarities with the VRP-variants and SRSP, it is also different from these problems. Table 2.2 summarizes the differences between VRPP, VRP-variants and SRSP.

Firstly, the vehicles in VRP-variants and SRSP start from the depots to different customer locations, which means that the origins and destinations need to be explicitly considered. Meanwhile, the depots of inland vessels are usually located in the hinterland, for VRPP which is within the port area, there is no origin for each vessel, and the terminals are their destinations. Consequently, the origins and destinations are not all considered in VRPP. This leads difference 1 and 2 in the table.

The most important difference between VRPP and other traditional planning problems lies in the way the waiting time of a inland vessel at a (un)loading locations (terminal) is considered, and this major difference leads to difference 3, 4 and 5 in Table 2.2. The waiting time of a inland vessel at a terminal is due to several reasons.

In the first place, in VRPP, the fact that each container terminal can only serve a limited number of vessels causes the vessels with later arrival times to wait or go to other terminals. This leads to difference 3, as the VRP-variants do not consider the capacity of (un)loading locations for handling vehicles. Although the capacity constraint of terminals also exists in SRSP, it is unlikely to cause the waiting of sea-going vessels. This is because sea-going vessels have made appointments with terminals for (un)loading operations a long time in advance and that they always have priorities over inland vessels at terminals. Therefore, it is common that in literature the capacity constraint of terminals are not considered in SRSP.

In the second place, in VRPP, if a vessel operator decides to wait until being handled, the loading and unloading operations of the vessel only starts after the vessels that arrived earlier at the terminal have been handled. This implies that inland vessels are allowed to

missed the scheduled time window at terminals, as long as they are willing to wait. This leads to difference 4, as in VRP-variants and SRSP, the vehicles must start within a given time window, otherwise the generated route is no longer feasible.

In the third place, in VRPP, the length of the waiting time that an inland vessel spends at a terminal not only depends on how many other vessels are being handled at the terminal, but also depends on the sequence in which this vessel visits the preceding terminals. Moreover, if a vessel arrives earlier at a terminal, it will also be served earlier. Consequently, the arrival time of a vessel at a terminal also affects the its waiting time. As the sequence in which a vessel visits the preceding terminals determines its arrival time at the forthcoming terminal, this visiting sequence also affects the length of waiting time this vessel spends at the forthcoming terminal. Meanwhile, in VRP-variants, a vehicle usually waits at a (un)loading location because it arrives earlier than the given handling time window, which is not caused by the visiting sequences or arrival times of any other vehicle. In SRSP, the waiting time caused by other vessels is not common and therefore is often not considered in literature. This leads to difference 5.

In addition, the inland vessels in VRPP and the vehicles in VRP-variants both have flexibility in changing their visiting sequence on routes. The vehicles in VRP-variants can even change routes under certain circumstances. However, this type of flexibility is not possible for SRSP, due to the dispersed geographical region of multiple ports. This leads to difference 6.

Last but not least, the vehicles in VRP-variants, as well as the sea-going vessels in SRSP are actually cooperating with each other to minimize the total costs, as they are all planned and scheduled within one liner organization. Meanwhile, the vessels in the VRPP are owned by different parties, and they are usually in competitive relations and therefore are not willing to cooperate with each other. This leads to difference 7.

These differences make it difficult to directly apply the existing methods of the traditional planning problems to the VRPP, and it is therefore both challenging and crucial to investigate how the existing methods from these problems could be applied to VRPP. The following section discusses the applicability of existing solution methods to the VRPP.

2.4 Applicability of existing solution methods

It can be concluded from Section 2.3.1 and Section 2.3.2 that there are mainly three types of solution methods that are applicable for the coordination problem considered in this thesis, including mathematical programming, constraint programming, and meta-heuristics.

Mathematical programming (MP) is the study of optimization problems with the aim to minimize or maximize a real function of real or integer variables, subject to constraints of the variables [207]. In broad terms, mathematical programming is a mathematical representation aimed at programming or planning the best possible allocation of scarce resources [38]. Mathematical programming models can be categorized into linear programming (LP) models, non-linear programming (NLP) models, and integer programming (IP) models [319].

Constraint programming (CP) is a useful framework for solving combinatorial search problems, which replies on a number of techniques from artificial intelligence, computer science, and operations research [269]. It concerns a formulation of the problem as a con-

straint satisfaction problem and solving it by means of domain specific or general methods [9]. A constraint can be viewed as a requirement that states which combination of values from the variable domains are allowed [9]. More formally, a constraint satisfaction problem consists of a set of variables, each with different value domains, and a set of relations on subsets of these variables[269]. Constraint programming covers a large spectrum of research, including algorithms, artificial intelligence, combinatorial optimization, computer systems, computational logic, operations research, and programming languages [269].

Meta-heuristics are a class of approximate methods that are designed to solve hard combinatorial optimization problems where classical heuristics have failed to be effective and efficient [230]. This chapter uses the definition of meta-heuristics from [230]. Meta-heuristic is an iterative generation process, in which different learning strategies are combined intelligently to guide a specific heuristic to explore and exploit the search spaces of the formulated problem, in order to find near-optimal solutions in an efficient way. Meta-heuristic is closely related with mathematical programming, as the optimization techniques from mathematical programming can be integrated into meta-heuristics and vice versa. Therefore, similar to mathematical programming, meta-heuristics have also been commonly used for solving operational planning problems. Meta-heuristics that have been applied to VRP-variants and SRSP include simulated annealing, Tabu search, variable neighborhood search, large neighborhood search, genetic algorithm, Memetic algorithm, ant colony optimization, multi-start heuristic, greedy adaptive, guided local search and iterated local search.

As can be seen from the literature review in Table 2.1, most planning problems are formulated based on mathematical programming techniques, and numerous meta-heuristics have been applied. Meanwhile, less attention has been paid to constraint programming techniques.

The main difference between MP and CP exists in their perspective on constraints. For CP, a constraint is regarded as a procedure that operates on the solution space, normally by reducing variable domains [269]. For MP, a constraint is integrated with other elements in the model, and the solution algorithm operates on the whole problem instead of individual constraints. This difference also leads to the differences with respect to the optimization techniques of these two methods. MP uses relaxation techniques to replace the whole problem with a simpler one, and enlarges the search space in a way that makes it easier to examine. CP uses inference methods to find out implicit information over the search space, so that less search is necessary. In other words, CP can deduce from a individual constraint that certain variables will never take certain values in an optimal solution. Even though inference methods also exist in MP, the forms are different from CP. Inference methods in MP usually refers to cutting planes and pre-processing techniques, while inference methods in CP are in the form of domain filtering and bounds propagation [132].

This thesis mainly uses CP techniques instead of MP techniques for several reasons. Firstly, a MP model typically consists of several linear or non-linear equalities, while the VRPP involves many equalities and logical conditions between inland vessels. For example, the waiting time of a inland vessel at a terminal not only depends on the time that the other vessels that are currently being handled at the same terminal spend, but also depends on the sequences how the other vessels visit the previous terminals. This implies that the calculation of the waiting time of a inland vessel at a terminal involves a sequence of variables representing how this vessel visit the previous terminals, as well as a sequence of variables how the other vessels visit the previous terminals. Therefore, it is more straightforward to

express such relations in logical constraints instead of inequalities.

Secondly, CP has representational advantages that can make problems easier to model. There are a number of global constraints that can be applied to the model as a whole and incorporated in the CP system. For example, the global constraint $allDifferent(x_1, x_2, ..., x_n)$ indicates that the variables in the set $(x_1, x_2, ..., x_n)$ must take different values. This constraint can be used to prevent the inland vessel from visiting a terminal for more than once. In addition, the global constraint

$$cumulative((s_1, \ldots, s_n)|(p_1, \ldots, p_n), (c_1, \ldots, c_n), C),$$

where, variables $(s_1, \ldots, s_n)$ represent the start time of certain generic jobs, parameters $(p_1, \ldots, p_n)$ contain certain processing time p_j of job j, and $(c_1, \ldots, c_n)$ contains the so-called consumption rate c_j for each job j. The constraint requires that the total rate of resource consumption of jobs running at any time t never exceeds C. In VRPP, this constraint can be used to represent the capacity constraint of a terminal for handling inland vessels. Variables $(s_1, \ldots, s_n)$ can be used to represent time at which the vessels that will be at terminal i. Processing time p can be considered as the service time for a vessel at a terminal. Consumption rate c can be set to 1 because one vessel can only be handled by one terminal. Parameter C can be the number of vessels one terminal serves simultaneously. Although this constraint can be expressed by conventional MP formulations by introducing auxiliary variables, it is cumbersome and would substantially increase the number of constraints in the model and therefore make the MP model complicated.

Thirdly, although this thesis mainly focus on CP techniques, there are several integrated methods that can incorporate the strength of MP into CP. For example, the relaxation technique from MP can help reduce domains and guide the search in CP. In addition, the recursive structure of MP techniques such as dynamic programming and bucket elimination can also be applied in CP techniques. Moreover, CP-based branch-and-price methods and generalized Benders decomposition can decompose the problems and apply CP and MP techniques to different parts of the problems.

Fourthly, there are extensions based on CP such as distributed constraint solving which can formulate the problem more closely to practice [269]. Last but not least, CP has been rarely applied to the operational planning problems in seaports, as well as the vessel rotation planning problem. Therefore, the potential of CP has not been fully investigated and research on application of CP techniques for solving VRPP would a novel contribution to literature.

In this thesis, the rotation planning problem of inland vessels is formulated based on CP. CP-based solution methods, as well as combined methods that integrate MP techniques and meta-heuristics with CP are proposed to solve the problem. Further details with respect to the proposed solution methods will be given in Chapters 3, 4 and 5.

2.5 Key performance indicators

To answer the research questions of this thesis, it is important to define key performance indicators (KPI) to evaluate the algorithmic and logistical performance of the proposed methods. To conclude, the KPIs used in this thesis are listed as follow:

- KPI 1: Computation time (seconds), which refers to the time spend on solving a particular optimization problem with a specific solution algorithm;
- KPI 2: Quality of solutions, which is a ratio that equals to the objective value of a solution generated by a specific solution algorithm, divided by the objective value of a solution generated by another solution algorithm that is used for comparison.
- KPI 3: Total number of messages (by type), which equals to the sum of the number of messages that have been exchanged between agents in a specific solution algorithm;
- KPI 4: Number of messages sent/received per agent (by type), which equals to the number of messages each agent receives from and sends to the other agents in a specific solution algorithm;
- KPI 5: Total amount of information (by bytes), which equals to the accumulated size of messages that have been exchanged between agents in a specific solution algorithm;
- KPI 6: Amount of information sent/received per agent (by bytes), which equals to the accumulated size of messages each agent receives from and sends to the other agents in a specific solution algorithm;
- KPI 7: Total round-trip time (hours), which equals to the simulated value of the sum of the times that all vessels make a total round and get back in the port;
- KPI 8: Round-trip time per vessel (hours), which refers to the simulated value of the time an inland vessel makes a total round and get back in the port;
- KPI 9: Total waiting time (hours), which equals to the simulated value of the sum of the waiting times of all vessels in the port;
- KPI 10: Waiting time per vessel (hours), which equals to the simulated value of the waiting time of a vessel in the port;
- KPI 11: Port departure time (hours), which refers to the simulated value of the time when the last vessel from a set of upcoming vessels leaves the port.

Firstly, both exact and approximate solution methods are proposed in thesis. For the exact solution methods (complete algorithms) that are proposed in Chapter 3, which are guaranteed to be able to find optimal solutions, computation time (KPI 1) is an important algorithmic KPI. For the approximate methods (incomplete algorithms) that are proposed in Chapters 3-5, the quality of solutions (KPI 2) is also important algorithmic KPI, as it reflects the deviation of the obtained solutions from the optimal solutions.

Secondly, there are also several KPIs that are commonly used in evaluating algorithmic performance of specific optimization techniques. For example, the number of messages and amount of information sent/received are two important types of KPIs for distributed constraint optimization (DCOP) algorithms [167, 236]. This is because the DCOP algorithms are operated based on the communication and information exchange among agents. Therefore, KPIs 3-6 are important KPIs to measure the algorithmic performance of the DCOP-based coordination methods that are proposed in Chapter 3.

Table 2.3: Overview of benchmark layouts.

	Port size	Chapter	KPIs	No. of upcoming vessels (per hour)	No. of terminal visits per vessel	Terminal capacity	ITT containers	Terminal closing time	Sea-vessels
Benchmark Layout 1	Small	Chapter 3	1,2,3,4,5,6	3-6	2	No	No	No	No
Benchmark Layouts 2.1, 2.2 and 2.3	Medium	Chapter 4	2,7,9,11	6-8	1-3	Optional	Yes	Yes	Yes
Benchmark Layout 3	Large	Chapter 5	2,7,8,9,10,11	8-16	1-4	Mandatory	Yes	Yes	Yes

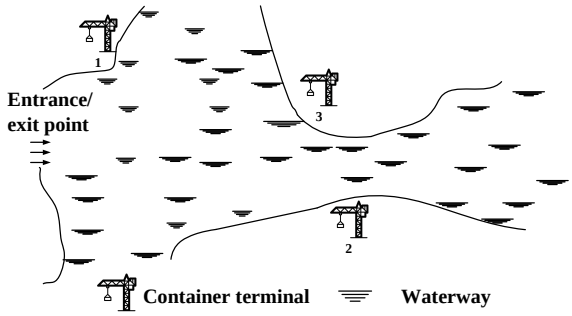

Figure 2.3: Simplified map of Benchmark Layout 1.

Besides the algorithmic KPIs 1-6, logistical KPIs are also required. The aim of this thesis is to improve the reliability and efficiency of inland vessel operations in the port. Meanwhile, as indicated earlier in Chapter 1, the waiting times and round-trip times of inland vessels reflect the reliability and efficiency of their operations in the port. Thus, this chapter defines logistical KPIs 7-11 that directly reflect those time-related logistical performance of inland vessel operations.

To conclude, KPIs 1-6 are algorithmic performance indicators to evaluate the performance of the proposed solution algorithms, and KPIs 7-11 are logistical performance indicators to evaluate the reliability and efficiency of inland vessel transport.

2.6 Benchmark systems

In order to evaluate the proposed methods in this thesis, five benchmark layouts are defined, which reflect vessel rotation planning problem in small, medium, and large sizes, respectively. Table 2.3 presents an overview of these benchmark systems.

According to [248], the number of vessels that enter the port of Rotterdam vary from 6 to 20 per hour, and 23.65 % of these incoming vessels are inland vessels. This means the estimation that 2 to 5 vessels enters the port per hour is valid.

Table 2.4: Different types of terminals and port layouts.

Terminal type	No. of Quays	No. of terminals in Layout 2.1	No. of terminals in Layout 2.2 and 2.3		
		Region1	Region1	Region2	Region3
α	1	4	4	3	3
β	2	3	1	1	2
γ	3	2	2	1	2

2.6.1 Benchmark Layout 1

Benchmark Layout 1 is used to evaluate the effectiveness and efficiency of partially-cooperative planning with single-level interactions in small ports in Chapter 3. Consequently, a small port with 3 terminals is presented, as shown in Figure 2.3, in which four different scenarios with different number of vessels are considered:

- Scenario 1: 3 vessels, each vessel needs to visit 3 terminals;
- Scenario 2: 4 vessels, each vessel needs to visit 3 terminals;
- Scenario 3: 5 vessels, each vessel needs to visit 3 terminals;
- Scenario 4: 6 vessels, each vessel needs to visit 3 terminals.

All vessels will visit all three terminals in the port for once, and for each terminal there are two quays available for handling inland vessels. For each scenario, 10 cases are considered with different parameters for utility values and traveling times. Further details of setting up the parameters will be given in Section 3.5.1.

2.6.2 Benchmark Layouts 2.1, 2.2 and 2.3

Benchmark Layout 2.1, 2.2 and 2.3 are used to evaluate the effectiveness and efficiency of partially-cooperative planning with multi-level interactions in medium ports in Chapter 4. These ports are larger than the port in Benchmark Layout 1, the particular layouts of which are shown in Figures 2.4, 2.5 and 2.6. The layouts are based on fictitious port data, which are similar to the settings used in [89]. These layouts are inspired by the geographical structure of large ports around the world (Singapore (Layout 2.1), Rotterdam (Layout 2.2), Antwerp (Layout 2.3), Shanghai (Layout 2.2) and Hamburg (Layout 2.3)). Although these layouts do not fit these ports exactly, they are reasonable approximations. The number of terminals varies in each region, as well as the number of quays in each terminal in our settings, as shown in Table 2.4.

Based on these settings, for each port layout, 10 different cases are defined in which 16 vessels arrive at the port within a 4-hour time range, in which 9 vessels visit 8 terminals, while 6 of them visit 6 terminals. In each case, the required number of containers the vessels need to transport and the ID of the terminals the vessels need to visit is varied.

In practice, the average call size of the container terminals in Rotterdam is 33 containers, but fluctuates largely between 15 and 52 containers [226]. Consequently, the same assumption is made in setting the required number of containers that need to be loaded and unloaded at each terminal in each case. In addition, some of the terminals are open 12h a day (6:00am - 6:00 pm). The sum of the number of the optional ITT containers that need

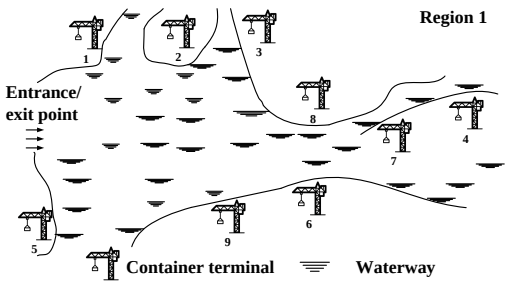

Figure 2.4: Simplified map of Benchmark Layout 2.1.

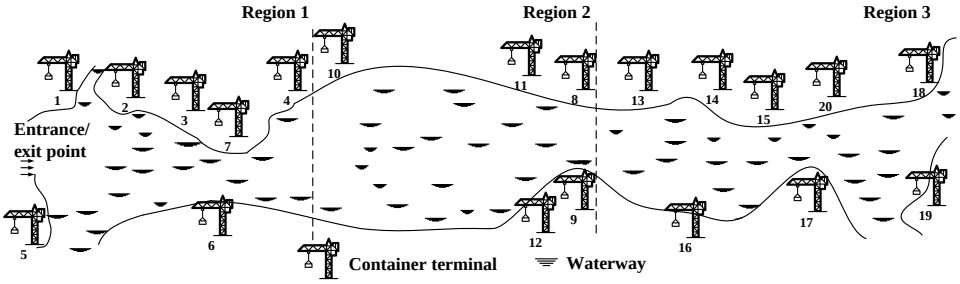

Figure 2.5: Simplified map of Benchmark Layout 2.2.

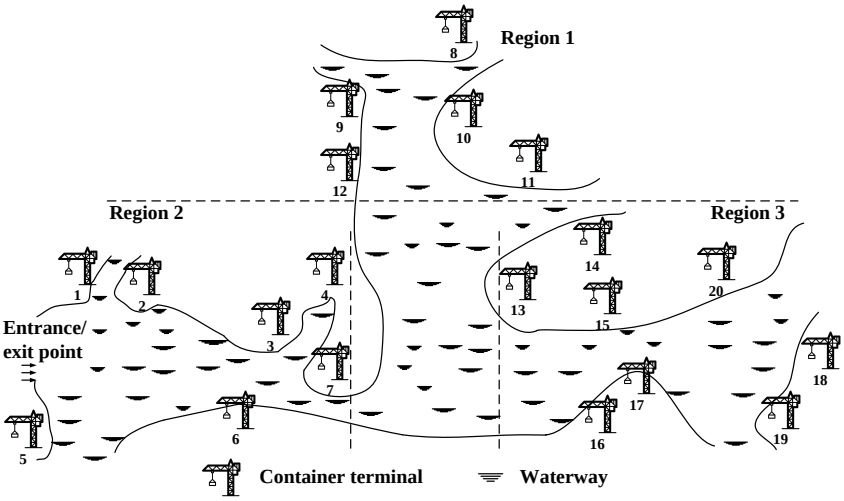

Figure 2.6: Simplified map of Benchmark Layout 2.3.

to be transported ranges from 50 TEUs to 150 TEUs, and the capacity of the inland vessels ranges from 150 TEUs to 250 TEUs. The distances between terminals are represented with different traveling times.

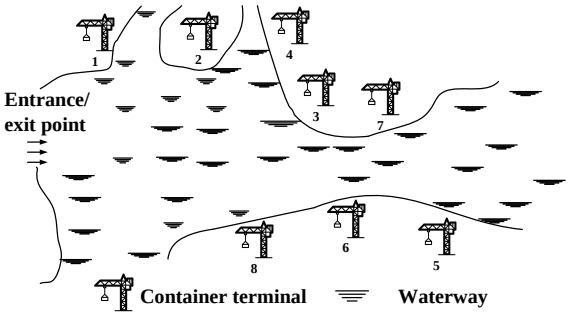

Figure 2.7: Simplified map of Benchmark Layout 3.

2.6.3 Benchmark Layout 3

Benchmark Layout 3 is used to evaluate the effectiveness and efficiency of fully-cooperative planning with multi-level interactions in large ports in Chapter 5. In Benchmark Layout 3, even larger ports are considered, as shown in Figure 2.7. Two scenarios are considered:

- Scenario 1: 8 vessels, each vessel visits 8 terminals;
- Scenario 2: 16 vessels, each vessel visits 8 terminals.

Although this port layout seems to be geographically smaller than Benchmark Layouts 2.1, 2.2 and 2.3, the problem considered in Benchmark Layout 3 is actually larger. Firstly, the port is even more busy, in which it is assumed that 8 to 16 inland vessels enter the port per hour. Secondly, the terminals in the port have larger capacities for handling vessels (more quay cranes in the terminal). Moreover, all the vessels need to visit all 8 terminals in the port, which means the terminals are more crowded than in Benchmark Layouts 2.1, 2.2 and 2.3. In Benchmark Layout 2.1, on average 78% of vessels will visit the same terminal in the port. In Benchmark Layout 2.2 and 2.3, on average 35% of the vessels will visit the same terminal in the ports. Meanwhile, in Benchmark Layout 3, 100% of the vessels will visit the same terminal in the port. This also implies that the problem in Benchmark Layout 3 is more tightly constrained.

For the terminals in Benchmark Layout 3, the same types of terminal as in Table 2.4 are considered. For each scenario, 10 different cases are created based on different port layouts, in which the ID of the terminals the vessels need to visit and the number of hinterland and ITT containers is varied. Moreover, in each case, the number of α-terminal, β-terminal, and γ-terminal in the port is also varied. It is assumed that some of the terminals open 12 hours a day, and the distances between terminals are represented with different traveling times, just as in Benchmark Layouts 2.1, 2.2 and 2.3. To investigate the impact of extra ITT containers on the round-trip time of the considered vessels, the number of ITT containers that need to be transported ranges from 5% to 30% of the sum of the mandatory hinterland containers of all considered vessels, while the extra capacity of each vessel for ITT containers also varies from 5% of 30% of its full capacity. The full capacity of the inland vessels ranges from 150 TEUs to 250 TEUs.

In addition, Benchmark Layout 3 also considers the following four types of disturbances: (1) unavailable quay cranes; (2) delay of sea-going vessel; (3) delay of hinterland

containers; (4) terminal closing due to extreme weather.

2.7 Conclusions

This chapter firstly presents a overview on the different planning problems in large seaport, and a literature review on the operational planning problems that are relevant to the handling of inland vessels is given. In contrast with other operational planning problems, vessel rotation planning problem is not fully investigated in literature. This implies that more improvements could be made on VRPP to ensure efficient handling of inland vessels in the port. Therefore, this thesis proposes different coordination methods to plan rotations for inland vessels. Secondly, a literature review on other traditional planning problems is also given, and the applicability of existing solution methods of these problems to the problem considered in this thesis is analyzed and discussed. Comparing with other solution methods, constraint programming is most suitable for solving the VRPP. Therefore, this thesis proposes different solution methods that are based on constraint programming in Chapters 3, 4 and 5 for solving the VRPP with different conceptual frameworks. Thirdly, to answer Key Research Question 1, key performance indicators and benchmark layouts are defined to evaluate the proposed conceptual frameworks in Chapter 1. Chapter 3 uses Benchmark Layout 1 for partially-cooperative planning with single single-level interactions in small ports. Chapter 4 uses Benchmark Layouts 2.1, 2.2 and 2.3 for partially-cooperative planning with multi-level interactions in medium ports. Chapter 5 uses Benchmark Layout 3 for fully-cooperative planning with multi-level interactions in large ports. These benchmark systems will be assessed using the corresponding KPIs.

Chapter 3

Partially-cooperative planning for single-level interaction in small ports

According to the relations between vessel operators and terminal operators that are described in Chapter 2, it is natural to model their interactions in a distributed way, as they are independent parties and all have different preferences and interests. Moreover, inland vessel operators are in competitive positions and are unwilling to share their information with each other. Therefore, this chapter investigates distributed and exact methods for improving the coordination of inland vessels for small ports. Specifically, this chapter proposes two methods based on distributed constraint optimization (DCOP) to solve the problem, namely non-layered DCOP (referred as method M1.1) and layered DCOP (referred as method M1.2). In each method, four representative DCOP solution algorithms are investigated.

The research discussed in the chapter is based on [178, 179, 183].

3.1 Introduction

The distributed constraint optimization problem (DCOP) is a theoretical model framework in which several agents that jointly make decisions on values of variables so as to minimize the sum of constraint costs, or to maximize the sum of utility values [317]. A DCOP is defined as consisting of a set of agents, variables and constraints between variables that reflect the costs/utilities of assignments to variables. Control of values of variables in DCOPs is distributed, with agents only able to assign values to variables that they own. Furthermore, agents are assumed to know only the constraints involving variables that they own.

In order to find a solution to a DCOP, agents need to communicate with each other through message exchange. It is commonly assumed that agents can only communicate with agents that hold variables constrained with their own variables. The agents with which an agent can communicate are called its neighbors [215, 233]. The DCOP formalism has been mainly applied in meeting scheduling [121, 199, 237], coordination of sensors in networks [135, 174, 333], resource allocation in disaster evacuation [49, 166], synchronization of

traffic lights [146], and management of power distribution networks [163]. Distributed constraint optimization is well suited for formulating those problems since they are distributed by nature. Consequently, this chapter adopts the DCOP formalism to model vessel operators and terminal operators as individual agents. Although this problem can also be solved in a centralized way, which saves time to exchange messages among agents. However, this is often not preferable for the operators, as they would like to keep some constraints as private. Moreover, a centralized solution method makes the whole system less robust.

This chapter proposes two methods based on DCOP for improving the inland vessel coordination, namely the non-layered DCOP and layered DCOP. In each method, the performances of 4 state-of-art DCOP solution algorithms are evaluated in terms of quality of the solution, communication load and computation time.

This chapter is organized as follows. The background of DCOP is presented in Section 3.2. Section 3.3 introduces the mathematical description of the vessel rotation planning problem based on non-layered (method M1.1) and layered DCOP (method M1.2). Section 3.4 describes how the different solution algorithms work for the formulated DCOP formulation. Experimental results on the algorithmic and logistical performances of different DCOP algorithms are given in Section 3.5. Section 3.6 concludes this chapter.

3.2 DCOP Background

This chapter adopts the DCOP formalism as defined in [236]. A DCOP is represented by a triple $\langle \mathcal{A}, \mathcal{COP}, \mathcal{R}^{\text{ia}} \rangle$, where:

- $\mathcal{A} = \{A_1, \ldots, A_M\}$ is a set of M agents;
- $\mathcal{COP} = \{COP_1, \ldots, COP_M\}$ is a set of disjoint, local Constraint Optimization Problems (COPs); COP_m is called the local sub-problem of agent A_m; COP_i is defined by a triple $\langle \mathcal{X}_m, \mathcal{D}_m, \mathcal{R}_i \rangle$, where $\mathcal{X}_m = \{X_{m1}, \ldots, X_{m|\mathcal{X}_m|}\}$ is a set of $|\mathcal{X}_m|$ variables that belong to A_m; $\mathcal{D}_m = \{d_{m1}, \ldots, d_{m|\mathcal{X}_m|}\}$ is a set of finite variable domains of the variables in $\mathcal{X}_m$; $\mathcal{R}_i = \{r_{m1}, \ldots, r_{m|\mathcal{R}_m|}\}$ is a set of $|\mathcal{R}_m|$ utility functions, where each utility function $r_{m|\mathcal{R}_m|}$ is with scope $\mathcal{X}_m$, $r_{m|\mathcal{R}_m} : d_{m1} \times \cdots \times d_{m|\mathcal{X}_m|} \to \mathbb{R} \cup \{-\infty\}$. The utility functions are used to represent objectives, as well as both hard and soft constraints. For hard constraints, the value of the utility function is 0 if the constraint is satisfied; otherwise the value is $-\infty$. For soft constraints, for different combinations of the values for variables, different values will be assigned to the utility functions.
- $\mathcal{R}^{\text{ia}} = \{r_1^{\text{ia}}, \ldots, r_{|\mathcal{R}^{ia}|}^{\text{ia}}\}$ is a set of so-called inter-agent utility functions defined over variables of multiple agents. Each $r_l^{\text{ia}} : scope(r_l^{\text{ia}}) \to \mathbb{R}$ expresses the utility for a joint decision obtained by the agents that have variables involved in r_l^{ia}. The agents that have variables can decide on the values of these variables involved in r_l^{ia} and are called "responsible" for r_l^{ia}. Inter-agent utility functions are considered known to all agents involved, i.e, those agents of which the local variables are part of the inter-agent utility function.

The objective of the agents solving a DCOP is to find the assignment to all variables such that the sum of values of all utility functions (representing the objectives, hard and soft constraints) are maximized. So, the agents determine:

$$X^* = \arg\max \sum_{m=1}^{M} \left(\sum_{v=1}^{|\mathcal{R}_m|} r_{mv}(X_{m1}, \ldots, X_{i|\mathcal{X}_m|}) \right) + \sum_{l=1}^{|\mathcal{R}^{ia}|} r_l^{\text{ia}}$$

Since variables from different agents can be constrained via inter-agent utility functions, to make sure these constraints represented by the inter-agent utility functions are satisfied and to find the optimal solution X^*, agents need to communicate and exchange messages. Those messages include information on the assignments of values to variables and the associated utility values. The total number of messages and the size of messages sent by the agents is therefore considered an important performance metric for measuring the efficiency of different DCOP solution algorithms, besides the quality of the solutions obtained.

DCOP solution algorithms can be categorized as complete and incomplete algorithms. Complete algorithms are guaranteed to find optimal solutions, if they exists. Complete algorithms typically do an exhaustive search over the problem space. Complete algorithms include Asynchronous Distributed Optimization (ADOPT) [215], Dynamic Programming Optimization Protocol (DPOP) [236], Memory-Bounded Dynamic Programming Optimization Protocol (MB-DPOP) [236, 238], Asynchronous Forward Bounding (AFB) [115] and Synchronous Branch and Bound (SyncBB) [129]. Incomplete algorithms usually use local search methods to find locally optimal solutions. As a result, they can get trapped in local minima. As a DCOP is a NP-hard problem [168], incomplete algorithms are more practical since finding solutions can be intractable. Incomplete DCOP algorithms include Distributed Stochastic Algorithm (DSA) [106], Maximum Gain Message (MGM) [200], and Max-Sum [102]. In [236], a detailed description and comparison of the DCOP algorithms can be found. This chapter applies complete algorithms SyncBB, AFB, DPOP and MB-DPOP to solve the VRPP.

3.3 Vessel rotation planning problem as a DCOP

As solving DCOP problems is NP-hard [215], increasing the number of variables increases the complexity of the problem exponentially. Therefore, using straightforward DCOP methods for solving VRPPs that consist of many vessels and terminals brings high computation and communication costs. Partitioning of the DCOP into multiple layers could significantly improve the solution process.

This section first defines the problem by giving assumptions and defining variables and parameters. Then both the non-layered and layered DCOP methods for vessel rotation planning problem is presented with detailed mathematical formulations.

3.3.1 Variables and parameters

VRPP concerns selecting the rotation plan consisting of sequences of visits to terminals and arrival and departure times at these terminals for a set of vessels in a port area. To model the VRPP as a DCOP, the following assumptions are made:

- Decisions are made at discrete time steps, so discrete *time slots* can be considered.
- Each vessel knows which terminals it needs to visit. The visiting sequence, however, needs to be determined.

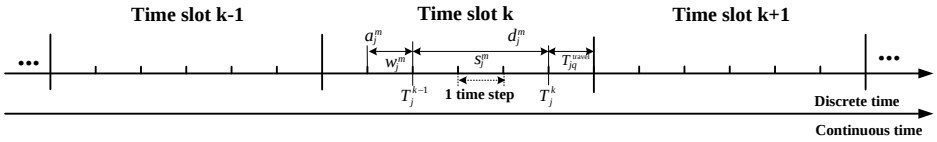

Figure 3.1: Time scales of the problem formulation.

Table 3.1: Decision variables used in the model.

Symbols	Definitions
x_i^m	the time slot at which vessel m is at terminal i
a_i^m	the arrival time steps of vessel m at terminal i
d_i^m	the departure steps of vessel m at terminal i
w_i^m	the number of time steps vessel m has waited at terminal i

Table 3.2: Parameters used in the model.

Symbols	Definitions
M	set of vessels entering the port
N_m	the set of terminal that vessel m needs to visit in the port
D	set of discrete time slots
D_s	set of discrete time steps
U_{ik}^m	preference of vessel m of being at terminal i during time slot k
W_{ik}^m	utility value of waiting time w_i^m during time slot k
s_i^m	service time of vessel m at terminal i
T_{ij}^{travel}	traveling time from terminal i to j
$[T_i^{k-1}, T_i^k]$	a fixed time window during which vessels can visit terminal i during time slot k

- Each vessel has its own preference regarding its optimal terminal visiting sequence.
- Distances between terminals are given.
- Service time of a vessel at a terminal is known.

Figure 3.1 illustrates the time scales that are considered. The continuous time is divided into discrete time slots, a number of time steps form a time slot. The problem can now be formalized using the variables shown in Table 3.1. Each time step is 1 hour. Integer variable x_i^m represents the time slot during which vessel m will be at terminal i, variables a_i^m, d_i^m represent, respectively, the specific time steps at each the vessel arrives at and leaves the terminal, w_i^m represents the number of time steps vessel m waited at terminal i.

In addition, the parameters are defined in Table 3.2. Parameter U_{ik}^m represents the preferences of vessel m at terminal i. For example, $U_{11}^1 = 5$, $U_{12}^1 = 4$, $U_{13}^1 = 3$ represents that vessel 1 prefers to visit terminal 1 during time slot 1, since U_{11}^1 has the highest preference value. These preferences are implemented in the DCOP framework by assigning different preference values when $x_i^m = k$, and assigning 0 when $x_i^m \neq k$.

Parameter W_{ik}^m represents the utility value a vessel m being at terminal i at time slot k. This is included in the DCOP framework via a constraint $w_i^m = k_s$, where $k_s \in D_s$ is the duration of time steps, representing the waiting time of vessel m at terminal i. A higher

value of k_s represents a longer waiting time of vessel m at terminal i. Since vessel operators usually prefers to have less waiting time at terminals, the higher the value of k_s is, the lower the value of W_{ik}^m will be.

Service time s_i^m is considered as an integer constant, e.g., $s_i^m = 3$ means that it takes 3 time steps for vessel m to be serviced at terminal i. The service times for different vessels at different terminals are different. Traveling time T_{mn}^{travel} is also an integer constant with different values per pair of terminals to represent the different distances between different terminals. Similarly to s_i^m, $T_{ij}^{\text{travel}} = 3$ represents that it takes 3 time steps for a vessel to travel from terminal i to j.

Time interval $[T_i^{k-1}, T_i^k]$ is a fixed time window during which vessels can visit terminal i during time slot k. For example, $[T_1^3, T_1^3] = [6, 10]$ represents that if vessel i visits terminal 1 during time slot 3 (when $x_1^m = 3$), the arrival time of vessel i at terminal 1, represented by a_{i1}, must be within the time window $[6, 10]$. The concept of time window is introduced to ensure that a vessel that visits terminal j during time slot k can be serviced only if it arrives at the terminal within a specified time window $[T_i^{k-1}, T_i^k]$. Parameter T_j^k is the deadline for a vessel's arrival to be serviced at terminal i if it visits the terminal during time slot k.

The objective is to find the optimal rotation plans for vessels, which include the sequences of visits' to terminals, the departure and arrival times of vessels at each terminal, such that a set of utility functions are maximized. Different utility values represent different preferences for visiting terminals at different time slots. Therefore, maximizing the sum of the utility functions means satisfying the preferences of all the agents as much as possible. As all the vessel agents considered are in equal positions, utility values U_{ik}^m and W_{ik}^m for different vessel agents are within the same range of values.

As described in Section 3.2, to model a problem as a DCOP, three main elements must be defined: agents, local constraint optimization problems of the agents, and the inter-agent utility functions. The VRPP can be cast as a DCOP by considering the vessels and terminals as agents, the constraints related with vessels and terminals as constraints of local problems of the corresponding agents. As the constraints for terminal agents usually involve variables from different vessel agents, these constraints can be considered as inter-agent utility functions. Based on this structure, a general definition of the VRPP as DCOP is generated, as a tuple of the following form: $\langle \mathcal{A}, \mathcal{COP}, \mathcal{R}^{\text{ia}} \rangle$, in which $\mathcal{A}$ is the set of vessel agents and terminal agents; $\mathcal{COP}$ is the set of local constraint optimization problems for each vessel/terminal agent; $\mathcal{R}^{\text{ia}}$ is the set of inter-agent utility functions defined over variables from different vessel/terminal agents. Vessel-related utility functions include hard constraints to make sure that for each vessel rotation a vessel visits each terminal only once, arrival and departure time windows required by terminals are respected, and soft constraints representing their preferences over the time slots at which to visit terminals are taken into account. Terminal-related utility functions include hard constraints to make sure the capacity of terminals is considered, i.e, one terminal can serve at most two vessels simultaneously. Inter-agent utility functions connect the variables in terminal agents with variables in vessel agents that have the same value domains.

In this model, agent A_m represents vessel m with its own variables x_i^m, a_i^m, d_i^m and w_i^m, agent B_i represents terminal i and owns variable y_i^m. In the local problem of A_m, values of variables x_i^m, a_i^m, d_i^m and w_i^m are determined by A_m, while the value of y_i^m is determined by B_j. Variable y_i^m is introduced as auxiliary variable that has the same value as x_i^m but is owned by B_i: it is used to represent the capacity constraints for the terminals. These variables

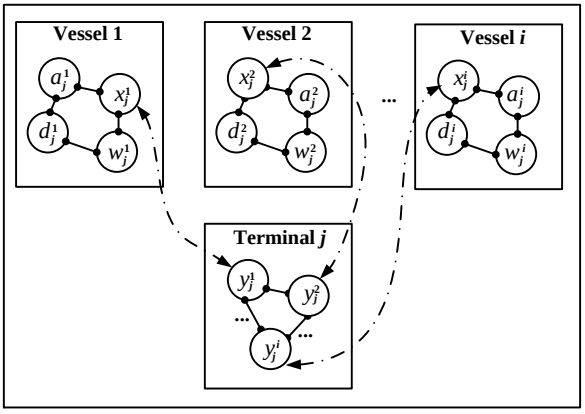

Figure 3.2: Non-layered DCOP framework.

are exclusively controlled by their owning agent, and the control of a utility function is shared among the agents who own a variable in their scopes. The inter-agent and local utility functions represent the hard constraints related to a vessel m and terminal j. These constraints are terminal capacity constraints at each terminal. The preferences of vessels of being at particular terminals during different time slots are also considered as the utilities of the local constraint optimization problem of the vessel agents. For our DCOP, the objective is to find the assignments of values from domains D and D_s to variables x_i^m, y_i^m, a_i^m, d_i^m and w_i^m, that maximizes the sum of the values of the utility functions of local COPs and the inter-agent utility functions.

3.3.2 M1.1: Non-layered DCOP framework

The non-layered DCOP framework is formulated, as shown in Figure 3.2. For vessel agent m,

- $\mathcal{X}_m = \{x_1^m, \ldots, x_{|\mathcal{X}_m|}^m, a_1^m, \ldots, a_{|\mathcal{X}_m|}^m, d_1^m, \ldots, d_{|\mathcal{X}_m|}^m, w_1^m, \ldots, w_{|\mathcal{X}_m|}^m\}$ is the set of decision variables;
- $\mathcal{D}_m = \{d_1^m, \ldots, d_{|\mathcal{D}_m|}^m\}$ is the set of finite variable domains;
- $\mathcal{R}_m = \{r_1^m, \ldots, r_{|\mathcal{R}_m|}^m\}$ is the set of utility functions that represent the constraints related to vessel m and vessel m's preference over visiting different terminals during different time slots. In particular,

$$r_{m1} = \begin{cases} U_{ik}^m : \text{if} x_i^m = k & \forall m \in M, \forall i \in N_m, k \in D \\ 0 : \quad \text{otherwise} \end{cases} \tag{3.1}$$

$$r_{m2} = \begin{cases} 0 : \quad \textit{all-different}(x_1^m, \ldots, x_{|N_m|}^m) & \forall m \in M \\ -\infty : \text{otherwise} \end{cases} \tag{3.2}$$

$$r_{m3} = \begin{cases} 0 : \quad \text{if} x_i^m = k, \text{then} a_i^m \in [T_i^{k-1}, T_i^k] & \forall m \in M, \forall i \in N_m \\ -\infty : \text{otherwise} \end{cases} \tag{3.3}$$

$$r_{m4} = \begin{cases} 0: & \text{if } d_i^m = a_i^m + w_i^m + s_i^m \\ -\infty: & \text{otherwise} \end{cases} \qquad \forall m \in M, \forall i \in N_m \tag{3.4}$$

$$r_{m5} = \begin{cases} 0: & \text{if } x_i^m = k, \text{then } w_i^m = T_i^k - a_i^m \\ -\infty: & \text{otherwise} \end{cases} \qquad \forall m \in M, \forall i \in N_m \tag{3.5}$$

$$r_{m6} = \begin{cases} W_{ik'}^m: & \text{if } w_i^m = k' \\ 0: & \text{otherwise,} \end{cases} \qquad \forall m \in M, \forall i \in N_m \tag{3.6}$$

$$r_{m7} = \begin{cases} 0: & \text{if } x_{i'}^m = x_i^m + 1, \text{then} \quad a_{i'}^m = d_i^m + T_{ii'}^{\text{travel}} \\ -\infty: & \text{otherwise.} \end{cases} \qquad \forall m \in M, \forall i, i' \in N_m \tag{3.7}$$

where utility function r_{m1} represents the preferences of vessel m for being at a particular terminal i during time slot k. Utility value U_{ik}^m is a constant defined for different combinations of m, i and k. Utility function r_{m2} is defined using the so-called *all-different* constraint [269] to ensure that each vessel will be at most at one terminal during a time slot. Utility function r_{m3} represents that if vessel m visits terminal i during time slot k, the arrival time a_i^m should be within the time window $[T_i^{k-1}, T_i^k]$ of time slot k. If vessel m cannot arrive within the time window, it will not be serviced during time slot k and has to wait to be serviced during the next time slot. Utility function r_{m4} ensures that the departure time d_i^m of vessel m from terminal i equals the sum of the vessel's arrival time, waiting time and service time at the terminal. Utility function r_{m5} represents that the waiting time is the difference between the arrival time of a vessel m at terminal i and the start time for the vessel to be serviced at the terminal. Utility function r_{m6} represents that higher preference is given to lower waiting time w_i^m. This utility function is used to represent that the objective of maximization of the sum of utility values will lead to the minimization of the sum of waiting times. Utility function r_{m7} represents that if vessel m travels from terminal i to i', the arrival time at terminal i' is the sum of the departure time from terminal i and the traveling time $T_{ii'}^{\text{travel}}$.

For each terminal agent, utility functions are also defined for representing the capacity constraints of terminals. Each terminal can serve only a limited number of vessels simultaneously. A special function *cumulative* is incorporated from constraint programming to represent utility functions $r^{\text{ia,as}}$ of terminal agents, which is defined as in [269] in the following generic way:

$$cumulative((s_1, \ldots, s_n)|(p_1, \ldots, p_n), (c_1, \ldots, c_n), C),$$

where, variables $(s_1, \ldots, s_n)$ represent the start time of certain generic jobs, parameters $(p_1, \ldots, p_n)$ contain certain processing time p_j of job j, and $(c_1, \ldots, c_n)$ contains the so-called consumption rate c_j for each job j. The constraint requires that the total rate of resource consumption of jobs running at any time t never exceeds C. In our case, $(y_i^m, \ldots, y_i^{|N_m|})$ are the auxiliary variables representing time slots at which the vessels that will be at terminal i. Processing time p is set as s_i^m since the service time for a vessel at a terminal is s_i^m. Consumption rate c is set to 1 because a vessel will be serviced by one terminal. Parameter C represents the number of vessels one terminal can serve simultaneously, which is C_i. Thus, this constraint can be formulated as follows:

$$r_{i0} = \begin{cases} 0: & cumulative((y_i^m, \ldots, y_i^{|N_m|}, s_i^m, 1, C_i) \\ -\infty: & \text{otherwise.} \end{cases} \qquad \forall i \in N_m \tag{3.8}$$

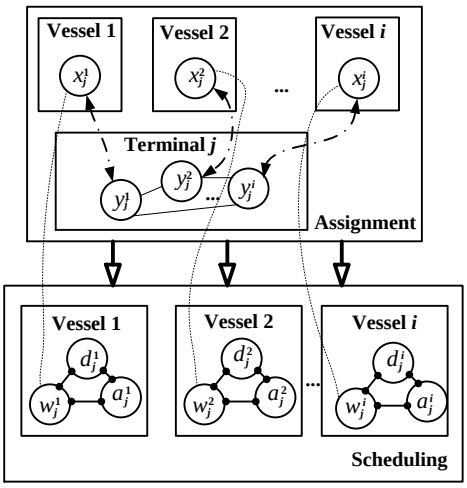

Figure 3.3: Layered DCOP framework.

In order to connect variable x_i^m of a vessel agent with duplicated auxiliary variables y_i^m of a terminal agent, the inter-agent utility function r_1^{ia} is defined as follows:

$$r_1^{\text{ia}} = \begin{cases} 0: & \text{if} \quad x_i^m = y_i^m \qquad \forall m \in M, \forall i \in N_m \\ -\infty: & \text{otherwise.} \end{cases} \tag{3.9}$$

For each agent, the objective is to maximize the sum of values of its utility functions. For the overall DCOP assignment problem, the objective is to maximize the sum of values of the individual utility functions and the inter-agent utility functions, defined as:

$$\max \left(\sum_{m=1}^{|M|} (r_{m1} + r_{m2} + r_{m3} + r_{m4} + r_{m5} + r_{m6} + r_{m7}) + \sum_{i=1}^{|N_m|} r_{i0} + r_1^{\text{ia}} \right). \tag{3.10}$$

3.3.3 M1.2: Layered DCOP framework

To formulate the layered DCOP, the VRPP is divided into two layers: an assignment layer and a scheduling layer, as shown in Figure 3.3. The upper layer defined as an assignment problem, which decides on the sequence in which the vessels visit different terminals. When the solutions have been obtained at the higher layer, the lower layer, which is defined as a scheduling problem, will decide the exact arrival, departure and waiting time of each vessel at each terminal. There is one DCOP in the upper layer that relates to all the vessels, while there are multiple smaller COPs in the lower layer, each relates only to 1 vessel. After finding an optimal solution for the upper layer DCOP problem, the lower layers DCOP solver starts solving each problem separately. There are common variables between the upper layer DCOP and the lower layer DCOPs.

Figure 3.4 shows the structure of the layered DCOP method for vessel rotation planning problem. The upper layer decides sequences for visiting different terminals and the lower layer decides the exact time of arrival, departure and waiting at each terminal. Problem formulations of each layer are introduced in the next.

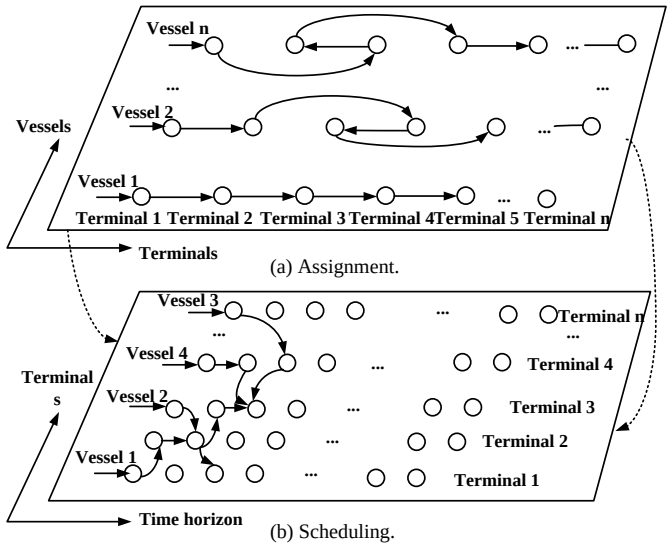

Figure 3.4: Layered DCOP method for VRPP.

Assignment Problem The upper layer, defined as an assignment problem, determines only the sequence in which vessels visit terminals is considered. Thus, only variable x_i^m, representing the time slot at which vessel m is at terminal i, as well as the auxiliary variable y_i^m, are involved. The exact arrival, departure and waiting time of vessels will not be included in this layer.

Terminal agents have the same structure as in the non-layered DCOP framework, and they only appears in the upper layer. The local problem COP_m of vessel agent A_m^{as} is defined by the triple $\langle \mathcal{X}_m^{\text{as}}, \mathcal{D}_m^{\text{as}}, \mathcal{R}_m^{\text{as}} \rangle$, where

- $\mathcal{X}_m^{\text{as}} = \{x_1^m, \ldots, x_{|\mathcal{X}_i^{\text{as}}|}^m\}$ is the set of variables, in which x_i^m represents the time slot at which vessel m is at terminal i;
- $\mathcal{D}_m^{\text{as}} = \{d_1^m, \ldots, d_{|\mathcal{D}_m^{\text{as}}|}^m\}$ is the set of finite variable domains, for each $d_i^m \in D_s$;
- $\mathcal{R}_m^{\text{as}} = \{r_{m1}^{\text{as}}, \ldots, r_{m|\mathcal{R}_m^{\text{as}}|}^{\text{as}}\}$ contains the utility functions that represent the constraints related with the vessel and its preferences regarding times at which to visit the different terminals, as well as the capacity constraints of terminals. Thus, the utility functions r_{m1}^{as}, r_{m2}^{as} and r_{i0}^{as} are the same as r_{m1}, r_{m2} and r_{i0} in non-layered DCOP, respectively. In addition, to connect the auxiliary variable y_i^m with variable x_i^m, inter-agent utility function $r^{\text{ia,as}}$ is also introduced as in r_1^{ia}.

For the overall upper layer DCOP assignment problem, the objective is to maximize the sum of the values of the individual utility functions and the inter-agent utility functions:

$$\max \left(\sum_{m=1}^{|M|} (r_{m1}^{\text{as}} + r_{m2}^{\text{as}}) + \sum_{i=1}^{|N_m|} r_{i0}^{\text{as}} + r^{\text{ia,as}} \right). \tag{3.11}$$

Scheduling Problem Given the solutions x_i^m from the upper layer assignment problem representing the sequence of visits of vessels to given terminals, the specific time slots during which each vessel will be at each terminal can be determined. The scheduling problem is formulated similarly as the assignment problem as a tuple $\langle \mathcal{A}^{\text{sc}}, \mathcal{COP}^{\text{sc}}, \mathcal{R}^{\text{ia,sc}} \rangle$. In this layer, only vessel agents are included. The local problem COP_m^{sc} of each vessel agent A_m^{sc} is defined by a triple $\langle \mathcal{X}_m^{\text{sc}}, \mathcal{D}_m^{\text{sc}}, \mathcal{R}_m^{\text{sc}} \rangle$, where

- $\mathcal{X}_m^{\text{sc}} = \{a_1^m, \ldots, a_{|\mathcal{X}_m^{\text{sc}}|}^m, d_1^m, \ldots, d_{|\mathcal{X}_m^{\text{sc}}|}^m, w_1^m, \ldots, w_{|\mathcal{X}_m^{\text{sc}}|}^m\}$ is a set of variables, in which a_i^m, d_i^m represent the arrival, departure time of vessel m from terminal i, w_i^m represents the waiting time of vessel m at terminal i;
- $\mathcal{D}_m^{\text{sc}} = \{k_{m1}, \ldots, k_{m|\mathcal{D}_m^{\text{sc}}|}\}$is a set of finite variable domains, for each $k_{mi} \in D_s$;
- $\mathcal{R}_m^{\text{sc}} = \{r_{m1}^{\text{sc}}, \ldots, r_{m|\mathcal{R}_m^{\text{sc}}|}^{\text{sc}}\}$ contains utility functions that represent the constraints related with vessels' arrival, departure and waiting at terminals. Here, r_{m1}^{sc}, r_{m2}^{sc}, r_{m3}^{sc}, r_{m4}^{sc}, and r_{m5}^{sc} are the same as r_{m3}, r_{m4}, r_{m5}, r_{m6}, r_{m7} in the non-layered DCOP.

Similar to the assignment problem, the objective of this DCOP problem is to maximize the sum of the values of the individual utility functions, defined as:

$$\max \left(\sum_{m=1}^{|M|} (r_{m1}^{\text{sc}} + r_{m2}^{\text{sc}} + r_{m3}^{\text{sc}} + r_{m4}^{\text{sc}} + r_{m5}^{\text{sc}}) \right). \tag{3.12}$$

3.4 DCOP solution algorithms

Once a problem has been modeled as a DCOP, a solution method is required to solve the DCOP. Since we are interested in determining the optimal solutions, this chapter focuses on complete algorithms. In this section, four complete DCOP algorithms are introduced, including Synchronous Branch and Bound (SyncBB), Asychrounous Forward Bounding (AFB), Dynamic Programming Optimization Protocol (DPOP) and Memory-bounded Dynamic Programming Optimization Protocol (MB-DPOP) to describe how the agents exchange messages and determine solutions. SyncBB and AFB are based on search (synchronous and asynchronous, respectively), while DPOP is based on dynamic programming, and MB-DPOP is a variant of DPOP that operates with bounded memory, therefore they are representative of the spectrum of the complete DCOP algorithms.

3.4.1 SyncBB

The simplest and first complete DCOP algorithm is SyncBB [129], which is a straightforward distributed adaptation of the centralized branch-and-bound mechanism. In branch-and-bound, the search for the optimal solution aimed at guiding by a global accumulated cost named bound. SyncBB is a distributed version of the branch-and-bound and aims to guide the search through a heuristic applied over the optimization function. Algorithm 1 shows the solution process of SyncBB. The variable x_i^m, owned by A_m^{as} in the upper layer of layered DCOP is used as an example. Firstly, all variables and agents are arranged along a total order with the priority $x_i^1 \succ \cdots \succ x_i^n$. The message passing starts with the highest priority variable x_i^1, the corresponding agent of which sends a so-called single Current Partial

Algorithm 1 SyncBB: Synchronous Branch and Bound.

Require: a fixed, known, linear ordering of variables $x_i^1 \succ x_i^m \succ \cdots \succ x_i^n$

//Join all utility functions involving x_i^m and only previous variables
$u_i^m(x_i^m, \cdot) \leftarrow \wedge_{u_{mi} \in \{u' \in \mathcal{U} | x_i^m \in scope(u') \wedge \forall y \in scope(u'), y \succ x_i^m\}} u(x_i^m, \cdot)$
$D'_{mi} \leftarrow D_{mi}$ // a copy of x_i^m's domain D_{mi}
$\bar{u}_i^m \leftarrow 0$ // utility value of the CPA up to and including x_j^{m-1}
$u^* \leftarrow \infty$ // utility value of the best solution found so far
if $m = 1$ **then** $x_i^1 \leftarrow$ first $x_i^{1*} \in D'_{1i}$ such that $u_i^1(x_i^{1*}) < \infty$
    if there exists such a x_i^{1*} **then** send message (CPA, (x_i^{1*}), $u_i^1(x_i^{1*})$) to x_i^2
    else broadcast messages INFEASIBLE
for each received message M **do**
    if M = (UB,$(x_i^{1*}, \ldots, x_i^{n*})$, u) **then**
        $u^* \leftarrow u$ **and** record $(x_i^{1*}, \ldots, x_i^{n*}, u)$ as the best solution found so far
        continue
    if M = (CPA,$(x_i^{1*}, \ldots, x_i^{m-1*})$, u) **then**
        $D'_{mi} \leftarrow D_{mi}$ **and** $(x_i^1, \ldots, x_i^{m-1}, u) \leftarrow (x_i^{1*}, \ldots, x_i^{m-1*})$ **and** $\bar{u}^m \leftarrow u$
    else if M=(BACK) **then** $D'_{mi} \leftarrow D'_{mi} \setminus \{x_i^{m*}\}$
// Look for a (better) value for x_i^m
$x_i^m \leftarrow$ first $x_i^{m*} \in D'_{mi}$ such that $\bar{u}_i^m + u_i^m(x_i^{m*}, \cdot) > u^*$
    if there exists such a x_i^{m*} **then**
        if $m = n$ **then** $x_i^n \leftarrow x_i^{n*} = \arg\min_{xn'_i \in D'_{ni}} \{u_i^n(x_i^{n'}, \cdot)\}$, $u^* \leftarrow \bar{u}_i^n + u_I^n(x_i^{n*}, \cdot)$
        Record $(x_i^{1*}, \ldots, x_i^{n*})$ as the best solution found so far
        Broadcast message M = (UB,$(x_i^{1*}, \ldots, x_i^{n*})$, u^*)
            if $D'_{ni} = 0$ **then** broadcast message TERMINATE
            else send message BACK to x_i^{n-1}
        else send message (CPA, $(x_i^{1*}, \ldots, x_i^{n*})$, $\bar{u}_i^m + u_i^m(x_i^{m*}, \cdot)$) to x_i^{m+1}
    else
        if $m = 1$ **then** broadcast message TERMINATE
        else send message BACK to x_i^{m-1}

Assignment (CPA) message that includes the value assignment to x_i^1 and the current associated utility value to the next agent(line 6-8). Each agent that receives the CPA extends it by including a value assignment of its own variable, as well as the associated utility value based on the utility function it shares with other variable assignments appearing in the received CPA (line 13-14). Whenever a CPA reaches a new full assignment at the last agent (line 19), the accumulated utility value of the CPA is the utility value of the full variable assignment. This utility value will then be broadcast to all other agents, and each agent can use this utility value as an upper bound (UB). When the utility value of a new CPA exceeds the utility value of the currently known upper bound (line 19-21), it will be broadcast to all agents as the new upper bound (line 10-12). Recursively, each agent holding CPA then checks whether its CPA accumulated utility value is larger than the upper bound. If this is true (line 17), it means the old previous variable assignment is sub-optimal, and the agent will assign the next value in the domain of its variable instead of the current value, and send it again to the next agent (line 24) and checks again. An agent encountering an empty domain of values (when all values have been used) erases its assignment (and its utility value) and sends the CPA back to the previous agent (line 22-23). When the domain of the first agent is exhausted, the last discovered full assignment is reported as the optimal solution (line 26-27).

3.4.2 AFB

AFB [115] is similar to SyncBB, in which the search is performed in a synchronous way, but the forward bounding is performed in parallel. AFB uses the same type of CPA as in SyncBB, which contains the assignments of the agents involved, as well as the accumulated utility values of the utility functions among the variables.

Algorithm 2 shows the solution process of AFB. Firstly, variable x_i^m starts by creating an empty CPA and then begins the search process by calling *assign_CPA* (line 29-35). When an agent receives the CPA (line 19), it adds the variable assignments in its local view only when the utility value is larger than the global lower-bound. Otherwise, backtracking is performed and the current agent sends the CPA to the previous agent to revise its variable assignment. As a result, a timestamp mechanism is needed to determine the most current CPA and delete old CPAs. If the CPA's timestamp is not the up-to-date CPA, the message will be discarded. When the message is not discarded, the agent updates the timestamp of the CPA (line 20-21), and saves the received variable assignment (line 22). Only one agent performs an assignment on the CPA at a time. If utility of received variable assignment exceeds the upper-bound, it performs backtrack (line 23), otherwise, *assign_CPA* is called to reassign another value to its variable (line 24).

Procedure *assign_CPA* is used to find a value assignment for the current agent within the current bounds of the CPA. Firstly, clear all estimates related to previous assignments (line 30). Then the agent try to assign every value in the domain that it has not tried yet (line 31-33). The assigned value must make sure that the sum of the utility values in the CPA and the estimate so far are larger than the upper-bound, and then the agent adds the selected value on the CPA (line 35). Otherwise, the assignments of some higher priority agents must be changed, and thus backtrack is called (line 34). If the agent is the last agent, then a complete assignment will be reached, and it is broadcast to all agents through SOLUTION and CPA messages (line 36). Upon receiving SOLUTION message, each agent checks to

Algorithm 2 AFB: Asynchronous Forward Bounding.

Require: a fixed, known, linear ordering of variables $x_i^1 \succ x_i^m \succ \cdots \succ x_i^n$
//Join all utility functions involving x_i^m and only previous variables
$u_i^m(x_i^m, \cdot) \leftarrow \wedge_{u_i^m \in \{u' \in \mathcal{U} | x_i^m \in scope(u') \wedge \forall y \in scope(u'), y \succ x_i^m\}} u(x_i^m, \cdot)$
$D'_{mi} \leftarrow D_{mi}$ // a copy of x_i^m's domain D_{mi}
$\bar{u}_i^m \leftarrow 0$ // utility value of the CPA up to and including x_i^{m-1}
$u^* \leftarrow \infty$ // utility value of the best solution found so far
if $m = 1$ **then** $x_i^1 \leftarrow$ first $x_i^{1*} \in D'_{1i}$ such that $u_i^1(x_i^{1*}) < \infty$
    if there exists such a x_i^{1*} **then** send message (CPA, (x_i^{1*}), $u_i^1(x_i^{1*})$) to x_i^2
    else broadcast messages INFEASIBLE
for each received message M **do**
    if M = (FB_CPA,$(x_i^{1*}, \ldots, x_i^{m-1*})$, u) **then**
        if compare(FB_CPA.$timestamp$, $timestamp$, $m-1$) = 'bigger' **then**
        $timestamp \leftarrow$ FB_CPA.$timestamp$
        $estimate_{mi} \leftarrow$ local utility + future utility for x_i^m
        send message (FB_ESTIMATE, $(x_i^{1*}, \ldots, x_i^{m-1*})$, m, $estimate_{mi}, u$) to x_i^{m-1}
    if M = (FB_ESTIMATE,$(x_i^{1*}, \ldots, x_i^{m+1})$,$m+1$, $estimate_{m+1,i}$) **then**
        if compare(FB_ESTIMATE.$timestamp$, $timestamp$, $m-1$) = 'bigger' **then**
        $estimate_{mi} \leftarrow estimate_{m+1,i}$,
            if u + all saved estimates $\leq UB$ **then** calls *assign_CPA* to change variable assignment
    if M = (CPA,$(x_1^{1*}, \ldots, x^{m-1*} - i)$, u) **then**
        if compare(CPA.$timestamp$, $timestamp$, m) = 'bigger' **then**
        $timestamp \leftarrow$ CPA.$timestamp$
        $D'_{mi} \leftarrow D_{mi}$, $(x_i^1, \ldots, x_i^{m-1}, u) \leftarrow (x_i^{1*}, \ldots, x_i^{m-1*})$, $\bar{u}^m \leftarrow u$,**and** update CPA
            if $\bar{u}_i^{m-1} \leq UB$ **then** backtrack
            else calls *assign_CPA*
    if M = (SOLUTION,$(x_i^{1*}, \ldots, x_i^{n*})$, u) **then**
        if SOLUTION has not already been recorded **then**
        record assignments and optimal utility value
        broadcast TERMINATE
// procedure *assign_CPA*
    clear estimates
    if CPA contains an assignment $x_i^m = w$ **then** remove it
        iterate (from last assigned value) over D_{mi} until found
        $v \in D_{mi}$ such that u_i^m + $estimate$(CPA,m,v) $\geq$ UB
            if no such value exists **then** backtrack
            else assign $x_i^m = v$
                if CPA is full assignment **then** broadcast (SOLUTION,CPA,u)
                $u^* \leftarrow u$ **and** *assign_CPA*
                else send CPA to x_i^{m+1} **and** $\forall n > m$ send (FB_CPA, x_i^m, CPA) to x_i^n
// procedure *backtrack*
    clear estimates
    if $m = 1$ **then** broadcast TERMINATE
    else send CPA to x_i^{m-1}

see if the SOLUTION has already been recorded, and record the variable assignments if necessary and optimal utility value, and terminates all agents (line 27-30). Then this agent continues by updating bound and calling *assign_CPA* (line 37). If it is not the last agent, a CPA message will be send to the next agent (line 38).

Each agent adds its variable assignment in CPA and replies through FB_CPA messages for agent who do not have assignments in the current CPA (line 10). When the agent receives a FB_CPA message, the upper-bound is computed considering the utility of variable assignments on the CPA and the current agent (line 10-14). This estimate utility value is returned to the sender by a FB_ESTIMATE message (line 15-18). Whenever the current agent cannot find a valid value, the agent clears the saved estimates, and sends the CPA backwards to the previous agent. If the agent is the first agent, then broadcast terminates to end the search process in all agents (line 42-44).

3.4.3 DPOP

DPOP [236] is based on dynamic programming. DPOP operates in three phases. Algorithm 3 shows the solution process of DPOP. In the first phase, a depth-first-search (DFS) structure is established using a distributed DFS algorithm. Each variable is considered as a node in this structure. Each agent controls a set of variables. The nodes are consistently labeled as parent or child nodes, and the edges are identified as tree or back edges (line 1). Variable x_i^m, owned by A_m^{as} is also used as an example.

The second phase is called UTIL propagation, the objective of which is to propagate aggregated utilities up the DFS tree. Initially, messages are send by agents up in the DFS tree, propagating information about the aggregated optimal utility values. For example, for each variable x_i^m belonging to agent A_m^{as}, agent A_m^{as} joins the constraints involving x_i^m together, while ignoring all constraints that involve at least one descendant of x_i^m(line 3-4).

Then agent A_m^{as} waits for the reception of a so-called UTIL message from each of the child nodes of x_i^m, and joins them all together with its constraints (line 6-8). Finally, agent A_m^{as} eliminates x_i^m from the join, and sends the resulting utility to the parent node of x_i^m (line 10-13). This utility corresponds to the maximum achieved utility for the whole subtree rooted at x_i^m, as the function of the value for the parent node of x_i^m and also the values for other variables that are higher than x_i^m in the DFS tree (line 13). The set of variables in the UTIL message for variable x_i^m sent by agent A_m^{as} is called the separator $sep(x_i^m)$, which includes x_i^m's parent and pseudo-parents nodes, i.e., the ancestors of x_i^m but not directly connected, as well as x_i^m's descendant's pseudo-parent that are above x_i^m in the DFS tree. Therefore, a UTIL message for variable x_i^m contains the optimal utility obtained in the subtree of each instantiation of separator $sep(x_i^m)$.

The third phase of DPOP is called VALUE propagation. At the end of the UTIL propagation phase, the root variable (the variable node that initiates the DFS tree, with itself as the root) has obtained its own local optimal value based on the messages it received (line 14). The agent that controls the value of the root variable sends this optimal value to each of the child nodes of the root variable through VALUE messages (line 15). Recursively, for each variable x_i^m, when the corresponding agent receives the VALUE message from the parent node of that variable x_i^m, is able to look up its own corresponding optimal value during the UTIL propagation phase (line 16-18). To each of the child nodes of x_i^m, agent A_m^{as} sends not only the optimal value of x_i^m, but also the optimal values for all the variables in x_i^m's chil-

Algorithm 3 DPOP: Dynamic Programming Optimization Protocol.

Require: Set of variables, utility functions, vessel and terminal agents

Phase 1: DFS structure generation: establish DFS

1: Run DFS generation algorithm, each variable is considered as a node and each each agent controls a set of variables. Label nodes as parent/child nodes, edges are identified as tree/back edges.

Phase 2: UTIL propagation: bottom-up UTIL message propagation

2: //Join local utility functions:

3: $p_{x_i^m} \leftarrow parent_{x_i^m}$

4: $m(x_i^m, p_{x_i^m}, \cdot) \leftarrow \sum_{u \in \{u' \in \mathcal{U} | x_i^m \in scope(u') \wedge scope(u') \cap (children_{x_i^m} \cup pseudo-children_{x_i^m}) = \varnothing\}} u(x_i^m, \cdot)$

5: //Join with received messages

6: **for** each $y_i^m \in children_{x_i^m}$ **do**

7: Wait for the message (UTIL,$f_{mi}(x_i^m, \cdot)$) from y_i^m

8: $f(x_i^m, p_{x_i^m}) \leftarrow m(x_i^m, p_{x_i^m}, \cdot) + f_{mi}(x_i^m, \cdot)$

9: //Project out x_i^m:

10: **if** x_i^m is not the root variable **then**

11: $x_i^{m*}(p_{x_i^m}, \cdot) \leftarrow \arg\max_{x_i^m} \{f(x_i^m, p_{x_i^m}, \cdot)\}$

12: $f(p_{x_i^m}, \cdot) \leftarrow \max_{x_i^m} f(x_i^m, p_{x_i^m}, \cdot)$

13: Send the message (UTIL, $f(p_{x_i^m}, \cdot)$) to $p_{x_i^m}$

14: **else** $x_i^{m*} \leftarrow \arg\max_{x_i^m} \{m(x_i^m)\}$

Phase 3: VALUE propagation: top-down VALUE message propagation

15: //Root variable sends its optimal value to children nodes

16: **if** x_i^m is not the root variable **then** receive message (VALUE, $m(x_i^m, p_{x_i^m}, \cdot)$ from parent p_x

17: Wait for message (VALUE, $p^*_{x_i^m}, \cdot)$ from parent p_x

18: $x_i^m \leftarrow x_i^{m*}(p_{x_i^m} = p^*_{x_i^m}, \cdot)$

19: **for** each $y_i^m \in children_{x_i^m}$ **do**

20: Send the message (VALUE, x_i^{m*}, $sep^*_{y_i^m}$) to y_i^m

Algorithm 4 MB-DPOP: Memory Bounded Dynamic Programming Optimization Protocol.

Require: Set of variables, utility functions, vessel and terminal agents, control parameter k
Phase 1: DFS structure generation: establish DFS
1: Run DFS generation algorithm, each variable is considered as a node and each each agent controls a set of variables. Label nodes as parent/child nodes, edges are identified as tree/back edges.
Phase 2: Labeling Phase: delimit high-width clusters
2: **if** $|sep(x_i^m)| \leq k$ **then**
3: **if** $\cup CClists \neq \emptyset$ then label self as cluster root
4: **else** label self as normal node, $CC_{x_i^m} \leftarrow \emptyset$
5: **else** let $N = sep(x_i^m) \setminus \cup CClists$, select a set CC_{new} of $|N| - k$ nodes from N
6: return $CC_{x_i^m} = CC_{new} \cup CClists$
7: send $LABEL_{x_i^m}^{P_{x_i^m}} = [sep(x_i^m), CC_{x_i^m}]$ to $p_{x_i^m}$
Phase 3: UTIL propagation:
8: wait for UTIL messages from all children nodes
9: **if** x_i^m is normal node **then** do Phase 2 and Phase 3 as in DPOP
10: **else** do propagations for all instantiations of $CClists$
11: **if** x_i^m is cluster root **then** update UTIL and CACHE for each propagation,
12: when propagations finish, send UTIL to parent $p_{x_i^m}$
Phase 4: VALUE propagation: top-down VALUE message propagation
13: **if** x_i^m is cluster root **then** find in CACHE the CC^* corresponding to $sep_{x_i^m}^*$
14: assign self according to a cached value
15: send CC^* to nodes in CC via VALUE messages
16: **else** perform last UTIL propagation with CC nodes assigned to CC^*
17: assign self accordingly

dren node's separator, which are contained in the VALUE message it receives (line 19-20). Optimal decisions are hereby propagated down the DFS tree, until all variables have been assigned optimal values.

3.4.4 MB-DPOP

Due to the fact that DPOP is time and space exponential in the induced width of the constraint graph, a hybrid algorithm MB-DPOP that operates with bounded memory is proposed in [238]. MB-DPOP is an extension of the DPOP algorithm, providing tradeoff between the linear number of messages of DPOP and the polynomial memory. A parameter k is introduced to allows the users to specify the maximal message dimensionality. This parameter is chosen such that the available memory at each node is greater than d^k (d: domain size) [236].

MB-DPOP identifies subgraphs of the original problem that have width higher than k, in which it is not possible to perform full inference as in DPOP due to memory limitations. Each of these subgraphs are bounded at the top by the lowest node in the tree that has separator of size k or less. These nodes are defined as *cluster roots (CR)*. *Cycle-cut* (CC) nodes are a subset of nodes such that once removed, the remaining problem has width k or less.

The first phase of MB-DPOP is similar to DPOP, in which a DFS tree is generated. In Phase 2, the nodes in the DFS tree are grouped into clusters of high-width, and a subset of

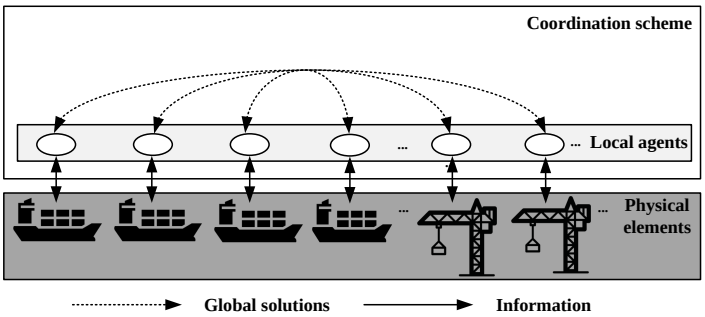

Figure 3.5: Partially-cooperative planning for single-level interaction from Chapter 1.

these nodes are designated to be CC (Cycle-cut) nodes (line 2-4). Each nodes x_i^m waits for *LABEL* message from its children nodes, computes its own label *LABEL* and sends to its parent node (line 7). The *LABEL* message consists of separator of the node, and a list $CC_{x_i^m}$ of CC nodes. The list is computed through a heuristic function based on the separator of the node, and on the lists of CC nodes received from the children (line 5-6). Phase 2 determines the areas on the constraint graph with the width higher than k, and the corresponding CC nodes. In Phase 3, bounded-memory exploration is performed in these areas. In the other areas on the constraint graph, DPOP algorithm applies (line 8-9). If x_i^m is the root of a cluster, agent A_m^{as} creates UTIL message which includes the best utility value the subtree of x_i^m can achieve for all combinations of values in $sep(x_i^m)$. For its children node y_i^m with a larger separator ($|sep(y_i^m)| > k$), the agent which owns y_i^m creates *CACHE* table with one entry $Cache(sep(y_i^m)$ that corresponds to each instantiation of the separator. In the subtree of x_i^m that have sent non-empty *CClists*, it cycles through all instantiations of the *CC* variables in the cluster (line 10).

Node x_i^m updates the optimal utility values found so far for each separator, and also updates the *CACHE* table with the current instantiation of the CC nodes in case a better utility was found (line 11). When all the instantiations are explored, x_i^m sends to its parent the updated UTIL message that now contains the best utility values of x_I^m's subtree for all instantiations of variables in the separator, exactly as in DPOP (line 12).

In Phase 4, if x_i^m is a cluster root, the agent owns x_i^m finds in its CACHE table the optimal assignment corresponding to this particular instantiation of the separator (line 13). This assignment contains its own value, and the values of all CC nodes in the cluster (line 14). Then the agent sends all the CC nodes in the cluster their optimal values through VALUE message (line 15). For the non-CC nodes in the cluster, a final UTIL propagation is required to derive the utility values that correspond to the particular instantiation of the CC nodes that was determined to be optimal (line 16-17).

3.5 Coordination structure based on DCOP

As indicated in Section 3.1, in VRPP, each vessel/terminal is considered as an individual agent that owns a set of variables, and each variable is exclusively controlled by the corresponding agent. In DCOP algorithms, a variable constitutes a variable node. These

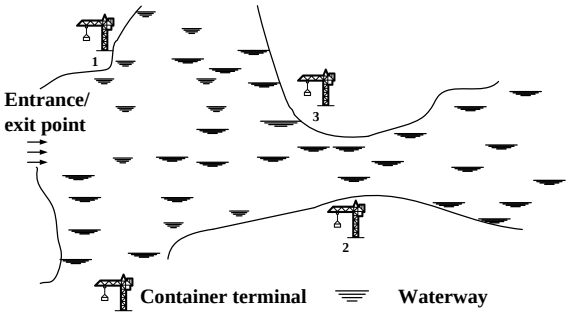

Figure 3.6: Simplified map of Benchmark Layout 1.

variable nodes are connected based on different structures: nodes in SyncBB and AFB are connected with a linear ordering; while the nodes in DPOP and MB-DPOP are connected with a depth-first-search structure. Based on the way the variable nodes are structured, the variable assignments and the corresponding utility values are passed from one node that is owned by an agent to the nodes owned by other agents. Different DCOP algorithms define how the variables assignments and the utility values are passed from one node to another, as well as from one agent to other agents. Therefore, the DCOP-based methods not require a central controller to receive and send information from/to all the agents. This also means that a central coordinator is not necessary required to coordinate the rotations of different vessel agents. With the DCOP-based methods, the interactions among vessel agents and terminal agents take place at a single level.

Moreover, each agent only knows the variable assignments and utility values from the agents with whom it shares inter-agent utility functions. This implies that each agent does not know the variable assignments and utility values from the agents with whom it does not share any inter-agent utility function. Therefore, the proposed DCOP-based methods guarantees to some extent the privacy of vessel or terminal agents, as they do not necessarily need to reveal all the information to all the other agents. This also means that the vessel agents are coordinated in a partially-cooperative way.

Thus, the proposed DCOP methods fulfills partially-cooperative planning for single level interaction, as shown in Figure 3.5. A detailed analysis of the information exchange between agents will be given in the following section.

3.6 Experimental results

The DCOP algorithms are implemented in the latest version of the FRODO2 DCOP toolbox [169]. The number of local vessel/terminal agents equal to the number of vessel/terminal that is considered in the 4 scenarios of Benchmark Layout 1. The distributed system is simulated using an Intel Core i5-2400 CPU with 4GB RAM running Windows 7.

This section presents the comparison of the experimental results for vessel rotation planning based on the non-layered and layered DCOP methods. Benchmark Layout 1 is used to evaluate the proposed methods, as shown in Figure 3.6. Experimental results are measured based on KPIs that were introduced in Chapter 2.5, including computation time (KPI 1),

quality of solutions (KPI 2) and communication load (KPI 3, KPI 4, KPI 5 and KPI 6). As the DCOP algorithms are operated based on the information exchange among agents, it is important to measure the total number of messages exchanged, number of messages exchanged per agent, total amount of information exchanged, as well as amount of information exchanged per agent. Therefore, KPIs 3–6 are important KPIs to measure the algorithmic performance of the DCOP algorithms.

3.6.1 Experimental settings

Four different scenarios with Benchmark Layout 1 are considered. For each scenario, 10 experiments are set up with different parameters for utility values U_{ik}^m and W_{ik}^m, traveling time T_{ij}^{travel}. Utility value U_{ik}^m is given an integer value in $[3,5]$ when $x_i^m = k$ to represent preferences of vessels. Utility value W_{ik}^m is also given an integer value in $[2,5]$ when $w_i^m = k_s$ and $x_i^m = k$.

For example, for a vessel agent m that will visit terminal 1, 2 and 3 during 3 time slots, variables x_1^m, x_2^m and x_3^m represents the preferences of i for visiting terminal 1, 2 and 3, respectively. In the corresponding utility function r_{m1}, utility value $U_{11}^m = 5$ when $x_1^m = 1$, $U_{12}^m = 4$ when $x_1^m = 2$, and $U_{13}^m = 3$ when $x_1^m = 3$. This means vessel m prefers most to visit terminal 1 in the first time slot, and prefers least to visit terminal 1 in the third time slot. Similar procedures of setting utility values are carried out for variable x_2^m and x_3^m. As different vessels may have different preferences for visiting sequences, the utility values for U_{21}^m, U_{22}^m and U_{23}^m, as well as for U_{31}^m, U_{32}^m and U_{33}^m may differ. In addition, w_1^m, w_2^m and w_3^m represents the waiting time of m at terminal 1, 2 and 3. In the corresponding utility function r_{m6}, $W_{11}^m = 5$ when $w_1^m = 1$, $W_{12}^m = 4$ when $w_1^m = 2$, and $W_{13}^m = 3$ when $w_1^m = 3$. The same applies for variable w_2^m and w_3^m. Utility values W_{11}^m, W_{21}^m and W_{31}^m are always given higher values since vessels always prefer shorter waiting times.

Traveling time T_{ij}^{travel} is given different integer values in $[2,5]$ to represent different distances between terminals. Service time s_i^m is considered as constants, and is 2 time teps (2h). For each of the experiments the following algorithms are used to determine rotation plans: non-layered SyncBB, layered SyncBB, non-layered AFB, layered AFB, non-layered DPOP, layered DPOP, non-layered MB-DPOP, and layered MB-DPOP.

3.6.2 Generated rotation plan

Based on the solutions obtained from the experiments, the arrival time, departure time, service time and travel time for each vessel from variable a_i^m, d_i^m, s_i^m and T_{ij}^{travel} are determined, respectively. This section only presents an example of the rotation plans generated using DPOP in both assignment and scheduling problem in the Scenario 2 as shown in Figure 3.7. In this figure, different colors represent different terminals. It can be seen that when a vessel does not arrive at the terminal within the corresponding time window, it has to wait at the terminal until the next available time window of the terminal. From this rotation plan, each vessel operator can determine the sequence in which to visit the different terminals and exact arrival and departure time.

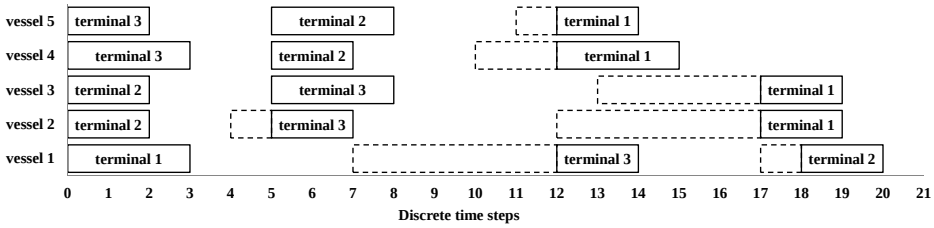

Figure 3.7: An example of rotation plans (5 vessels, 3 terminals).

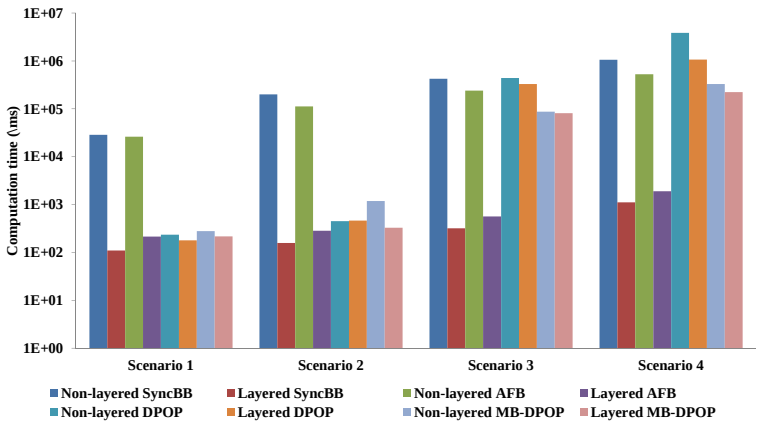

Figure 3.8: Comparison of computation time (KPI 1).

3.6.3 Computation time

Figure 3.8 shows the computation times that were spend in solving the VRPP in different scenarios with different algorithms. It is clear that the computation time for all algorithms increases when the size of the scenario increases in the four scenarios. For SyncBB and AFB, regardless of it is non-layered or layered, the computation time increases steadily in all scenarios, while the computation time of DPOP increases substantially when the problem size increases. The reason is that for SyncBB and AFB, the memory requirements are polynomial in the number of agents, and the memory requirements for DPOP is exponential in the induced-width the DCOP problem, which depends on the number of backedges (the links between nodes and pseudo-parents). As a result, DPOP performs best when the problem size is smaller as in Scenario 1 and Scenario 2. When the problem size increases it is no longer the best option. In Scenario 4, it performs worst with the maximum computation time. Layered DPOP has a similar pattern. MB-DPOP is an extension of DPOP: it performs full inference in areas of induced width (number of back edges) lower than k, and bounded inference in areas with width higher than k. Subsequently, the memory requirement of MB-DPOP is less than that of DPOP. In our experiments, the computation time of MB-DPOP increase steadily. Among all non-layered DCOP algorithms, MB-DPOP performs best when the problem size increases as in Scenario 3 and Scenario 4.

All layered DCOP algorithms require shorter computation time than non-layered DCOP

Table 3.3: Solution quality of layered DCOP compared with optimal solutions (KPI 2).

	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Max (%)	14	12.2	10.9	10
Average (%)	6.2	3.2	5.4	3.4
Optimality (%)	70	80	100	90

algorithms, in which layered SyncBB and AFB have significant improvements with shorter computation time compared with non-layered SyncBB and AFB, while layered DPOP does not have significant improvements. Among all layered DCOP algorithms, layered SyncBB performs best as it always requires the shortest computation time.

3.6.4 Quality of solutions

Since SyncBB, AFB, DPOP and MB-DPOP are all complete algorithms, the solution quality of non-layered SyncBB, AFB, DPOP and MB-DPOP, are all optimal and have the same utility value. However, since the partition of DCOP framework can be at the cost of lower solution quality, layered DCOP sometimes generate solutions with lower solution quality. As a result, layered SyncBB, AFB, DPOP and MB-DPOP, sometimes have lower solution quality, as is shown in Figure 3.9 and Table 3.3.

Figure 3.9 shows the ratio of utility values of the solutions of different algorithms over the utility value of optimal solution generated in each experiment. This chapter define this ratio as deterioration ratio, which is used to evaluate how much the solution quality of a algorithm is less than the optimal solution. It equals to the utility value of the solution of each algorithms, divided by the utility value of optimal solution in that experiment. This figure can be used to evaluate the solution quality of the non-layered and layered DCOP methods. This figure shows that the results of 40 experiments with four different scenarios. Table 3.3 shows the maximum and the average deterioration ratios of the utility value of the solutions in percentage, as well as optimality of the layered DCOP method. The maximum and average deterioration ratios refer to the maximum and average deterioration ratios obtained in the 10 experiments of each scenario. The optimality of the layered DCOP equals to the percentage of experiments in which the solution quality of layered DCOP is worse than non-layered DCOP in all 10 experiments of each scenario. It is observed that the solution quality of the layered DCOP is worse than the non-layered DCOP under most circumstances, as is to be expected.

3.6.5 Information exchange analysis

Table 3.4 presents a qualitative analysis of information exchange with respect to different DCOP algorithms. Here, the information in the table refers to the information exchanged between agents. The information that is exchanged among the variables within one agent is not included. It can be seen that the content of information exchanged between vessel agents is similar, including their potential visiting sequences (x_i^m) to different terminals and the corresponding utility values ($U_{x_i^m}^m$). This is because x_i^m are involved in the inter-agent utility function r_1^{ia}. For vessel agent m, it needs to know the variable assignments from the agents that it shares inter-agent utility functions with, so as to determine its own variable

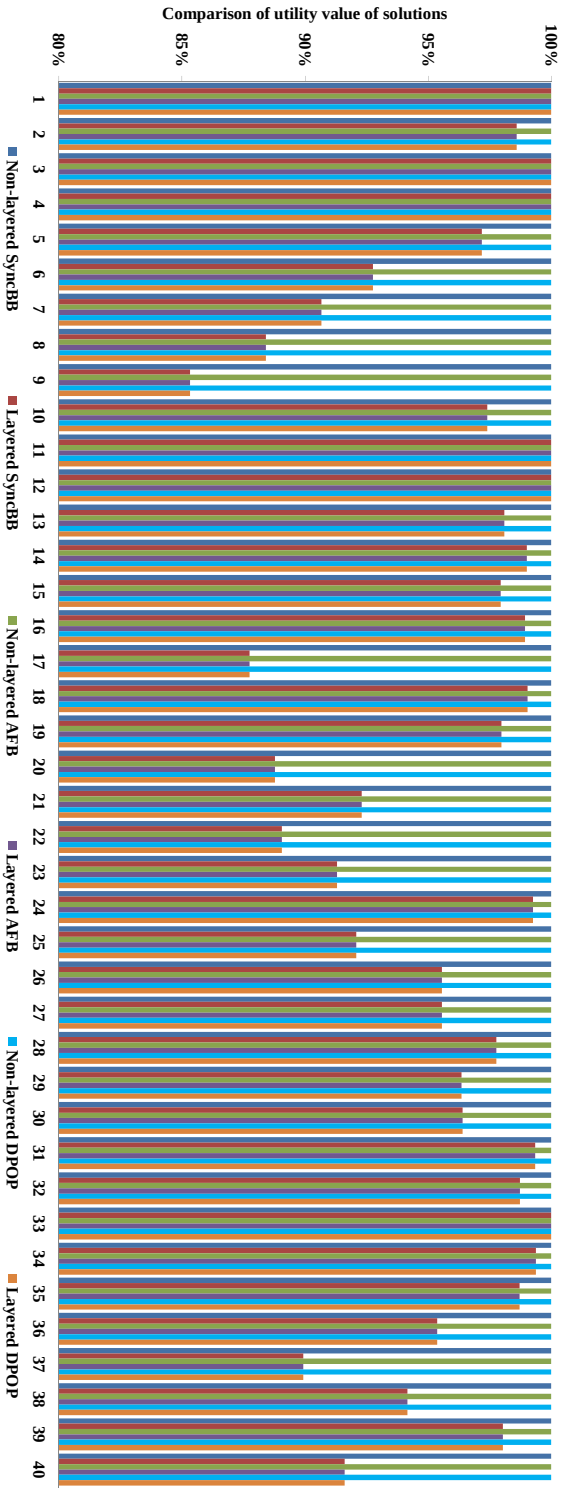

Figure 3.9: Comparison of quality of solutions (KPI 2).

Table 3.4: Information exchange of different DCOP algorithms.

	Exchanged messages between agents	Involved variables	Involved information
SyncBB	CPA	x_i^m, $U_{i,x_i^m}^m$, $(\forall m \in M, \forall i \in N_m)$	The currently chosen visiting sequences of a vessel agent m and the associated utility value
	UB	$\sum_{m=1}^{\lvert M \rvert} U_{i,x_i^m}^m$, $i \in N_m$	The sum of the utility values of the currently chosen visiting sequences from all vessel agents
AFB	CPA	x_i^m, $U_{i,x_i^m}^m$, $(\forall m \in M, \forall i \in N_m)$	The currently chosen visiting sequences of a vessel agent m and the associated utility value
	FB-CPA	x_i^m, $U_{i,x_i^m}^m$, $(\forall m \in \text{CPA}, \forall i \in N_m)$	The currently chosen visiting sequences of a subset of vessel agents that have added their currently chosen visiting sequences in the CPA
	FB-ESTIMATE	$U_{i,x_i^m}^m$ $(\forall m \notin \text{CPA}, \forall i \in N_m)$	The estimated utility values calculated by a subset of vessel agents who have not added their currently chosen visiting sequences in the CPA
	SOLUTION	x_i^m $(\forall m \in M, \forall i \in N_m)$	The currently chosen visiting sequences of all vessel agents
DPOP	UTIL	$U_{i,x_i^{m_1}}^{m_1*}, U_{i,x_i^{m_2}}^{m_2*}$ $(m_1 \in M, \forall m_2 \in Separator(m_1), \forall i \in N_m)$	The current local optimal utility values obtained from a subset of vessel agents in the DFS structure
	VALUE	$x_i^{m_1*}, x^{m_2*}$ $(m_1 \in M, m_2 \in Separator(children_{m_1}))$	The optimal visiting sequences of a subset of vessel agents
MB-DPOP	LABEL	$\forall m_1 \in CClists(children(m))$, $\forall m_2 \in Separator(m)$	For vessel agent m, it receives a list of vessel agents that have been labeled as CC (Cyclic cut) nodes and a list of vessel agents that have been labeled as separators in the DFS structure
	UTIL	$U_{i,x_i^{m_1}}^{m_1*}, U_{i,x_i^{m_2}}^{m_2*}$ $(m_1 \in M, \forall m_2 \in Separator(m_1), \forall i \in N_m)$	The current local optimal utility values obtained from a subset of vessel agents in the DFS structure
	VALUE	$x_i^{m_1*}, x^{m_2*}$ $(m_1 \in M, m_2 \in Separator(children_{m_1}))$	The optimal visiting sequences of a subset of vessel agents

assignments. Therefore, variable x_i^m is an important part of the information exchanged between the agents. A detailed quantitative analysis of the communication load with respect to different DCOP algorithms will be given in the next.

3.6.6 Communication load

This section presents a quantitative analysis of the total information exchange of different DCOP algorithms as well as information exchange per agent.

Total information exchange

Figure 3.10 and Figure 3.11 show the total number of messages exchanged between agents and the total amount of information (in terms of the size of messages) exchanged between agents, respectively. In all experiments, the number and amount of messages of all algorithms increases when the number of terminals increases. In Figure 3.10, comparing with the non-layered algorithms, layered SyncBB, DPOP and MB-DPOP do not have obvious improvements with respect to the number of messages exchanged, while layered AFB has substantial improvements with fewer number of exchanged messages comparing with non-layered AFB. In Figure 3.11, with the increase of problem size going from Scenario 1 to Scenario 4, layered AFB also has improved performance with less information exchange compared with non-layered AFB. From Figure 3.10 and Figure 3.11, among algorithms SyncBB, AFB, DPOP and MB-DPOP, non-layered and layered AFB perform the worst with the maximum number and amount of messages exchanged, while non-layered and layered DPOP always performs best with minimum number of message exchanged in all scenarios. In addition, non-layered and layered SyncBB performs the best with least amount of information with the increase of problem size.

Table 3.5 shows the ratio of the total number of messages of each algorithm to the minimum total number of messages generated in each case. Layered DCOP algorithms have significant improvements with fewer number of messages exchanged, except for layered DPOP and layered MB-DPOP, which is almost the same as non-layered DPOP. However, the situation in Table 3.6 with respect to the amount of information exchanged is different. Table 3.6 shows the ratio of the total amount of information of each algorithm to the least amount of information generated in each case. Layered DPOP has improvements compared with non-layered DPOP with the increased problem sizes. In Scenario 1 and Scenario 2, layered DPOP performs the best, while in Scenario 3 and Scenario 4, SyncBB performs best with the least amount of information. The reason is that the size of messages of non-layered and layered DPOP increase substantially with the increase of the problem sizes. For DPOP, UTIL messages aggregate from the communication between parent agents and childeren agents in the DFS tree, which is exponential in the induced width of the DFS tree used. Thus, it generates exponentially larger messages with the increase of problem sizes. For SyncBB, the increase on size of messages is linear. Unlike DPOP, SyncBB is based on a linear ordering of agents instead of a DFS tree, so the amount of information size increase linearly instead of exponentially.

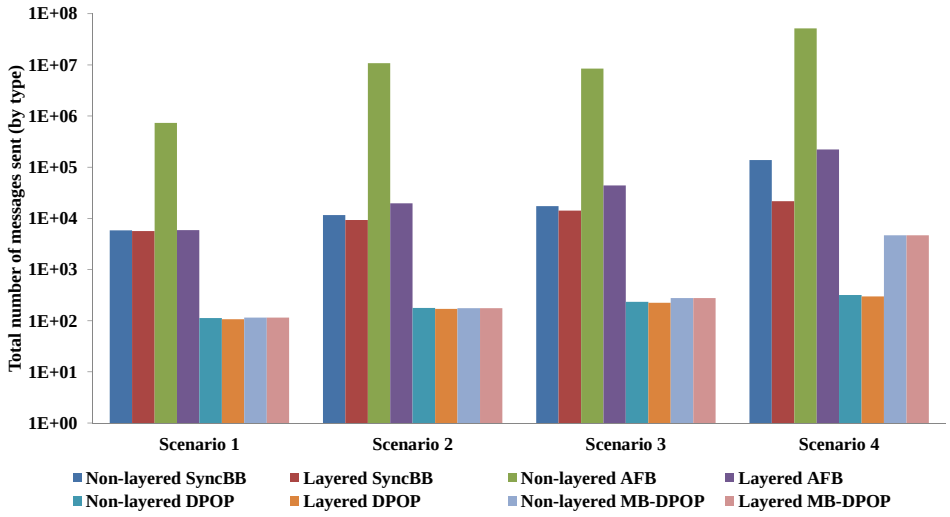

Figure 3.10: Total number of messages exchanged (by type) (KPI 3).

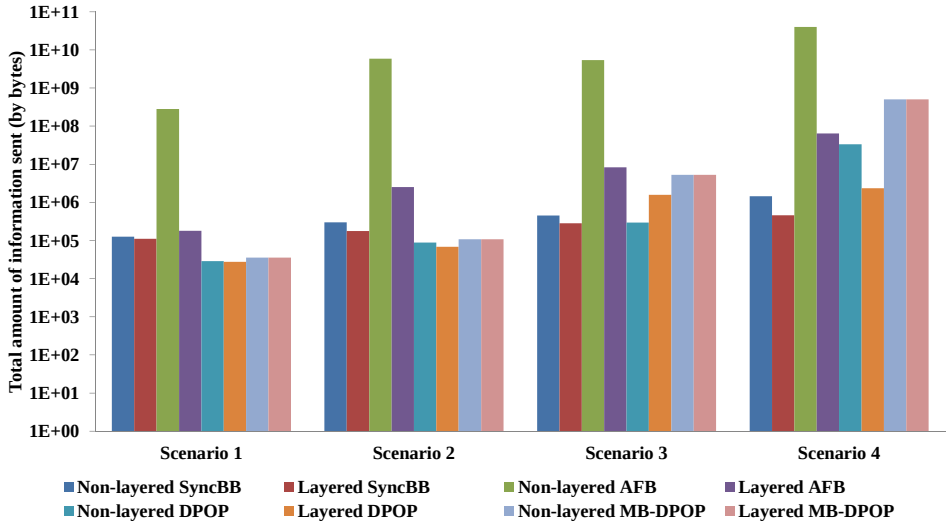

Figure 3.11: Total amount of information exchanged (by bytes) (KPI 5).

Table 3.5: Ratio of total number of messages (KPI 3) of each algorithm to minimum messages generated in each case.

	Non-layered SyncBB	Layered SyncBB	Non-layered AFB	Layered AFB	Non-layered DPOP	Layered DPOP	Non-layered MB-DPOP	Layered MB-DPOP
Scenario1	54.69	52.67	6837.06	55.21	1.05	1.00	1.07	1.07
Scenario2	65.75	52.62	62848.74	115.65	1.03	1.00	1.02	1.02
Scenario3	77.23	63.27	37508.79	194.97	1.04	1.00	1.23	1.23
Scenario4	462.86	72.16	172190.86	741.36	1.06	1.00	15.67	15.67

Table 3.6: Ratio of amount of information (KPI 5) of each algorithm to the smallest size generated in each case.

	Non-layered SyncBB	Layered SyncBB	Non-layered AFB	Layered AFB	Non-layered DPOP	Layered DPOP	Non-layered MB-DPOP	Layered MB-DPOP
Scenario1	4.56	4.02	10140.80	6.51	1.05	1.00	1.28	1.28
Scenario2	4.35	2.58	85684.10	36.94	1.29	1.00	1.57	1.57
Scenario3	1.62	1.00	19138.72	29.35	5.57	1.05	18.76	18.76
Scenario4	3.17	1.00	86496.14	139.14	73.26	5.14	1098.43	1098.43

Information exchange per agent

There are two types of agents in our model formulation, including terminal agents and vessel agents. Figure 3.12-3.19 show the information exchange between individual agents in terms of number and size of messages. In Figure 3.12 and Figure 3.13, information exchanged of terminal agents are almost evenly distributed. Among those algorithms, layerd DPOP has minimum number of messages and minimum amount of information exchanged, while non-layered AFB has maximum number of messages and amount of information exchanged.

In Figure 3.14 and Figure 3.15, the information exchange is similar to Figure 3.12: for the same type of agents, the number of messages and amount of information are evenly distributed in all algorithms except for non-layered and layered AFB. In Figure 3.16, the number of messages and amount of information exchanged per agent are also even for same types of agents except for layered AFB, while in Figure 3.17, only layered SyncBB has evenly amount of information exchange per agent. In Figure 3.18, only layered SyncBB, non-layered and layered DPOP has evenly distributed number of exchanged messages. For the amount of information exchanged in Figure 3.19, it is no longer balanced for vessel agents for all algorithms, only terminal agents have even information exchange.

3.6.7 Results analysis

For the different DCOP methods, it can be concluded from simulation results that, firstly, layered DCOP (method M1.2) always has a lower solution quality than non-layered DCOP (method M1.1); secondly, the improvements of the layered DCOP do not have obvious improvements on the number of messages sent/received except for AFB, however, layered DCOP outperforms non-layered DCOP with respect to the amount of information sent/received; thirdly, layered DCOP outperforms the non-layered DCOP method on computation time.

For the four non-layered and layered DCOP solution algorithms, layered SyncBB seems to be the best option in general, considering its short computation time and less communication load. It gives quick solutions even for larger problem instances. Nevertheless, layered DCOP always has lower performance on the quality of solutions comparing with non-layered, under certain circumstances that require high-quality solutions, it might not always be the best choice. Thus, for smaller problem instances, non-layered DPOP is the best option as it gives optimal solutions with short computation time. For larger problem instances, non-layered MB-DPOP performs best among all non-layered DCOP algorithms

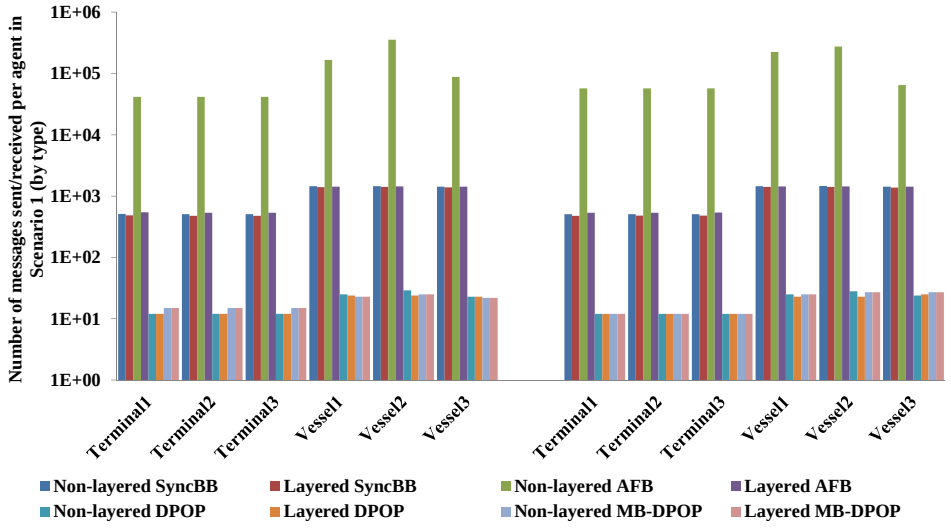

Figure 3.12: Number of messages sent/received per agent (KPI 4) in Scenario 1 (by type).

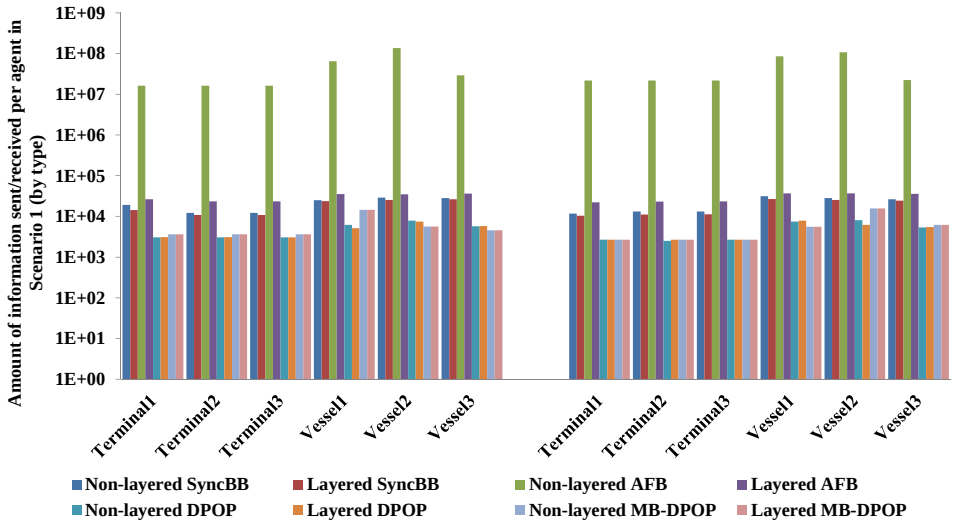

Figure 3.13: Amount of information sent/received per agent (KPI 6) in Scenario 1 (by bytes).

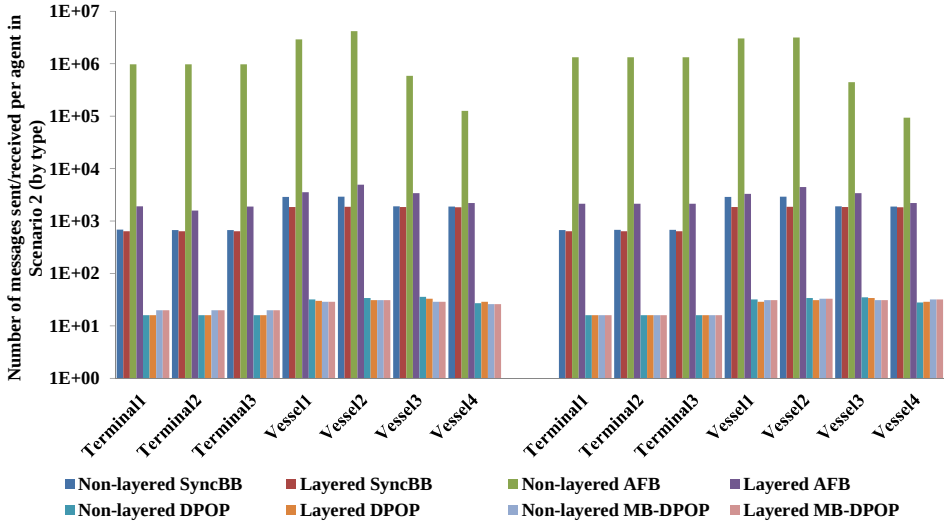

Figure 3.14: Number of messages sent/received per agent (KPI 4) in Scenario 2 (by type).

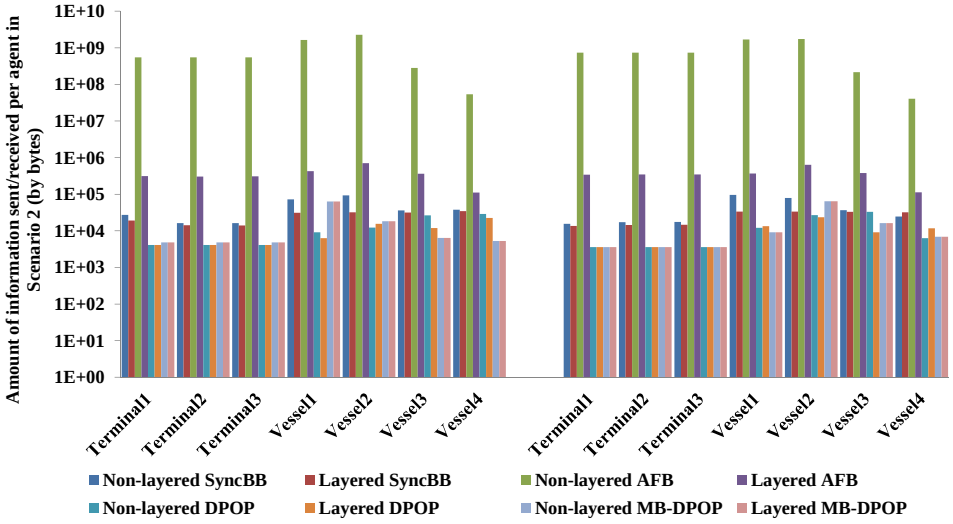

Figure 3.15: Amount of information sent/received per agent (KPI 6) in Scenario 2 (by bytes).

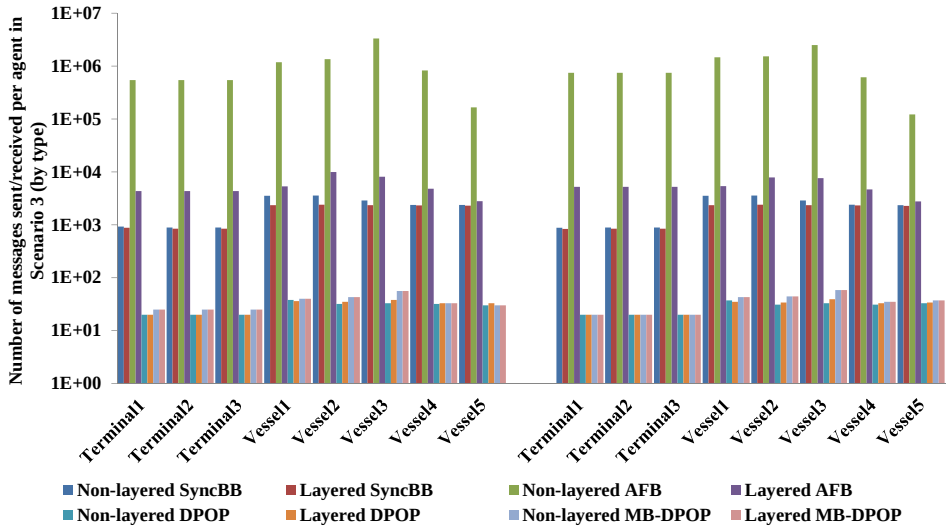

Figure 3.16: Number of messages sent/received per agent (KPI 4) in Scenario 3 (by type).

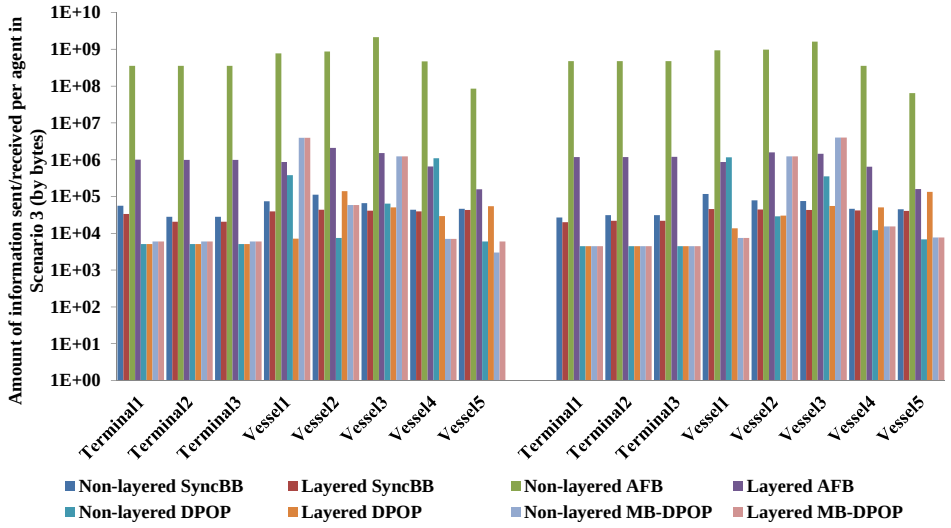

Figure 3.17: Amount of information sent/received per agent (KPI 6) in Scenario 3 (by bytes).

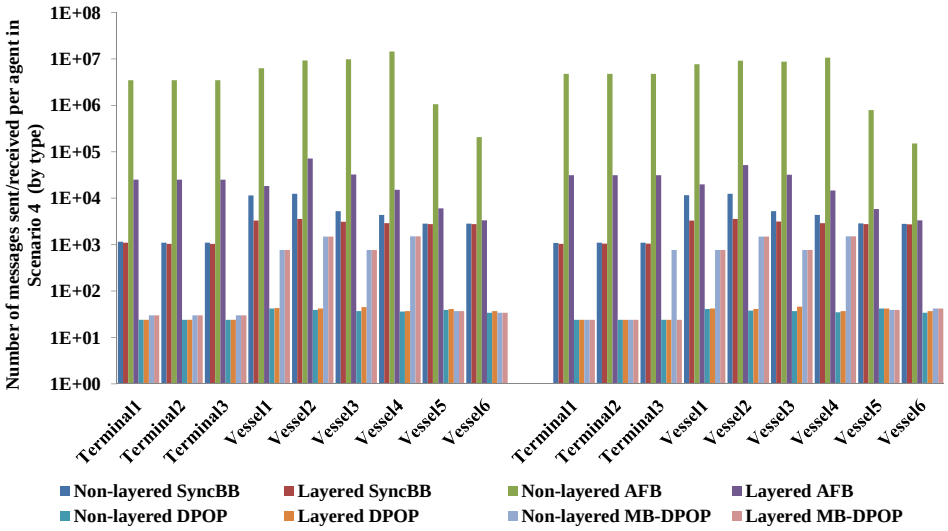

Figure 3.18: Number of messages sent/received per agent (KPI 4)in Scenario 4 (by type).

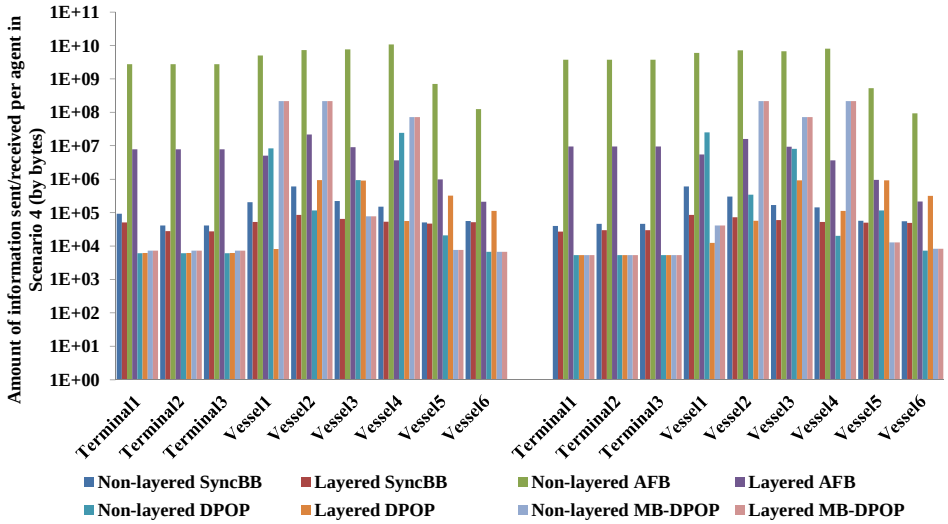

Figure 3.19: Amount of information sent/received per agent (KPI 6) in Scenario 4 (by bytes).

considered. As memory requirements are not an issue in our realistic settings, the MB-DPOP algorithm shows only slightly better performance with respect to the computation time than the DPOP algorithm in our experiments. For even larger problem instances with memory limits, the MB-DPOP algorithm could show a larger improvement. In addition, in problem settings where the communication capacity is limited, and allows for longer computation time, non-layered and layered DPOP will the best option as they always require relatively less number of messages and amount of information.

3.7 Conclusions

In this chapter, distributed methods M1.1 and M1.2 based on DCOP are proposed for tackling the coordination problem of inland vessels in small ports. The performance of these methods was investigated in several case studies with respect to the computation time (KPI 1), quality of solutions (KPI 2) and communication load (KPIs 3, 4, 5 and 6).

It can be concluded from simulation results that: firstly, layered SyncBB in method M1.2 is the best option for solving the VRPP, as it finds quicker and good rotation plans for inland vessels without having to spend long waiting time and heavy communication load. Although the rotation plans generated using the layered DCOP method M1.2 are not guaranteed to be optimal, the deviation from optimality is only around 5%, which is acceptable considering its shorter computation time. On the other hand, for situations in which the vessel operators always prefer optimal rotation plans and long computation time is not an issue, non-layered DPOP in method M1.1 is the best option for small-sized problems, and non-layered MB-DPOP in method M1.1 is the best option for larger-sized problems that have running memory limits. Besides, for situations in which long computation time is acceptable, but the communication capacity is limited, non-layered DPOP in method M1.1 and layered DPOP in method M1.2 are the best options, as these algorithms involve fewer number of messages and less amount of information exchanged. This chapter answers partially the Key Research Questions 2 and 3 and mainly applies for the VRPP in small ports. For larger ports, the DCOP-based methods (methods M1.1 and M1.2) are not able to generate solutions with a reasonable amount of time. Therefore, in Chapters 4 and 5, different solution methods are proposed to solve the VRPP in medium and large ports, respectively.

Chapter 4

Partially-cooperative planning for multi-level interaction in medium-sized ports

As indicated in Chapter 3, it is difficult to find solutions for the coordination problem of inland vessels in larger sizes using a distributed approach. Meanwhile, a centralized approach would require fully-shared information among the inland vessels, which seems to be unacceptable for real operators due to the concern of information privacy. To consider the privacy issue and enlarge the problem size that can be solved, this chapter therefore decomposes the problem using a two-phase approach. Each vessel first decides on locally optimal solution and then partly shares the information with other vessels. This approach thereby involves multi-level interactions and the vessel agents are coordinated in a partially-cooperative way. Three types of solution methods M2.1, M2.2 and M2.3 are proposed. This chapter also considers containers that need to be transported between terminals inside the port area, referred to as inter-terminal transport (ITT) containers. The proposed approach in this chapter provides the inland vessel operators the option to transport a number of ITT containers besides conventional hinterland containers, as a potential solution for alleviating congestion of ITT on roads.

The research discussed in this chapter is based on [180–182, 185].

4.1 Introduction

This chapter proposes a two-phase approach that integrates mixed-integer programming (MIP) with coordination rules and constraint programming (CP) for tackling the coordination problem of inland vessels, as shown in Figure 4.1. The first phase is defined as single vessel optimization problem. In this phase, for each vessel there is a vessel agent that is in charge of its local problem. Hence, multiple unconnected local problems for different vessels are considered. To benefit from the possible available space for carrying containers on inland vessels, this chapter investigates the possibility for carrying optional inter-terminal containers besides the hinterland containers. The problem is formulated for each vessel

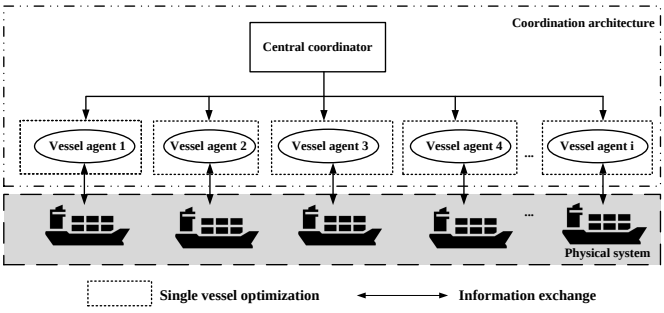

Figure 4.1: The proposed integrated framework.

Table 4.1: List of solution methods.

Methods	1st Phase	2nd Phase
M2.1	MIP	Coordination rules
M2.2	MIP	CP solver
M2.3	MIP	LNS-based heuristics

agent, considering constraints on vessel capacity, required number of containers to be transported, as well as time constraints for arrival and departure. The objective of this phase is to minimize the round-trip time of each vessel for loading and unloading the required number of containers at each terminal in the port, while transporting as many inter-terminal containers as possible. The solution obtained after solving the single vessel optimization problem for an individual vessel agent is a rotation plan. Here, not only use the optimal solution is saved, but also a set of feasible but not optimal solutions. Those solutions are all possible rotation plans for the vessels, and the set of possible solutions is referred as a solution pool. It could happen that the locally optimal rotation plans are conflicting with one another at certain terminals. If one vessel takes the priority, then the other vessels have to wait or adjust their rotation plans accordingly. This could cause domino effects that increase the total round-trip time substantially. To reduce such conflicts, coordination between the vessel agents is needed.

Consequently, in the second phase, a multiple vessel coordination problem is defined, which aims to find better rotation plans after considering rotation plans of the other vessels. It is assumed that there is a central coordinator that is in charge of the multiple vessel coordination problem, to whom the vessel agents send the solutions obtained from the single vessel optimization.

To solve the multiple vessel coordination problem, this chapter proposes three types of methods; one type of method is based on coordination rules; one type of method is based on CP; and the other type of method is based on large neighborhood search (LNS) heuristics. Therefore, this chapter proposes three types of solution methods to solve the problem formulated using the two-phase approach, as shown in Table 4.1.

The method type that includes single vessel optimization in the first phase and multiple vessel coordination using coordination rules in the second phase is referred as method M2.1. The method type that includes single vessel optimization in the first phase and multiple vessel coordination using CP in the second phase is referred as method M2.2. The method type

Table 4.2: Parameters for the single vessel optimization.

Symbols	Definitions
M	set of vessels entering the port
$M^{+\infty}$	a very large positive number
N_m	the set of terminals that vessel m needs to visit in the port
T_i^{entrance}	the traveling time from the entrance of the port to terminal i
T_{ij}^{travel}	traveling time for vessel m from terminal i to j
l_i^m/u_i^m	number of hinterland containers that need to be loaded/unloaded by vessel m at terminal i
C_m^{capacity}	carrying capacity of vessel m in TEU
C_m^{original}	initial number of containers on vessel m before entering the port in TEU
R_{ij}^m	number of inter-terminal containers that need to be transported from terminal i to j by vessel m
$t_i^{\text{load}}/t_i^{\text{unload}}$	average loading/unloading time, per loaded container at terminal i

that includes single vessel optimization in the first phase and multiple vessel coordination using LNS heuristics in the second phase is referred as method M2.3.

In method M2.1, two coordination rules, including preference-based and utility-based rules, are implemented as the interaction rules between the vessel agents. Given the set of feasible solutions obtained from single vessel optimization, vessel agents exchange information of the possible visiting sequences, arrival/departure times at different terminals based on the coordination rules, to estimate their possible waiting times and round-trip times for different solutions, so as to decide the best option of solutions. Method M2.2 use a CP solver to minimize the total round-trip time of all vessels. In method M2.3, three large neighborhood search (LNS)-based heuristics are also proposed.

With this two-phase approach, the agents can have a better coordination plan without sacrificing to share the important decision-related information such as the number of containers each vessel loads/unloads per terminal and the total time of stay in the port area. Therefore, each vessel agents are coordinated in a partially-cooperative way. In addition, this approach reduces the problem size by decomposing the problem into two parts, and thereby can solve vessel rotation planning problems of larger sizes.

This chapter is organized as follows. Section 4.2 presents the mathematical formulation of single vessel optimization and the corresponding solution method. Section 4.3 introduces the mathematical description of multiple vessel coordination and the proposed solution methods M2.1, M2.2, and M2.3. Section 4.4 presents the experimental results and compares the performance of the different solution methods with respect to solution quality and three logistical performance indicators. Section 4.5 concludes this chapter.

4.2 Single vessel optimization

Table 4.2 summarizes the parameters and decision variables used in the first phase. Two types of containers are considered: the mandatory hinterland containers that need to be loaded and unloaded at different terminals and the optional inter-terminal containers that need to be transported between terminals i and j in the port.

Table 4.3: Decision variables for the single vessel optimization.

Symbols	Definitions
y_{ij}^m	$y_{ij}^m = 1$ if vessel m will travel from terminal i to j, otherwise $y_{ij}^m = 0$
k_i^m	the segment at which vessel m visits terminal i
α_i^m	arrival time of vessel m at terminal i
β_i^m	departure time of vessel m from terminal i
b_i^m	number of containers on-board of vessel m when it departs from terminal i
v_{ij}^m	number of inter-terminal containers directly transported from terminal i to terminal j by vessel m
μ_i^m	$\mu_i^m = 1$ if vessel m visits terminal i in the first, otherwise $\mu_i^m = 0$
σ_i^m	$\sigma_i^m = 1$ if vessel m visits terminal i in the end, otherwise $\sigma_i^m = 0$

For a inland vessel m, its vessel agent has information on the set of terminals N_m to be visited; the number of containers to be loaded (l_i^m) and unloaded (u_i^m) at each terminal; and the distance between any two terminals, represented by T_{ij}^{travel}. The capacity of vessel m is defined as C_m^{capacity} and the initial number of containers on-board as C_m^{original}. In addition, the average loading and unloading times per container at a terminal are also considered known, represented by t_i^{load} and t_i^{unload}, respectively. This chapter assumes that each terminal will be visited for exactly once, and that every vessel enters and leaves the port via a port entrance and exit point. The traveling time from terminal i to the port entrance/exit point by vessel m is represented by T_i^{entrance}.

To determine the arrival, departure time, visiting sequence, and the number of inter-terminal containers that a vessel can transport, decision variables are summarized in Table 4.3. Variable y_{ij}^m is used to determine the sequence in which a vessel visits terminals. A rotation plan is defined as a series of segments. For example, if vessel m needs to visit 10 terminals, then there are 10 segments in its rotation plan. Variable k_i^m is used to construct a rotation plan. As an example, $k_i^m = 4$ represents that at vessel m will visit terminal i the fourth according to its rotation plan. As vessel m needs to visit N_m terminals, therefore $k_i^m \in \{1, 2, \ldots, N_m\}$. Variables α_i^m and β_i^m together constitute the arrival/departure time window at terminal i for vessel m. Variable v_{ij}^m represents the number of inter-terminal containers that are directly transported from terminal i to j by vessel m.

The total round-trip time of a vessel includes the service time and the travel time. Waiting time is not considered in this phase, instead, it is considered in the second phase. To take into account the number of optional inter-terminal containers transported in the objective function, the difference between required and actual number of containers transported is transferred into a penalty time. Parameters p_1 and p_2 are the weights assigned to round-trip time and penalty time, respectively. The penalty time represents the time that could have been spent on the inter-terminal containers, which is the sum of the unloading and loading time of the inter-terminal containers that have not been transported. Minimizing this penalty time implies giving preference to maximizing the number of inter-terminal containers carried by the vessel. By introducing weights this approach can balance minimizing the total round-trip time with maximizing the total number of containers transported. Choosing a value for p_1 and p_2 depends on the vessel operator's decision. A higher p_1 represents that the vessel operator is more interested in minimizing the total round-trip time, while a higher p_2 represents that the vessel operator is more willing to also carry inter-terminal

containers besides hinterland containers to gain profits. The complete problem for vessel m is formulated as follows:

$$\min p_1 \left((\sum_{i\in N_m}\sum_{j\in N_m} v_{ij}^m \left(t_i^{\text{unload}} + t_i^{\text{load}}\right) + \sum_{i\in N_m}\sum_{j\in N_m} y_{ij}^m T_{ij}^{\text{travel}} + \sum_{i\in N_m} u_i^m t_i^{\text{unload}} + \sum_{i\in N_m} l_i^m t_i^{\text{load}} \right)$$
$$+ p_1 \left((\sum_{i\in N_m} \mu_i^m + \sum_{i\in N_m} \sigma_i^m) T^{\text{entrance}} \right) + p_2 \left(\sum_{i\in N_m}\sum_{j\in N_m} (R_{ij}^m - v_{ij}^m) \left(t_i^{\text{unload}} + t_i^{\text{load}}\right) \right) \tag{4.1}$$

subject to

$$v_{ij}^m \le R_{ij}^m \sum_{q\in N_m} y_{iq}^m \qquad \forall i,j \in N_m, \forall m \in M \tag{4.2}$$
$$v_{ji}^m \le R_{ji}^m \sum_{q\in N_m} y_{qi}^m \qquad \forall i,j \in N_m, \forall m \in M \tag{4.3}$$
$$\beta_i^m = \alpha_i^m + \sum_{q\in N_m} v_{iq}^m t_i^{\text{load}} + \sum_{p\in N_m} v_{pi}^m t_i^{\text{unload}} + u_i^m t^{\text{unload}} + l_i^m t_i^{\text{load}} \qquad \forall i \in N_m, \forall m \in M \tag{4.4}$$
$$y_{ij}^m + y_{ji}^m \le 1 \qquad \forall i,j \in N_m, \forall m \in M \tag{4.5}$$
$$k_i^m + 1 \le k_j^m + M^{+\infty}(1 - y_{ij}^m) \qquad \forall i,j \in N_m, \forall m \in M \tag{4.6}$$
$$k_i^m + 1 + M^{+\infty}(1 - y_{ij}^m) \ge k_j^m \qquad \forall i,j \in N_m, \forall m \in M \tag{4.7}$$
$$\alpha_j^m + M^{+\infty}(1 - y_{ij}^m) \ge \beta_i^m + T_{ij}^{\text{travel}} \qquad \forall i,j \in N_m, \forall m \in M \tag{4.8}$$
$$\alpha_j^m \le \beta_i^m + T_{ij}^{\text{travel}} + M^{+\infty}(1 - y_{ij}^m) \qquad \forall i,j \in N_m, \forall m \in M \tag{4.9}$$
$$\alpha_i^m \ge T_i^{\text{entrance}} - M^{+\infty}(k_i^m - 1) \qquad \forall i \in N_m, \forall m \in M \tag{4.10}$$
$$\alpha_i^m - M^{+\infty}(k_i^m - 1) \le T_i^{\text{entrance}} \qquad \forall i \in N_m, \forall m \in M \tag{4.11}$$
$$1 - k_i^m \ge M^{+\infty}(\mu_i^m - 1) \qquad \forall i \in N_m, \forall m \in M \tag{4.12}$$
$$k_i^m - |N_m| \ge M^{+\infty}(\sigma_i^m - 1) \qquad \forall i \in N_m, \forall m \in M \tag{4.13}$$
$$1 - \sum_{j\in N_m} y_{ji}^m = \mu_i^m \qquad \forall i,j \in N_m, \forall m \in M \tag{4.14}$$
$$1 - \sum_{j\in N_m} y_{ij}^m = \sigma_i^m \qquad \forall i,j \in N_m, \forall m \in M \tag{4.15}$$
$$b_i^m \le C_m^{\text{capacity}} \qquad \forall i \in N_m, \forall m \in M \tag{4.16}$$
$$b_i^m = (C_m^{\text{original}} - u_i^m + l_i^m + \sum_{j\in N_m} v_{ji}^m)\mu_i^m + (\sum_{j\in N_m} b_j^m y_{ji}^m - u_i^m + l_i^m + \sum_{j\in N_m} v_{ij}^m - \sum_{j\in N_m} v_{ji}^m)(1 - \mu_i^m)$$
$$\forall i \in N_m, \forall m \in M. \tag{4.17}$$

Constraints (4.2) and (4.3) ensure that the number of optional inter-terminal containers transported directly from terminal i to j is less than the total number of all available inter-terminal containers transported from i to j. Constraint (4.4) defines per vessel the departure time from terminal i, using arrival time, waiting time and service time. Constraint (4.5)

ensures that there will be no round-trip between two terminals. The big-M method is used to linearize Constraints (4.6)–(4.13). Constraints (4.6) and (4.7) ensure the consecutiveness of the rotations. Constraints (4.8) and (4.9) ensure that if vessel m travels from terminal i to j, then the arrival time at j equals the departure time from terminal i, plus the traveling time T_{ij}^{travel}. Constraints (4.10) and (4.11) ensure that if vessel m visits terminal i first, that the arrival time at terminal i equals $T\text{entrance}_i$.

Constraints (4.12) and (4.13) determine the terminals that will be visited in the first and in the last. If terminal i will be visited the first, then $k_i^m = 1$, and auxiliary variable $\mu_i^m = 1$; otherwise $\mu_i^m = 0$. If terminal i will be visited the last, then $k_i^m = |N_m|$, and auxiliary variable $\sigma_i^m = 1$; otherwise $\sigma_i^m = 0$. Constraints (4.14) and (4.15) are used to ensure that all the terminals will be visited. If terminal i will be visited as the first terminal, then vessel m cannot travel from any other terminal to i. This implies that if $\mu_i^m = 1$, then $\sum_{j \in N_m} y_{ji}^m = 0$; otherwise, if $\mu_i^m = 0$, then $\sum_{j \in N_m} y_{ji}^m = 1$. Similarly, if terminal i will be visited as the last terminal, then vessel m cannot travel from terminal i to any other terminal. This implies that if $\sigma_i^m = 1$, then $\sum_{j \in N_m} y_{ij}^m = 0$; otherwise, if $\sigma_i^m = 0$, then $\sum_{j \in N_m} y_{ij}^m = 1$.

Constraints (4.16) and (4.17) ensure that the number of containers on board of a vessel will not exceed the capacity of the inland vessel at any segment of the trip. Constraint (4.17) implies that: if vessel m visits terminal i first, then the number of containers on-board equals the initial number of containers minus the already unloaded containers at terminal i, plus the hinterland containers and ITT containers already loaded at terminal i; otherwise, the number of containers on-board equals the number of containers on-board at the previous terminal, plus the number of inter-terminal and hinterland containers that need to be loaded at terminal i, minus the number of inter-terminal and hinterland containers that need to be unloaded at terminal i.

Problem (4.2)-(4.17) is a mixed integer programming (MIP) problem for which solvers are commercially available. An optimal solution provides a rotation plan that minimizes the objective function. Besides, a number of suboptimal, but locally feasible solutions found by the solver (typically rotations with a longer time and/or less containers) are also kept, and in this way construct a so-called solution pool. Each vessel agent owns different rotation plans, and for each rotation plan, there is a corresponding round-trip time and number of not transported ITT containers obtained from the objective value. To ensure that the vessels can transport as many containers as possible and finish the transport tasks in a shorter time, candidate rotation plans with the same largest number of containers transported, but with different round-trip times are kept in the solution pools. The rotation plans and the corresponding round-trip time and transported containers will be used as input for the multiple vessel coordination.

4.3 Multiple vessel coordination

Due to the fact that the best rotation plans obtained from the single vessel optimization of a vessel may conflict with the rotation plans of other vessels, the second phase coordinates the arrival and departure times of different vessels, so as to reduce conflicts and minimize the sum of total time of stay of all vessels in the port area.

Firstly, the second phase uses the solution pools from the single vessel optimization to build a so-called solution network, see Figure 4.2. The arrows in the figure represents the

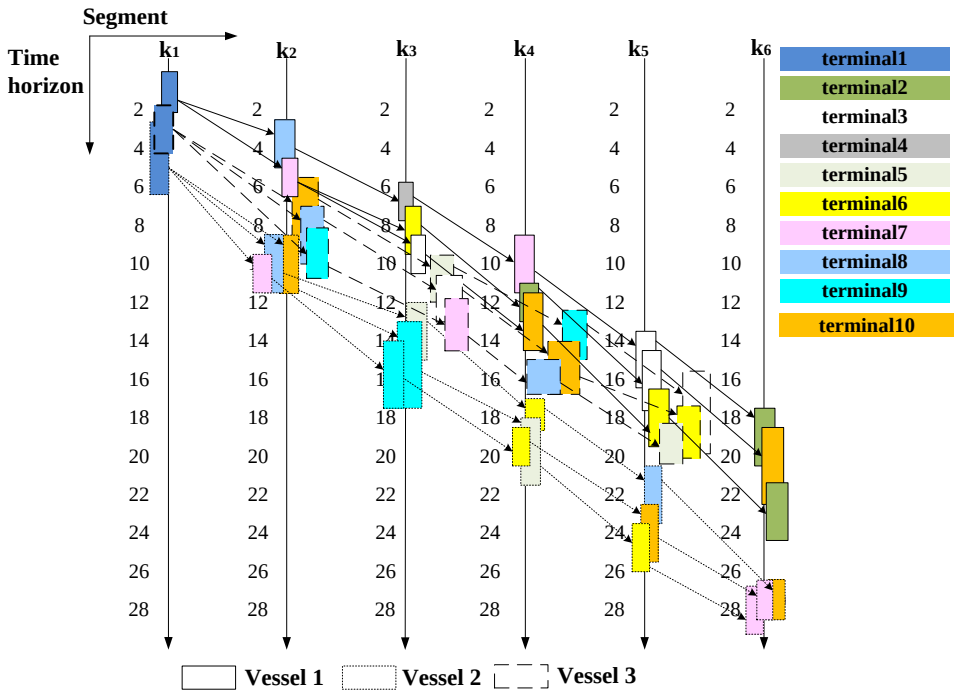

Figure 4.2: Sequence-based solution network.

order in which a vessel visits different terminals. A block in the figure represents a arrival/departure time window of a vessel at a terminal. Different colors represent different terminals. The blocks that are connected with the same line indicate that these blocks together constitute a complete rotation plan. As can be seen, some of the rotation plans are conflicting with each other (as the time window blocks at some terminals are overlapping), which means that some of the vessels have to wait to be serviced at certain terminals due to limited terminal quay capacity.

Each vessel agent has a group of candidate rotation plans in its solution pool, resulting from the first phase. The criteria for choosing candidate plans are set in different ways, depending on the objective of a particular vessel. Here, this chapter assumes that each vessel agent chooses P candidate plans with the same number of containers transported, but with different round-trip times.

To solve the multiple vessel coordination, this chapter proposed three types of methods, M2.1, M2.2 and M2.3. Method M2.1 is based on coordination rules, while both methods M2.2 and M2.2 are formulated based on CP formulations, as introduced next.

4.3.1 M2.1: Coordination rules

The coordination rules operate on the segments of the rotation plans. In each segment, the preference of each vessel agent in choosing candidate rotation plans will be updated based on the coordination rules. Initially, each vessel agent has a set of candidate rotation plans,

Algorithm 5 Basic steps of preference-based coordination rule

Require: vessel agents exchange information about the possible terminal to visit and possible arrival/departure times on each segment of their rotation plans
1: **while** $1 \leq$ segment $\leq \max_{m \in M} |N_m|$, for each segment **do**
2: **for** each vessel agent **do** check for each rotation plan in its preference-profile:
3: **if** the terminal it visits on this segment is not same as any other vessel visits on this segment: the vessel agent keeps the ranking order of this rotation plan in the preference profile;
4: **otherwise**: the vessel agent checks the arrival/departure time window of this rotation plan:
5: **if** this time window is not overlapping with the arrival/departure time window of any other vessel that visits the same terminal: the vessel agent keeps the original ranking of the rotation plan in the preference profile;
6: **otherwise**: the vessel agent compares the arrival time on this rotation plan with the arrival times of other vessel agents: whenever the arrival time of this vessel agent is later than another vessel, it puts the ranking of this rotation plan one position lower and updates preference profile, while the other vessel agent with earlier arrival time keeps its original preference profile;
7: Go to the next segment.
8: After the coordination has finished at the last segment
9: **return** the rotation plan with the highest ranking as the best option to each vessel agent.

resulting from the first phase. Then the vessel agent ranks these candidate plans based on the round-trip time. The shorter the corresponding round-trip time is, the higher the ranking of the candidate plan will be. After that, each agent has a preference profile indicating the priorities of different candidate plans, and their corresponding utility values. The priorities of rotation plans are represented as an order of rankings. For example, if a vessel agent has n possible rotation plans in the solution pool, then the highest ranking, which is the best option for the agent is 1, while the least-preferred solution is n. The utility value is represented by the round-trip time of the rotation plan. Thus, the preference profile of a vessel agent consists of rankings and utility values of several candidate rotation plans.

Preference-based coordination

The coordination rule that is based on the rankings of solutions is referred as preference-based coordination. Assume there are multiple vessels entering the port area, the preference-based coordination rule is described in Algorithm 5.

In each segment, vessels exchange information about the possible terminal and the possible arrival/departure time at this terminal on this segment of their rotation plans (line 1). Then each vessel agent updates its preference profile accordingly, based on the information it receives from other vessel agents (lines 2–6). After final coordination has finished at the last segment $|N_m|$, the preference profiles of all considered vessel agents have been updated with all segments on the rotation plans. Based on these preference profiles, each vessel agent will choose the rotation plan with the highest ranking as the best option (lines 8–9). The reason is that this plan has the least conflicting arrival/departure time windows with other vessel agents.

Algorithm 6 Basic steps of utility-based coordination rule

Require: vessel agents exchange information about the possible terminal to visit and possible arrival/departure times on each segment of their rotation plans

1: **while** $1 \leq$ segment $\leq \max_{m \in M} |N_m|$, for each segment **do**
2: **for** each vessel agent **do** check for each rotation plan in its preference-profile:
3: **if** the terminal it visits on this segment is not same as any other vessel visits on this segment: keeps the original utility value in the preference profile;
4: **otherwise**: the vessel agent checks the arrival/departure time window of this rotation plan:
5: **if** this time window is not overlapping with the arrival/departure time window of any other vessel that visits the same terminal: the vessel agent keeps the original ranking of the rotation plan in the preference profile;
6: **otherwise**: the vessel agent compares the arrival time on this rotation plan with the arrival times of other vessel agents: if the arrival time of this vessel agent is later than several vessels, it adds the maximum possible waiting time caused by these vessels to the utility value of this rotation plan and updates preference profile, while the other vessel agents with earlier arrival time keep their original preference profiles;
7: Go to the next segment.
8: After the coordination has finished at the last segment
9: **return** the rotation plan with the minimum utility value as the best option to each vessel agent.

Utility-based coordination

Utility-based problem solving is based on a similar procedure as the preference-based coordination, the basic steps is shown in Algorithm 6. The difference is that this coordination rule updates the utility value of the candidate rotation plans in the coordination process instead of ranking orders in the preference profiles.

In the each segment, vessel agents exchange information about the possible terminal ID and possible arrival/departure time on this segment of their rotation plans. Then each vessel agent updates its preference profile accordingly, based on the information it receives from other vessel agents (lines 2–6). In line 6, when overlap of arrival/departure time window happens for two vessel agents, the vessel agent with a later arrival time has to wait until the vessel agent with an earlier arrival time leaves the terminal. This waiting time is therefore calculated as the difference of the arrival time of the later arrived vessel and the departure time of the earlier arrived vessel. As a matter of fact, usually there will be more than two vessels with overlapping time windows. The maximum waiting time caused by the other vessels is thereby used and defined as maximal possible waiting time. This maximal possible waiting time will be added to the corresponding utility value of the rotation plan in the preference profile.

After the final coordination has finished at the last segment $|N_m|$, the preference profiles of all considered vessel agents have been updated with all segments on the rotation plans (lines 8-9). Based on these preference profiles, each vessel agent will choose the rotation plan with the minimum utility value as the best option (line 9). This utility value obtained after coordination is the sum of the original round-trip time of the corresponding rotation plan, and the possible waiting time caused by the conflicts with other vessel agents. As a result, the plan with minimum utility value has the shortest possible round-trip time, and is thus considered as the best option.

Table 4.4: Parameters for the multiple vessel coordination problem.

Symbols	Definitions
Q_i	set of quays in terminal i
P_m	set of candidate rotation plans in the solution pool of vessel m
$K_m^{\max}$	number of segments in the rotation plan of vessel m
A_{kp}^m	arrival time of vessel m on segment k for its rotation plan p
D_{kp}^m	departure time of vessel m on segment k for its rotation plan p
T_{kp}^m	the terminal that is on segment k of vessel m's rotation plan p
$T_{jq}^{\text{departure}}$	latest departure time of the vessels being served at quay q of terminal j
$[W_{\text{start}}^j, W_{\text{end}}^j]$	closing time window of terminal j
$[S_q^j, E_q^j]$	service time window of the sea vessel at quay q of terminal j

Table 4.5: Decision variables for the multiple vessel coordination problem.

Symbols	Definitions
I_m	rotation plan that vessel m chooses
a_{mk}	arrival time of vessel m on segment k after coordination
d_{mk}	departure time of vessel m on segment k after coordination
w_{mk}	waiting time of vessel m on segment k after coordination
s_{mk}	service time of vessel m on segment k after coordination
r_{mk}	ranking of the arrival time of vessel m on segment k after coordination
t_{mk}	the terminal vessel m visits on segment k after coordination

4.3.2 M2.2 and M2.3: CP-based methods

Mathematical description

Besides the coordination rules, methods M2.2 and M2.3 use the CP formalism to formulate the multiple vessel coordination problem based on the solution network, with the aim of minimizing the total time of stay of all vessels in the port area. The objective is defined as follows:

$$\min \sum_{m=1}^{M} d_{mK_m^{\max}}, \tag{4.18}$$

where $K_m^{\max}$ is the last segment on vessel m's rotation plan, and $d_{mK_m^{\max}}$ is the departure time of vessel m from the last segment on its rotation.

The parameters that will be used for formulating the multiple vessel coordination problem are shown in Table 4.4. Due to the capacity limits of terminal quay resources, a limited number of vessels can be served simultaneously at the same terminal, reflected by parameter Q_i. Parameter P_m represents the set of feasible rotation plans in vessel m's solution pool. Parameter $K_m^{\max}$ represents the number of segments in the rotation plan of vessel m. Parameters A_{kp}^m, D_{kp}^m and T_{kp}^m are obtained from the single vessel optimization. Parameter $T_{jq}^{\text{departure}}$ represents the latest departure time of vessels that being served at quay crane q of terminal j. Moreover, to consider the impact of terminal opening/closing time and the priority of sea-vessels, time window $[W_{\text{start}}^j, W_{\text{end}}^j]$ represents the closing time window of terminal j and $[S_q^j, E_q^j]$ represents the service time window of a sea vessel at quay crane q of terminal j.

The decision variables are shown in Table 4.5. Variable I_m represents the rotation plan that vessel m chooses. Variables a_{mk}, d_{mk}, s_{mk} and w_{mk} represent the arrival, departure, service and waiting time of vessel m at segment k after coordination, respectively. Variable r_{mk} represents ranking of vessel m at segment k after coordination. Variable t_{mk} represents the terminal vessel m visits on segment k after coordination.

Auxiliary variables $e_{mm'k}, g^q_{jkr}, f^q_{jkr}$ and h^q_{jk} are introduced to calculate variable r_{mk} and w_{mk}. As the vessels are served first-come-first-served at terminals, the vessels with earlier arrival time will always be served first. Therefore, the handling order of the vessels at each terminal need to be determined. Thus, auxiliary variable $e_{mm'k}$ is used to calculate the ranking of the arrival time of vessel m at segment k. Auxiliary variables g^q_{jkr}, h^q_{jk} and f^q_{jkr} are used to calculate the waiting time of vessel m at segment k. Auxiliary variable g^q_{jkr} represents the possible starting time on quay crane q for vessels that will visit terminal j on segment k with the ranking of r. Auxiliary variable f^q_{jkr} represents the departure time on quay crane q of terminal j on segment k of the vessel with rank r. Auxiliary variable h^q_{jk} represents the latest departure time on quay crane q of terminal j on segment k. For clarity, the constraints are divided into different groups and introduced as follows:

A. Rotation plans-related constraints

$$\textbf{\textit{element}}\left(I_m, t_{mk} | (T^m_{k1}, \ldots, T^m_{kP_m})\right) \qquad \forall m \in M, \forall k = \{1, 2, \ldots, K^{\max}_m\} \quad (4.19)$$

$$\textbf{\textit{element}}\left(I_m, a_{m1} | (A^m_{11}, \ldots, A^m_{1P_m})\right) \qquad \forall m \in M \quad (4.20)$$

$$\textbf{\textit{element}}\left(I_m, d_{m1} | (D^m_{11} + w_{m1}, \ldots, D^m_{1P_m} + w_{m1})\right) \qquad \forall m \in M \quad (4.21)$$

$$\textbf{\textit{element}}\left(I_m, s_{mk} | (D^m_{k1} - A^m_{k1}, \ldots, D^m_{kP_m} - A^m_{kP_m})\right) \qquad \forall m \in M, \forall k = \{1, 2, \ldots, K^{\max}_m\} \quad (4.22)$$

$$\textbf{\textit{element}}\left(I_m, a_{mk} | (d_{m,k-1} + A^m_{k2} - D^m_{k2}, \ldots, d_{m,k-1} + A^m_{kP_m} - D^m_{kP_m})\right) \qquad \forall m \in M, \forall k = \{2, \ldots, K^{\max}_m\} \quad (4.23)$$

$$d_{mk} = a_{mk} + s_{mk} + w_{mk} \qquad \forall m \in M, \forall k = \{2, \ldots, K^{\max}_m\} \quad (4.24)$$

Constraints (4.19)–(4.23) incorporate a special global constraint, which is defined in [269] in the following generic way:

$$\textbf{\textit{element}}\,(y, z | (a_1, \ldots, a_m))$$

where, y and z are variables, and $(a_1, \ldots, a_m)$ is a set of values. It ensures that the variable z takes the y-th value in the tuple $(a_1, \ldots, a_m)$. Consequently, Constraint (4.19) ensures that if vessel m chooses plan p, that the terminal it visits during segment k is T^m_{kp}. Constraints (4.20) and (4.21) ensure that if vessel m chooses plan p at the first segment of its rotation, the arrival time is A^m_{1p}, and the departure time equals to the original departure time plus the waiting time. Constraint (4.22) ensures that the service time at the different segments is respected. Constraint (4.23) ensures that in the successive segments ($k > 1$), the arrival time at segment k equals the departure time from the previous segment and the traveling time ($A^m_{kp} - D^m_{k-1,p}$) according to the plan p the vessel chooses. In Constraint (4.24), the departure time from segment k equals the sum of arrival time, service time and the waiting time at segment k.

B. Quay crane-related constraints

$$t_{mk} = t_{m'k} \rightarrow a_{mk}(1 - e_{mm'k}) < a_{m'k} \qquad \forall m, m' \in M, \forall m' \neq m, \forall k = \{1, \ldots, K_m^{\max}\} \quad (4.25)$$

$$t_{mk} = t_{m'k} \rightarrow a_{mk} \geq a_{m'k} e_{mm'k} \qquad \forall m, m' \in M, \forall m' \neq m, \forall k = \{1, \ldots, K_m^{\max}\} \quad (4.26)$$

$$r_{mk} = \sum_{m'}^{M} e_{mm'k} \qquad \forall m, m' \in M, \forall m' \neq m, \forall k = \{1, \ldots, K_m^{\max}\} \quad (4.27)$$

$$g_{j11}^{q} = T_{jq}^{\text{departure}} \qquad \forall q \in Q_j, \forall j \in N_m \quad (4.28)$$

$$g_{jk'1}^{q} = h_{j,k'-1}^{q} \qquad \forall q \in Q_j, \forall j \in N_m, \forall k' = \{2, 3, \ldots, K_m^{\max}\} \quad (4.29)$$

$$h_{jk}^{q} = f_{jkr_j^{\max}}^{q} \qquad \forall j \in N_m, \forall k = \{1, \ldots, K_m^{\max}\}, \forall q \in Q_j \quad (4.30)$$

$$g_{t_{m'k}kr_{mk}+1}^{q} = f_{t_{mk}kr_{mk}}^{q} \qquad \forall k = \{1, \ldots, K_m^{\max}\}, \forall q \in Q_{t_{mk}} \quad (4.31)$$

$$\left(g_{t_{mk}kr_{mk}}^{q} \neq \min(g_{t_{mk}kr_{mk}}^{1}, g_{t_{mk}kr_{mk}}^{2}, \ldots, g_{t_{mk}kr_{mk}}^{Q_{t_{mk}}})\right) \rightarrow \left(h_{t_{mk}k}^{q} = g_{t_{mk}kr_{mk}}^{q}\right) \qquad \forall m \in M, \forall k = \{1, \ldots, K_m^{\max}\}, \forall q \in Q_{t_{mk}}. \quad (4.32)$$

Constraints (4.25) and (4.26) show the comparison of the arrival time of the vessels that arrive at the same terminal on the same segment in order to calculate the ranking of vessel m on segment k in Constraint (4.27). Constraint (4.28) represents that the possible starting time for the first vessel arriving at terminal j on its first segment on quay q equals $T_{jq}^{\text{departure}}$, which is the latest departure time of the vessels already being served before the upcoming vessels. Constraint (4.29) represents that in the subsequent segments ($k' > 1$), the possible starting time for the first vessel arriving at quay q of terminal j on segment k' equals the latest departure time of the vessels arrived earlier at terminal j on their segment $k' - 1$. Constraint (4.30) ensures that the latest departure time at quay q of terminal j on segment k equals the maximum departure time of the vessels that arrive at terminal j on segment k. Constraint (4.31) represents that for the next vessel that will arrive at terminal t_{mk} on segment k with ranking $r_{mk} + 1$, the possible starting time for it will be the departure time of the vessel that arrived earlier with ranking r_{mk}. Constraint (4.32) represents that if quay q does not have the closest possible starting time, the vessel will not be served at quay q and the latest departure time will not be updated.

C. Opening/closing-related constraints

On the other hand, if $g_{t_{mk}kr_{mk}}^{q} = \min(g_{t_{mk}kr_{mk}}^{1}, g_{jkr_{mk}}^{2}, \ldots, g_{t_{mk}kr_{mk}}^{Q_{t_{mk}}})$, representing that quay q has the closest possible starting time, vessel m will be served at quay q at terminal t_{mk}. The waiting time and departure time of vessel m at terminal t_{mk} are thereby need to be identified. Constraints (4.33)–(4.38) determine the waiting time of each vessel at each terminal, considering the closing time of the terminal:

$$\left(\max(a_{mk}, g_{t_{mk}kr_{mk}}^{q}) \leq W_{\text{start}}^{t_{mk}}\right) \wedge \left(\max(a_{mk}, g_{t_{mk}kr_{mk}}^{q}) \leq W_{\text{end}}^{t_{mk}}\right) \wedge \left(\max(a_{mk}, g_{t_{mk}kr_{mk}}^{q}) + s_{mk} \geq W_{\text{start}}^{t_{mk}}\right)$$

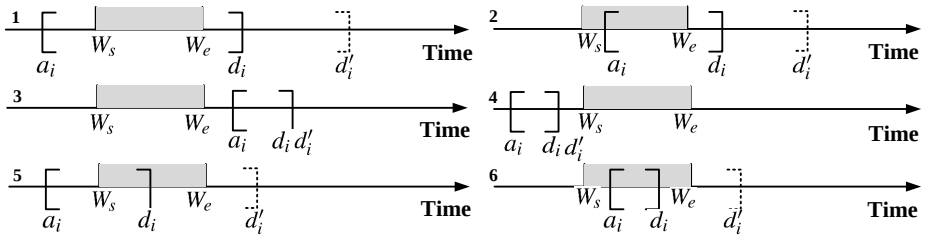

Figure 4.3: Six possible relations between a vessel's arrival, departure time window and the closing time of a terminal (adapted from [89]).

$$\rightarrow \left(w_{mk} = W_{\text{end}}^{t_{mk}} + \max(a_{mk}, g^q_{t_{mk}kr_{mk}}) - W_{\text{start}}^{t_{mk}} - a_{mk}\right) \qquad \forall m \in M, \forall k = \{1,\ldots,K_m^{\max}\}, \forall q \in Q_{t_{mk}}; \tag{4.33}$$

$$\left(\max(a_{mk}, g^q_{t_{mk}kr_{mk}}) \leq W_{\text{start}}^{t_{mk}}\right) \wedge \left(\max(a_{mk}, g^q_{t_{mk}kr_{mk}}) \leq W_{\text{end}}^{t_{mk}}\right) \wedge \left(\max(a_{mk}, g^q_{t_{mk}kr_{mk}}) + s_{mk} \geq W_{\text{start}}^{t_{mk}}\right)$$

$$\rightarrow \left(f^q_{t_{mk}kr_{mk}} = W_{\text{end}}^{t_{mk}} + s_{mk} + \max(a_{mk}, g^q_{t_{mk}kr_{mk}}) - W_{\text{start}}^{t_{mk}}\right) \qquad \forall m \in M, \forall k = \{1,\ldots,K_m^{\max}\}, \forall q \in Q_{t_{mk}}; \tag{4.34}$$

$$\left(\max(a_{mk}, g^q_{t_{mk}kr_{mk}}) > W_{\text{start}}^{t_{mk}}\right) \wedge \left(\max(a_{mk}, g^q_{t_{mk}kr_{mk}}) < W_{\text{end}}^{t_{mk}}\right) \wedge \left(\max(a_{mk}, g^q_{t_{mk}kr_{mk}}) + s_{mk} > W_{\text{start}}^{t_{mk}}\right)$$

$$\rightarrow \left(w_{mk} = W_{\text{end}}^{t_{mk}} - a_{mk}\right) \qquad \forall m \in M, \forall k = \{1,\ldots,K_m^{\max}\}, \forall q \in Q_{t_{mk}}; \tag{4.35}$$

$$\left(\max(a_{mk}, g^q_{t_{mk}kr_{mk}}) > W_{\text{start}}^{t_{mk}}\right) \wedge \left(\max(a_{mk}, g^q_{t_{mk}kr_{mk}}) < W_{\text{end}}^{t_{mk}}\right) \wedge \left(\max(a_{mk}, g^q_{t_{mk}kr_{mk}}) + s_{mk} > W_{\text{start}}^{t_{mk}}\right)$$

$$\rightarrow \left(f^q_{t_{mk}kr_{mk}} = W_{\text{end}}^{t_{mk}} + s_{mk}\right) \qquad \forall m \in M, \forall k = \{1,\ldots,K_m^{\max}\}, \forall q \in Q_{t_{mk}}; \tag{4.36}$$

$$\left(\max(a_{mk}, g^q_{t_{mk}kr_{mk}}) > W_{\text{end}}^{t_{mk}}\right) \vee \left(\max(a_{mk}, g^q_{t_{mk}kr_{mk}}) + s_{mk} < W_{\text{start}}^{t_{mk}}\right)$$
$$\rightarrow \left(w_{mk} = \max(a_{mk}, g^q_{t_{mk}kr_{mk}}) - a_{mk}\right) \qquad \forall m \in M, \forall k = \{1,\ldots,K_m^{\max}\}, \forall q \in Q_{t_{mk}}; \tag{4.37}$$

$$\left(\max(a_{mk}, g^q_{t_{mk}kr_{mk}}) > W_{\text{end}}^{t_{mk}}\right) \vee \left(\max(a_{mk}, g^q_{t_{mk}kr_{mk}}) + s_{mk} < W_{\text{start}}^{t_{mk}}\right)$$
$$\rightarrow \left(f^q_{t_{mk}kr_{mk}} = \max(a_{mk}, g^q_{t_{mk}kr_{mk}}) + s_{mk}\right) \qquad \forall m \in M, \forall k = \{1,\ldots,K_m^{\max}\}, \forall q \in Q_{t_{mk}}; \tag{4.38}$$

In practice, some terminals will be closed in the night. Therefore, it is important to consider the terminal of terminals when calculating the waiting times of the vessels that arrive at this terminal. Figure 4.3 shows the six possible relations of a vessel's arrival, departure time window and the closing of a terminal. In situations 1 and 5, the updated

departure time $d_i' = W_e + d_i - W_s$; in situations 2 and 6, the updated departure time $d_i' = W_e + d_i - a_i$; in situations 3 and 4, the updated departure time $d_i' = d_i$. Therefore, the waiting time caused by situations 1 and 5 equals $d_i' - d_i = W_e - W_s$; the waiting time caused by the situations 2 and 6 equals $d_i' - d_i = W_e - a_i$; the waiting time caused by situations 3 and 4 equals $d_i' - d_i = 0$.

When a vessel is assigned to a quay of a terminal, it has to wait until the a possible other vessel that is being served at that quay has left. Thus, the actual start time for vessel m with arrival time a_{mk} is $\max(a_{mk}, g^q_{t_{mk}kr_{mk}})$, and the updated departure time is $\max(a_{mk}, g^q_{t_{mk}kr_{mk}}) + s_{mk}$. Then the updated arrival and departure time window $\left(\max(a_{mk}, g^q_{t_{mk}kr_{mk}}), \max(a_{mk}, g^q_{t_{mk}kr_{mk}}) + s_{mk}\right)$ is compared with the closing time of the terminal $[W^{t_{mk}}_{\text{start}}, W^{t_{mk}}_{\text{end}}]$.

The waiting time of the vessel, as well as and the updated departure time from the quay (when the vessel has left) caused by situations 1 and 5 in Figure 4.3 are represented by Constraints (4.33) and (4.34), respectively. Similarly, the waiting time of a vessel and the updated departure time from the quay caused by situations 2 and 6, are represented by Constraints (4.35) and (4.36). In addition, the waiting time of the vessel and the updated departure time from the quay caused by situations 3 and 4, are represented by Constraints (4.37) and (4.38).

D. Sea vessel-related constraints

$$\left(\max(a_{mk}, g^q_{t_{mk}kr_{mk}}) \le S^{t_{mk}}_q\right) \wedge \left(\max(a_{mk}, g^q_{t_{mk}kr_{mk}}) \le E^{t_{mk}}_q\right) \wedge \left(\max(a_{mk}, g^q_{t_{mk}kr_{mk}}) + s_{mk} \ge S^{t_{mk}}_q\right)$$
$$\rightarrow \left(w_{mk} = E^{t_{mk}}_q + \max(a_{mk}, g^q_{t_{mk}kr_{mk}}) - S^{t_{mk}}_q - a_{mk}\right)$$
$$\forall m \in M, \forall k = \{1, \ldots, K^{\max}_m\}, \forall q \in Q_{t_{mk}}; \tag{4.39}$$

$$\left(\max(a_{mk}, g^q_{t_{mk}kr_{mk}}) \le S^{t_{mk}}_q\right) \wedge \left(\max(a_{mk}, g^q_{t_{mk}kr_{mk}}) \le E^{t_{mk}}_q\right) \wedge \left(\max(a_{mk}, g^q_{t_{mk}kr_{mk}}) + s_{mk} \ge S^{t_{mk}}_q\right)$$
$$\rightarrow \left(f^q_{t_{mk}kr_{mk}} = E^{t_{mk}}_q + s_{mk} + \max(a_{mk}, g^q_{t_{mk}kr_{mk}}) - S^{t_{mk}}_q\right)$$
$$\forall m \in M, \forall k = \{1, \ldots, K^{\max}_m\}, \forall q \in Q_{t_{mk}}; \tag{4.40}$$

$$\left(\max(a_{mk}, g^q_{t_{mk}kr_{mk}}) > S^{t_{mk}}_q\right) \wedge \left(\max(a_{mk}, g^q_{t_{mk}kr_{mk}}) < E^{t_{mk}}_q\right) \wedge \left(\max(a_{mk}, g^q_{t_{mk}kr_{mk}}) + s_{mk} > S^{t_{mk}}_q\right)$$
$$\rightarrow \left(w_{mk} = E^{t_{mk}}_q - a_{mk}\right) \qquad \forall m \in M, \forall k = \{1, \ldots, K^{\max}_m\}, \forall q \in Q_{t_{mk}}; \tag{4.41}$$

$$\left(\max(a_{mk}, g^q_{t_{mk}kr_{mk}}) > S^{t_{mk}}_q\right) \wedge \left(\max(a_{mk}, g^q_{t_{mk}kr_{mk}}) < E^{t_{mk}}_q\right) \wedge \left(\max(a_{mk}, g^q_{t_{mk}kr_{mk}}) + s_{mk} > S^{t_{mk}}_q\right)$$

$$\rightarrow \left(f^{q}_{t_{mk}kr_{mk}} = E^{t_{mk}}_{q} + s_{mk}\right) \qquad \forall m \in M, \forall k = \{1,\ldots,K^{\max}_{m}\}, \forall q \in Q_{t_{mk}}; \quad (4.42)$$

$$\left(\max(a_{mk}, g^{q}_{t_{mk}kr_{mk}}) > E^{t_{mk}}_{q}\right) \vee \left(\max(a_{mk}, g^{q}_{t_{mk}kr_{mk}}) + s_{mk} < S^{t_{mk}}_{q}\right)$$
$$\rightarrow \left(w_{mk} = \max(a_{mk}, g^{q}_{t_{mk}kr_{mk}}) - a_{mk}\right) \qquad \forall m \in M, \forall k = \{1,\ldots,K^{\max}_{m}\}, \forall q \in Q_{t_{mk}}; \quad (4.43)$$

$$\left(\max(a_{mk}, g^{q}_{t_{mk}kr_{mk}}) > E^{t_{mk}}_{q}\right) \vee \left(\max(a_{mk}, g^{q}_{t_{mk}kr_{mk}}) + s_{mk} < S^{t_{mk}}_{q}\right)$$
$$\rightarrow \left(f^{q}_{t_{mk}kr_{mk}} = \max(a_{mk}, g^{q}_{t_{mk}kr_{mk}}) + s_{mk}\right) \quad \forall m \in M, \forall k = \{1,\ldots,K^{\max}_{m}\}, \forall q \in Q_{t_{mk}}. \quad (4.44)$$

Constraints (4.39)–(4.44) consider the sea-going vessels, which always have priority over inland vessels. When a sea-going vessel is being handled by one or multiple quay cranes at a terminal, these quay cranes will be unavailable for the inland vessels. In addition, even if an inland vessel arrives earlier than a sea-going vessel, during the time window that the sea vessel will be handled, the quay cranes that the sea-going vessel uses will still be unavailable for inland vessels. This is similar to the situation of opening/closing time of a terminal, in which several quay cranes will be unavailable for a period of time. Therefore, we use similar constraints as the opening/closing time-related Constraints (4.33)–(4.38) to consider the priority of sea-going vessels. The calculation of the waiting time caused by sea-going vessels is also similar to the calculation of the waiting time caused by the closing of terminals. The difference is that the time windows of the sea-going vessels are more specific on which quay the sea-going vessels will be assigned. The start and end time window $[S^{j}_{q}, E^{j}_{q}]$ represents the estimated start and end time of the sea-going vessel at quay q of terminal q.

The solutions obtained by solving the multiple vessel coordination problem include the rotation plan each vessel chooses, and the corresponding arrival and departure time at each terminal. If all vessels implement the solutions obtained, the total time they spent in the port area will be minimized.

Solution method

In method M2.2, problem (4.19)-(4.44) is solved using a commercially constraint programming solver. However, a commercial CP solver like CPLEX 12.6.1 CP solver is able to optimally solve small-sized problem instances, for larger-sized problems, the solver is not able to solve to problem within a reasonable amount of time. Therefore, in method M2.3, problem (4.19)-(4.44) is solved using three large neighborhood search (LNS)-based heuristics.

The LNS meta-heuristic, introduced by [278] considers a neighborhood defined implicitly by a destroy and a repair method. A destroy method destructs part of a current solution while a repair method rebuilds the destroyed solution. The destroy method typically contains an element of stochasticity such that different parts of the solution are destroyed in every invocation of the method [243]. An LNS heuristic consists of the following steps: (1) initialization with the current global best solution; (2) first application of the destroy method and then the repair method to obtain a new solution; (3) evaluation of the new solution, and determination of whether this solution should become the new current solution or whether

Algorithm 7 Basic steps of LNS-based heuristic

1: intialize solution s_{best}
(a): generate initial solution from single vessel optimization;
(b): save the objective value of best solution so far;
2: **while** termination criteria not reached **do**
(c): remove q vessels from the current solution s;
(d): keep the value of the rest of $M-q$ vessels;
(e): reinsert q vessels to solution s with CPLEX CP solver, get new solution s';
(f): **if** $\text{obj}(s') \leq \text{obj}(s_{\text{best}})$ **then** $s_{\text{best}} = s$, save the $\text{obj}(s)$
3: return s_{best} and $\text{obj}(s_{\text{best}})$

it should be rejected; (4) updating the new current global best solution if needed; (5) checking if the termination condition to end the search process is satisfied. More details of LNS heuristics can be found in [243, 267].

The algorithmic outline of the proposed method M2.3 is shown in Algorithm 7. Assume that there are M vessels that need to be coordinated, and the Phase 1 single vessel optimization has finished for each vessel. Each vessel agent chooses P rotation plans are chosen from the solution pool with relatively shorter round-trip time.

An illustration of our approach is shown in Figure 4.4. These steps are carried out by the central coordinator. The different number in the circle represents different terminal IDs, and the arrows represent the visiting sequences of each vessel to different terminals. Assume that 5 vessels are entering the port, and the central coordinator use the solution from the single vessel optimization of each vessel as the initial solution. Then the coordinator remove the solution of 2 vessels using the designed removal strategies. After removal, CPLEX CP solver is used to solve the problem (4.19)–(4.44), in which the solutions of the rest 3 vessels are kept and generate new solutions for the removed 2 vessels. At this point, a new solution has been found. The objective value of the new solution is subsequently compared with the current best solution found so far. If it is better than the current best solution, this new objective value is saved as the best solution. Otherwise, this new solution will be left out and go through the removal and reinsert procedure again until the termination criteria is met. The detailed procedure of the LNS-based approach is introduced as follow.

Each vessel agent chooses the rotation plan with the minimum utility value as the initial solution for the LNS-based approach in Algorithm 7. The corresponding objective value of implementing these initial solutions in the CP-based multiple vessel coordination formulation will be saved as the current best objective value. The initial solution will also be set as the current best solution s_{best}.

Based on the initial set of rotation plans for the vessels, the coordinator then starts to modify these rotation plans by removing and reinserting the vessels iteratively. In Algorithm 7, q vessels are first removed, and the solutions of the remaining $M-q$ vessels are kept. This chapter considers three removal heuristics to decide on which q vessels will be removed from the current solution (Algorithm 7(c)). The removal heuristics are described as follows:

- **Random removal.** The random removal simply selects q vessels at random and removes them from the solution.

- **Shaw removal.** The Shaw removal heuristic was proposed by [278]. This chapter

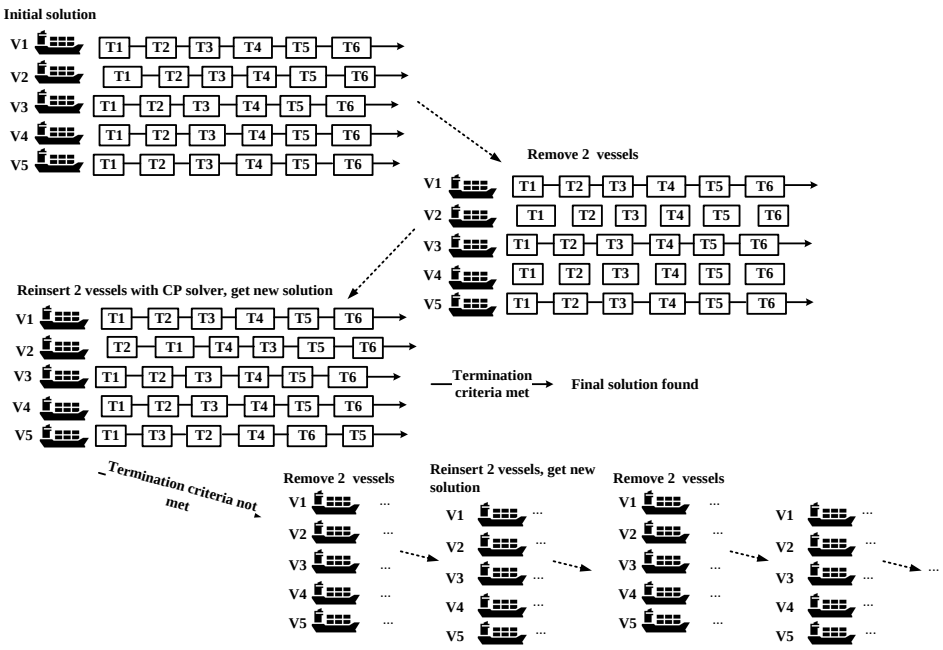

Figure 4.4: Illustration of the LNS-based heuristic.

modified it to suit the multiple vessel coordination problem. The general idea is to remove vessels that have a certain degree of similarity in rotation plans, as it is expected to be reasonably easy to remove similar vessels and thereby create new, possibly better solutions. If the coordinator chooses to remove vessels that have very different rotation plans, then the coordinator might not gain anything when reinserting vessels because it might only be able to insert vessels at their original positions or in some bad positions. This chapter defines the similarity of the rotation plans between two vessels as relativeness. The relativeness of two vessels, m_1 and m_2 is defined as follows:

$$R^{\mathrm{e}}(m_1,m_2) = \frac{\sum_{k=1}^{K_{m_1}} \sum_{p_1=1}^{P_{m_1}} \sum_{p_2=1}^{P_{m_2}} y_{kp_1p_2}^{m_1}}{K_{m_1} P_{m_1}}$$

in which $y_{kp_1p_2}^{m_1} = 1$ if $\left(T_{kp_1}^{m_1} = T_{kp_2}^{m_2}\right) \wedge \left(A_{kp_1}^{m_1} > A_{kp_2}^{m_2}\right) \wedge \left(A_{kp_1}^{m_1} < D_{kp_2}^{m_2}\right)$ (when there exist overlapping arrival/departure time windows between m_1 and m_2), otherwise $y_{kp_1p_2}^{m_1} = 0$. The procedure initially chooses a random vessel to remove, and in the subsequent iterations it chooses vessels that are similar to the already removed vessels. The Shaw heuristic selects vessels with higher relativeness $R^e(m_1,m_2)$ values.

- **Worst removal.** Given a vessel m served by terminals in a solution s, the cost of the vessel is defined as $J(m,s) = f(s) - f_{-m}(s)$, in which $f_{-m}(s)$ is the objective value in which the value of vessel m has been removed. It is reasonable to remove requests with hight costs and reinsert them at another place in the solution to obtain a better solution. Therefore, the worst removal heuristic removes vessels with high $J(m,s)$.

In other words, this removal heuristic selects the requests that appear to be placed in the wrong positions in the solution.

After the removal, CPLEX CP solver is used to solve problem (4.19)–(4.44), in which the values of the $M-q$ vessels are kept. The solver generates solutions for the already removed q vessels. At this point, a new solution s' has been found. This new solution is subsequently compared with the current best solution s_{best} in terms of the objective function value. If the objective value of the new solution is smaller than the current best objective value, the new solution is better than the current best solution, since it generates a shorter total round-trip time. Subsequently, the new solution is set as the current best solution, and the new objective value is saved as the current best objective value. Then the removal and reinsertion process is started over again. If the objective value of the new solution is larger than the current best objective value, the new solution is worse than the current best solution, since it generates a longer total round-trip time. Under this circumstance, this new solution is left out and the removal and reinsertion iteration is carried out again by removing another randomly chosen q vessels and reinserting them with the CPLEX CP solver. The whole procedure stops after a predefined running time limit.

4.4 Simulation experiments

Simulation experiments are carried out to assess and analyze the effectiveness of the proposed methods. This section first describes the experimental settings and then presents an example of feasible rotation plans generated by solving the single vessel optimization problem. To evaluate the performance of the proposed methods for solving the multiple vessel coordination problem, this section first presents an analysis of the quality of solutions of one case for example, with different running time limits. Then the logistical performances are evaluated using different performance indicators. This chapter uses the situation in which there is no coordination among vessels, as a benchmark scenario. In this situation, after the single vessel optimization for individual vessels, the corresponding vessel agents simply choose the solution (rotation plan) with the shortest round-trip time.

4.4.1 Experimental settings

Benchmark Layouts 2.1, 2.2 and 2.3, and KPI 2 (quality of solutions), KPI 7 (total round-trip time), KPI 9 (total waiting time), KPI 11 (latest departure time) that were introduced in Chapter 2 are used to evaluate methods M2.1, M2.2 and M2.3. For solving the MIP problem CPLEX 12.6 MIP solver is used. The coordination rules are implemented with MATLAB 2015b. As the CP solver the CPLEX 12.6 CP solver is used. The LNS-based heuristics are implemented in C++. For the LNS-based heuristics, the central coordinator removes and inserts 30% vessels in each iteration. The time limit for running the CP solver in the LNS-based heuristic is set as 20 seconds in each iteration. In addition, the results from the CP solver and the LNS-based heuristic are obtained with a running time of 600 seconds. For the ITT containers, it is assumed that there will always be sufficient number of containers to be transported by each inland vessel.

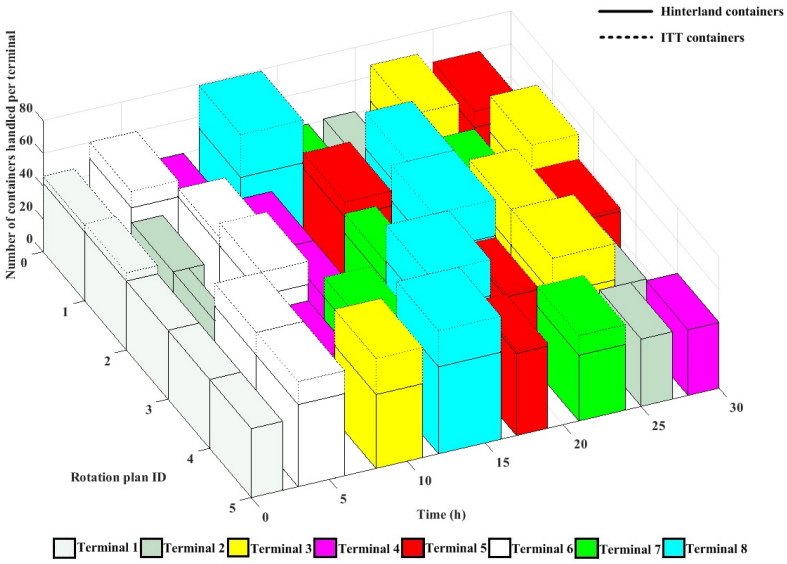

Figure 4.5: An instance of possible rotation plans for an inland vessel (capacity: 180 TEU).

4.4.2 Feasible rotation plans from single vessel optimization

In order to construct solution pools, during the single vessel optimization for each vessel, not only the optimal solutions are kept but also feasible solutions. As an example of the solutions generated from the single vessel optimization, Figure 4.5 illustrates 5 randomly chosen possible rotation plans for a typical inland vessel generated from the single vessel optimization. Different colors represent different terminals. Each block represents an arrival/departure time window at the terminal. The length of a block indicates the time the vessel stays at a terminal. The height of a block indicates the number of containers handled (both loaded and unloaded) at a terminal. The extra ITT containers are marked with dots. This could give the vessel operator an indication about the sequence and time at which to enter and leave terminals, as well as the number of containers to load and unload. In addition, as the number of extra ITT containers is also given, the vessel operator can decide by himself which rotation plans is more preferable. It would be the vessel operator's decision whether to choose the rotation plan with the shortest round-trip time or the one with the maximum number of extra ITT containers transported. In our experiments, the central coordinator chooses the rotation plans with the same amount of ITT containers transported, but with different round-trip times. Among those rotation plans in Figure 4.5, the plan with ID 4 gives the shortest round-trip time, which means it is the option in which the transportation tasks can be done most efficiently, if no other vessel is present. However, if there are other vessels in the port, this option might not be optimal. The other vessels may arrive earlier or at the same time at the same terminal, which causes waitings and delays in the previously chosen plan of this vessel. Thus, coordination among the inland vessels is needed.

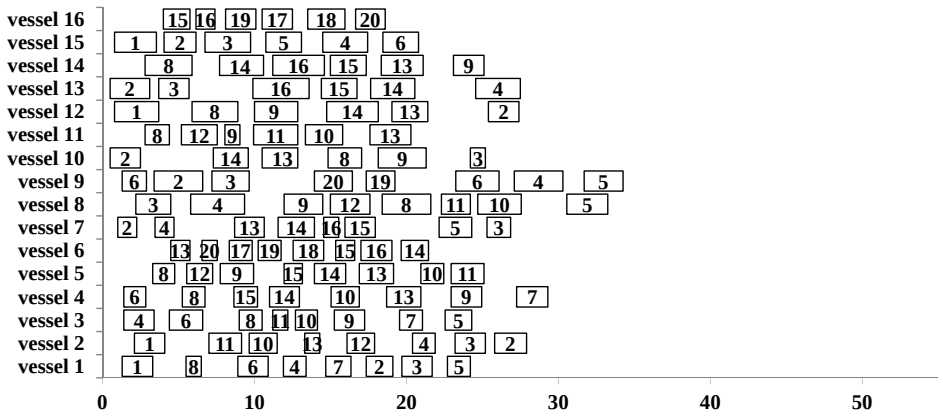

Figure 4.6: Rotation plans generated without coordination.

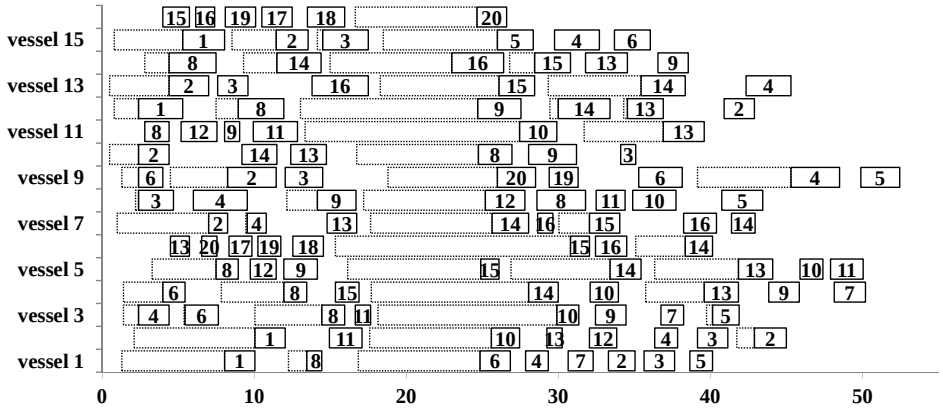

Figure 4.7: Expected implementations of rotation plans generated without coordination.

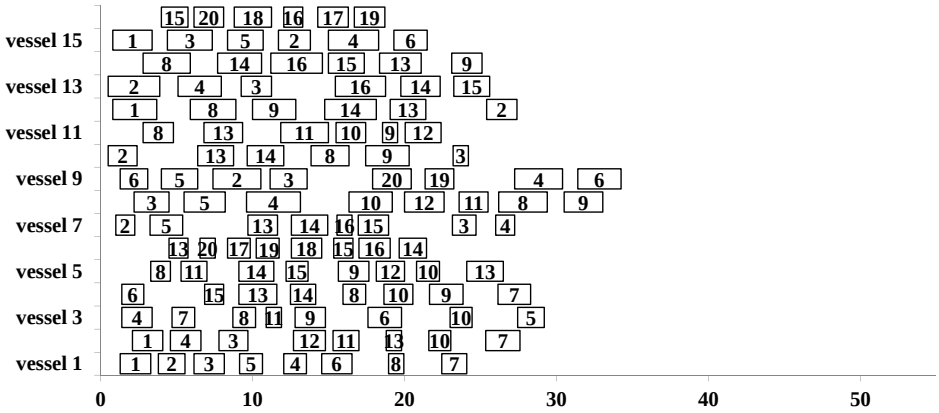

Figure 4.8: Rotation plans generated with multiple vessel coordination.

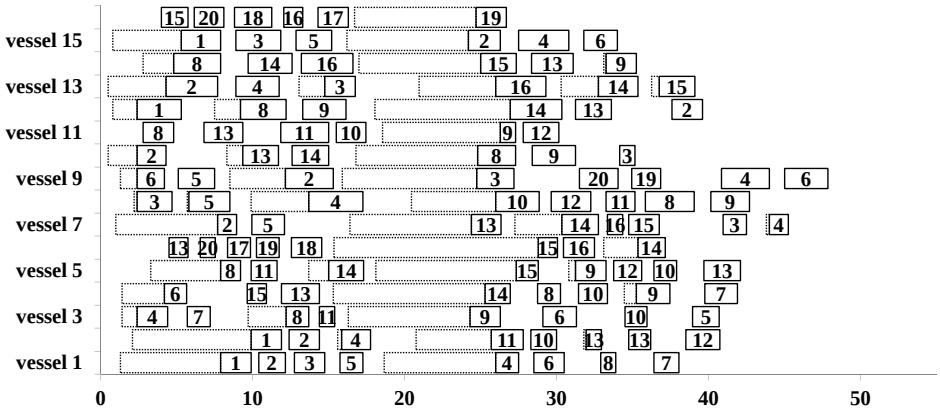

Figure 4.9: Expected implementations of rotation plans generated with multiple vessel coordination.

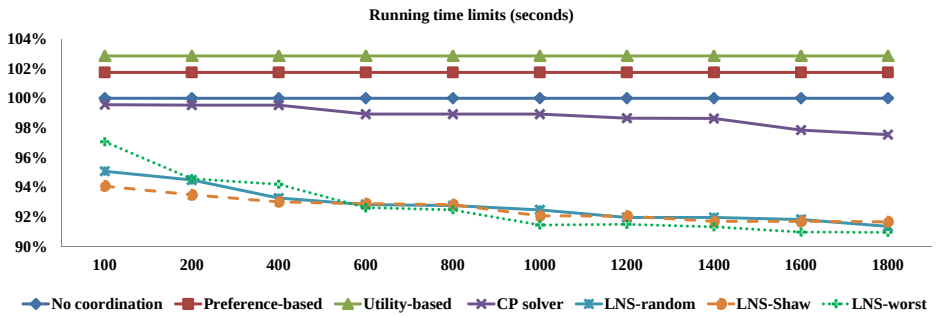

Figure 4.10: Comparison of quality of solutions (KPI 2) with varying running time limits.

4.4.3 With and without multiple vessel coordination

Figures 4.6, 4.7, 4.8 and 4.9 show the rotation plans generated before and after the multiple vessel coordination. This section presents the results from one of the experiments with Benchmark Layout 2.2 using the LNS-random heuristic in method M2.3, as an example to visualize the rotation plans and describe the differences. The number in the block represents the terminal that each vessel has visited. The length of each block represents the length of the time of stay at each terminal. Figure 4.6 and Figure 4.8 show the generated rotation plans with and without coordination, respectively. As can be seen, though the visiting sequences are different, the round-trip time of the rotation plan for the same vessel does not differ much. However, those rotation plans do not consider the practical situations, such as the closing time of terminals, the terminal capacities, and also the priority of sea-going vessels. After taking into account the practical situations, the expected implementation of the rotation plans are presented in Figure 4.7 and Figure 4.9. These implemented rotation plans include the waiting times at different terminals, represented as the blocks in dots. It can be seen that the implemented rotation plans with coordination have shorter round-trip times, as well as shorter waiting times. This implies that with the proposed method, these inland vessels can finish their transport tasks in the port in a shorter time.

4.4.4 Analysis of solution quality

Different running time limits are imposed to compare the solution quality of coordination rules, LNS-based heuristics and a commercial CP solver. As the results from Benchmark Layouts 2.1, 2.2 and 2.3 show similar patterns, this section presents the results from one of the cases with Benchmark Layout 3.3 as an example to show the quality of solution with respect to different time limits, as shown in Figure 4.10. The values reported are in percentage, which equal the objective value from different coordination methods divided by the objective value from the benchmark. According to Figure 4.10, the quality of solution increases with an increase in available time for CP-based methods, including the commercial solver (method M2.2) and LNS-based heuristics (method M2.3). The LNS-based heuristics always perform much better than the CP solver with different running time limits. Among the three LNS-based heuristics in method M2.3, the quality of solution varies between $1\% - 3\%$. In addition, the coordination rules do not outperform the benchmark in this case.

4.4.5 Comparison of logistical performance

Tables 4.6, 4.7 and 4.8 show the maximum, minimum, and average ratio of the proposed methods to the benchmark scenario with respect to the total round-trip time[1]; the total waiting time[2]; and the port departure time[3].

Table 4.6 demonstrates that LNS-based method (method M2.3) provide improvements on the total round-trip time, total waiting time and the port departure time, while the CP solver and preference-based coordination rule only show improvements on the latest departure time. Comparing with the no coordination scenario, LNS-random heuristic of method M2.3 reduces on average 5.64% total round-trip time, 9.78% reduction on the total waiting time, and 8.6% reduction on the port departure time; LNS-Shaw heuristic of method M2.3 reduces on average 4.43% total round-trip time, 7.83% reduction on the total waiting time, and 7.38% reduction on the port departure time; LNS-worst heuristic of method M2.3 reduces on average 5.52% total round-trip time, 9.41% reduction on the total waiting time, and 10.47% reduction on the port departure time. The preference-based rules (method M2.1) and CP solver (method M2.2) show improvements only with on average 4.83% and 5.08% reduction on the port departure time, respectively.

Table 4.7 demonstrates that the two coordination rules do not show improvements comparing with the benchmark. The CP solver (method M2.2) and the three LNS-based heuristics of method M2.3 show larger improvements on the total round-trip time, total waiting time and the port departure time in Benchmark Layout 2.2 than in Benchmark Layout 2.1. Morever, these two methods M2.2 and M2.3 have more improvements on total waiting time than on the total round-trip time and on the port departure time. The CP solver shows improvements with on average 2.1% less total round-trip time, 8.28% less total waiting time, and 1.45% reduction on the port departure time, while the LNS-random heuristic of method M2.3 can reduce on average 8.03% total round-trip time, 22.59% reduction on the total waiting time, and 8.5% reduction on the port departure time comparing with benchmark. LNS-Shaw heuristic of method M2.3 can reduce on average 7.33% total round-trip time, 20.63% reduction on the total waiting time, and 7.48% reduction on the port departure time. In addition, LNS-worst heuristic of method M2.3 can reduce on average 7.46% total round-trip time, 20.7% reduction on the total waiting time, and 7.68% reduction on the port departure time.

Table 4.8 demonstrates that the two coordination rules do not outperform the benchmark. The CP solver and LNS-based methods have larger improvements on total waiting time than on the total round-trip time and on the port departure time. The CP solver shows improvements with only on average 0.38% less total round-trip time, 3.93% less total waiting time, and 1.45% reduction on the port departure time, while the LNS-random heuristic of method M2.3 can reduce on average 6.84% total round-trip time, 16.08% reduction on the total waiting time, and 8.5% reduction on the port departure time comparing with benchmark. LNS-Shaw heuristic of method M2.3 can reduce on average 5.9% total round-trip time, 14.58% reduction on the total waiting time, and 7.48% reduction on the port departure time. In addition, LNS-worst heuristic of method M2.3 can reduce on average 6.97% total

[1] ratio of total round-trip time= $\frac{\sum_{m\in M} d_{mK_m^{\max}} \text{ from different methods}}{\sum_{m\in M} d_{mK_m^{\max}} \text{ from benchmark}}$

[2] ratio of total waiting time = $\frac{\sum_{m\in M}\sum_{k\in K_m^{\max}} w_{mk} \text{ from different methods}}{\sum_{m\in M}\sum_{k\in K_m^{\max}} w_{mk} \text{ from benchmark}}$

[3] ratio of the port departure time = $\frac{\max_{m\in M} d_{mK_m^{\max}} \text{ from different methods}}{\max_{m\in M} d_{mK_m^{\max}} \text{ from benchmark}}$

Table 4.6: Logistical performance comparison of solution methods for Benchmark Layout 2.1.

	KPI 7: Total Round-trip Time			KPI 9: Total Waiting Time			KPI 11: Port Departure Time		
	Max(%)	Min(%)	Avg(%)	Max(%)	Min(%)	Avg(%)	Max(%)	Min(%)	Avg(%)
Preference-based	106.67	97.05	103.14	109.22	94.10	103.73	97.67	90.43	95.17
Utility-based	112.96	101.80	108.57	117.90	100.81	111.22	106.15	93.3	100.03
CP Solver	109.05	96.53	102.35	110.75	92.89	102.05	98.83	89.13	94.92
LNS-random	96.92	89.72	94.36	94.04	82.29	90.22	95.33	88.06	91.40
LNS-Shaw	98.03	90.12	95.57	96.41	83.40	92.17	99.35	86.33	92.62
LNS-worst	97.24	90.31	94.48	96.18	83.88	90.59	93.64	83.66	89.53

Table 4.7: Logistical performance comparison of solution methods for Benchmark Layout 2.2.

	KPI 7: Total Round-trip Time			KPI 9: Total Waiting Time			KPI 11: Port Departure Time		
	Max(%)	Min(%)	Avg(%)	Max(%)	Min(%)	Avg(%)	Max(%)	Min(%)	Avg(%)
Preference-based	105.43	99.29	102.67	116.73	95.59	104.27	108.13	94.18	101.32
Utility-based	106.10	98.99	102.21	112.80	93.68	102.54	100.39	95.39	99.97
CP Solver	98.72	96.80	97.90	94.86	88.91	91.72	113.61	84.48	98.55
LNS-random	94.15	89.99	91.97	81.94	73.01	77.41	95.88	84.48	91.50
LNS-Shaw	94.93	90.78	92.67	84.57	72.05	79.37	97.65	88.27	92.52
LNS-worst	93.80	90.88	92.54	80.73	77.01	79.30	96.94	87.11	92.32

Table 4.8: Logistical performance comparison of solution methods for Benchmark Layout 2.3

	KPI 7: Total Round-trip Time			KPI 9: Total Waiting Time			KPI 11: Port Departure Time		
	Max(%)	Min(%)	Avg(%)	Max(%)	Min(%)	Avg(%)	Max(%)	Min(%)	Avg(%)
Preference-based	105.88	99.84	103.41	111.27	93.51	103.73	101.75	98.30	100.24
Utility-based	106.71	100.43	103.97	113.80	92.22	101.94	108.54	100.00	102.84
CP Solver	102.03	96.68	99.62	106.80	86.26	96.07	113.61	84.48	98.55
LNS-random	96.03	88.08	93.16	94.17	73.15	83.92	95.88	84.48	91.50
LNS-Shaw	96.86	91.33	94.10	93.10	79.85	85.42	97.65	88.27	92.52
LNS-worst	95.91	90.35	93.03	89.93	74.63	82.74	96.94	87.11	92.32

round-trip time, 17.26% reduction on the total waiting time, and 7.68% reduction on the port departure time.

4.4.6 Results analysis

Tables 4.9, 4.10 and 4.11 conclude our analysis with an evaluation of the proposed solution methods M2.1, M2.2 and M2.3, represented as the percentage of the experiments in which the solution methods performs the best, second-best, third-best, forth-best and fifth-best with respect to the three logistical performance indicators KPIs 7, 9, and 11.

Table 4.9: Percentages of experiments in which each solution method performs 1st-best, 2nd-best, 3rd-best, 4th-best, 5th-best, 6th-best, 7th-best on total round-trip time (KPI 7)

Ranking	1	2	3	4	5	6	7
No coordination	0%	0%	0%	33%	53%	14%	0%
Utility-based (M2.1)	0%	0%	0%	0%	6%	20%	74%
Preference-based (M2.1)	0%	0%	0%	4%	16%	54%	26%
CP solver (M2.2)	0%	0%	0%	63%	23%	14%	3%
LNS-random (M2.3)	46%	40%	14%	0%	0%	0%	0%
LNS-Shaw (M2.3)	0%	30%	70%	0%	0%	0%	0%
LNS-worst (M2.3)	54%	30%	16%	0%	0%	0%	0%

Table 4.10: Percentages of experiments in which each solution method performs 1st-best, 2nd-best, 3rd-best, 4th-best, 5th-best, 6th-best, 7th-best on total waiting time (KPI 9)

Ranking	1	2	3	4	5	6	7
No coordination	0%	0%	0%	23%	47%	24%	6%
Utility-based (M2.1)	0%	0%	0%	6%	13%	21%	60%
Preference-based (M2.1)	0%	0%	0%	3%	20%	46%	31%
CP solver (M2.2)	0%	0%	0%	64%	23%	10%	3%
LNS-random (M2.3)	50%	36%	14%	0%	0%	0%	0%
LNS-Shaw (M2.3)	14%	23%	63%	0%	0%	0%	0%
LNS-worst (M2.3)	36%	43%	21%	0%	0%	0%	0%

Table 4.11: Percentages of experiments in which each solution method performs 1st-best, 2nd-best, 3rd-best, 4th-best, 5th-best, 6th-best, 7th-best on the port departure time (KPI 11)

Ranking	1	2	3	4	5	6	7
No coordination	6%	16%	0%	10%	3%	50%	15%
Utility-based (M2.1)	3%	16%	3%	3%	29%	10%	36%
Preference-based (M2.1)	6%	16%	6%	16%	30%	13%	13%
CP solver (M2.2)	3%	10%	23%	39%	16%	6%	3%
LNS-random (M2.3)	36%	36%	20%	8%	0%	0%	0%
LNS-Shaw (M2.3)	43%	13%	25%	16%	3%	0%	0%
LNS-worst (M2.3)	33%	47%	17%	3%	0%	0%	0%

Table 4.9 shows that the LNS-worst heuristic in method M2.3 performs best on KPI 7, as it performs the first in 54% of the experiments on the total round-trip time. Table 4.10

shows that LNS-random heuristic in method M2.3 performs best on KPI 9, as it ranks first in 50% of the experiments on the total waiting time. Table 4.11 shows that LNS-shaw heuristic in method M2.3 performs best on KPI 11, as it ranks first in 43% of the experiments on the port departure time.

To conclude, the LNS-based heuristics (method M2.2) show better solutions than the other methods in most of the cases. The reason for this is that in every iteration, the upper bound of the formulated multiple vessel coordination problem will be updated with a lower objective value. Thus, the solutions generated after every iteration should at least be the same or better than the solutions generated in previous iterations. Among LNS-based heuristics, LNS-worst heuristic performs best on KPI 2 and KPI 7, LNS-random heuristic performs best on KPI 9, and LNS-shaw heuristic performs best on KPI 11. Meanwhile, although coordination rules do not outperform CP solver and LNS-based heuristics on all KPIs, they perform better than the benchmark on KPI 11.

4.5 Conclusions

This chapter proposes a two-phase approach for tackling the vessel rotation planning in medium-sized ports, in which the problem in the first phase is solved using a commercial MIP solver, and the problem in the second phase is solved using two coordination rules, a CP solver, and three LNS-based heuristics with CP formulations. In this approach, the vessels are coordinated in a partially-cooperative way with multi-level interactions.

According to the simulation results, method M2.3 is better than methods M2.1 and M2.2. Method 2.3 reduces on average 8.03% total round-trip time (KPI 7), 22.59% total waiting time (KPI 9) and 10.47% port departure time (KPI 11) compared with the benchmark in which there is no coordination between vessels. This is positive for vessel operators, as they could spend less time waiting and sailing in the port area using this method. All vessels are expected to finish their transport tasks in a shorter time, which is beneficial for the port authority as the utilization or throughput of the port is improved. An additional advantage is that this approach provides the option to also schedule extra inter-terminal containers for transport by vessels efficiently so that economic benefits can be gained. Moreover, among the different heuristics in method M2.3: if the vessel operators prefer to spend less time round-trip time in the port, then LNS-worst heuristic is the best option; if the vessel operators prefer to spend less waiting time in the port, then LNS-random heuristic is the best option; if the port authority prefer that the vessels can finish their transport tasks in a shorter time, then LNS-Shaw heuristic is the best option. This chapter answers partially the Key Research Questions 2 and 3.

Chapter 5

Fully-cooperative planning for multi-level interaction in large ports

A partially-cooperative perspective was taken in the Chapters 3 and 4 for solving vessel rotation planning problems in small and medium seaports, as the vessel operators may be hesitant to share information with each other. To motivate the vessel operators to be more cooperative in planning their rotations in large seaports, it is worth investigating the potential benefits that they can gain from sharing more information in a fully-cooperative way. Therefore, this chapter proposes a fully-cooperative planning approach to improve the coordination between inland vessel operators. In addition, this chapter also investigates the possibility of using inland vessels for ITT. While Chapter 4 also considers optional ITT containers, this chapter investigates more extensively the possibility of using inland vessels for ITT by considering mandatory ITT containers. This implies that vessel agents cooperate with each other not only to transport their hinterland containers but also mandatory ITT containers in this chapter. To solve the vessel rotation planning problem in even larger seaports than in Chapter 4, a hybrid solution method based on Benders decomposition and LNS is proposed in this chapter, referred as method M3.1. Moreover, a closed-loop perspective is taken, in which possible disturbances that may happen in practice are also considered.

The research discussed in this chapter is based on [184, 186].

5.1 Introduction

This chapter assumes a central coordinator. For each vessel and terminal there is an agent that sends its information to a central coordinator. The central coordinator determines rotations for the terminal and vessels, with the aim to minimize the sum of the round-trip times of the vessels and reduce the idle time of terminals. Consequently, during busy hours, the problem sizes will increase substantially with the increase of incoming vessels. Therefore, a decomposition technique and heuristics are required to solve the problem in larger sizes. Meanwhile, in practice, disturbances may happen in the transport process of inland vessels

in the port, including the failure of terminal equipment, delay of sea-going vessels, delay of hinterland containers, as well as unexpected terminal closing. It is important to consider these possible disturbances in planning the rotations of vessels. This chapter proposes a hybrid solution method based on Benders decomposition and LNS, which can handle larger problem sizes and take possible disturbances into account.

Benders decomposition has been originally proposed for solving large mixed-integer programming problems in [27]. The classical Benders decomposition solves a problem by partitioning it into a mixed-integer master problem and linear subproblems. The solution process iterates between solving the master problem and the linear subproblems [216]. Benders decomposition can profitably combine operation research and constraint programming, since one approach can be applied to solve the master problem and one to solve the subproblem, depending which is the most suitable for the particular problem structure. This sort of combination has yielded substantial speedups in the computation [269].

Conventional Benders decomposition assumes that the master problem is mixed-integer and the subproblems are linear. However, the vessel rotation planning problem involves logical relations between vessel operators. For example, the waiting time of an inland vessel at a terminal not only depends on the time that the other vessels that are currently being handled at the same terminal spend, but also depends on the sequences how the other vessels visit the previous terminals. This implies that the calculation of the waiting time of a inland vessel at a terminal involves a sequence of variables representing how this vessel visit the current terminals, as well as a sequence of variables how the other vessels visit the previous terminals. Therefore, it is more straightforward to express such relations using logical constraints or global constraints from constraint programming.

A generalized Benders decomposition, defined as logic-based Benders decomposition, is proposed to solve the problem. Logic-based Benders decomposition is introduced by [130] and further developed by [131]. A major advantage of logic-based Benders decomposition is that the subproblem does not need to have a specific form: it can be an optimization problem, a constraint program, or a simple feasibility problem [318].

This chapter uses the logic-based Benders decomposition framework to decompose the problem into a rotation generation master problem and a rotation evaluation subproblem, both formulated as constraint programming problems. The master problem is formulated as an optimization problem, and the subproblem is formulated as a satisfaction problem. The major constraints are considered in the master problem, and the subproblem includes constraints that considers possible disturbances. The most complicated constraints are considered in the subproblem and in this way the complexity of the master problem could be reduced. Moreover, to handle even larger problem sizes, Large Neighborhood Search (LNS) heuristic which is proposed in Chapter 4, is incorporated in the Benders' decomposition framework to get rotations for the inland vessels in the master problem.

This chapter is organized as follows. Section 5.2 presents the structure of the proposed method. A novel coordination model of inland vessels for carrying inter-terminal containers besides hinterland containers is introduced in Section 5.3. Section 5.4 describes the hybrid solution method. Experimental results are given with respect to four aspects in Section 5.5: quality of solutions; potential improvements of the proposed method on logistical key performance indicators; the impact of extra ITT containers and extra vessel capacity on the overall performance as well as on the individual performance per vessels; capability for handling system disturbances of the proposed method. Section 5.6 concludes this chapter.

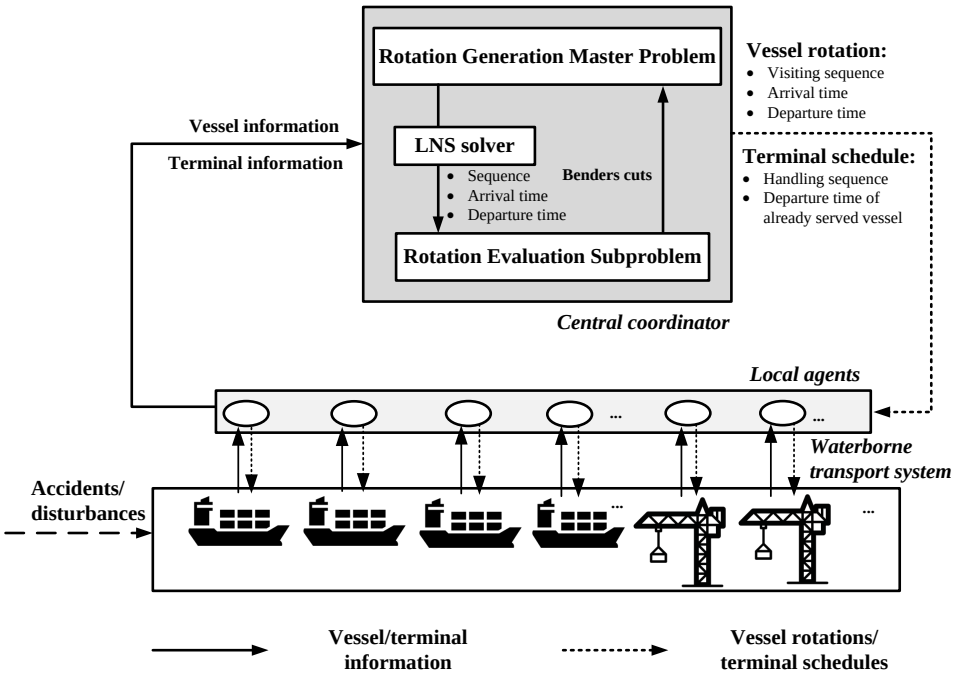

Figure 5.1: The structure of the fully-cooperative planning method.

5.2 The structure of the proposed method

Figure 5.1 illustrates the structure of the proposed fully-cooperative planning approach. The rotations of the vessels are planned by the central coordinator before they enter the port area with the current information of vessels and terminals. Therefore, the vessel information, including (1) the set of terminals to visit; (2) the number of hinterland containers to load and unload at each terminal; (3) the capacity of the vessel is already known. In addition, as the sea-going vessels usually have scheduled arrival and departure time at different terminals several days ahead, the arrival and departure time of sea-going vessels are also known. Similarly, the terminal information, including (1) the number of quays at each terminal; (2) the latest departure time of vessels that are currently being served by the terminal; (3) the distances between terminals; (4) the opening and closing time of terminals, are also considered known. The information of vessels and terminals will be used as the input for the central coordinator.

It is assumed that rotations will only be updated when disturbances happen. This means that the vessel agents will implement those rotations plans and will not change it unless disturbances happen. The terminal operators will serve the inland vessels first-come-first-served, as in practice. The vessel information and terminal information will be sent from the local agents to the central coordinator. Based on the vessel information and terminal information, the logic-based Benders decomposition framework is used to decompose the problem into a rotation generation master problem and a rotation evaluation subproblem.

The central coordinator firstly decides on the sequences of vessel visits to terminals and

the number of inter-terminal containers to be transported from one terminal to another by each vessel in the master problem. When the master problem generates rotations, the subproblem calculates the corresponding waiting times at different terminals of these vessels. Based on the values of the waiting times, Benders cuts are derived, which essentially are constraints on the waiting times in the subproblem and the visiting sequences of vessels in the master problem. These Benders cuts are added to the master problem in the next iteration in order to exclude variable assignments that can be no better than the variable assignments of the previous solution. Then the master problem is re-solved again with the newly added Benders cut to find a better solution. The details of the hybrid solution method will be described further in Section 5.4. When the solutions are obtained, the rotation plans will be sent from the central coordinator to the local agents in the waterborne transport system for implementation.

To make the proposed method useful for a longer period of time, disturbances that may occur in the inter-terminal transport process need to be considered. The following four types of disturbances are considered in this chapter:

1. The unavailability of quay cranes due to equipment failure or terminal strikes. This type of disturbance affects the waiting times of the vessels at the terminals.

2. The delay of sea-going vessels. This implies that the service time window of the sea-going vessels will be changed and updated by the terminals. As sea-going vessels have priority at terminals, during the updated service time window of the sea-going vessels, the inland vessels cannot be served. This affects the service time windows and waiting times of the inland vessels.

3. The delay of hinterland containers. This means that some of the hinterland containers are not ready when an inland vessel has arrived at the terminal. This causes the vessel to spend extra hours in the port until the containers are ready to be loaded.

4. The unexpected closing of terminals due to extreme weather. This means that the loading and unloading process of certain inland vessels are allowed to be interrupted by the accidental closing time of a terminal. This also affects the waiting times, as well as the service time windows of vessels at the terminals.

The details of how the disturbances are dealt with are introduced in Section 5.3. Whenever disturbances happen, the updated information of vessels and terminals will be resent to the central coordinator and then the Benders decomposition-based solution method starts over again.

5.3 Mathematical problem formulation

This section first introduces the rotation generation master problem and the rotation evaluation subproblem. Then the description of how the disturbances are dealt with in the formulation is given.

The problem formulation is based on time-segment graphs. An example of a time-segment graph is given in Figure 5.2, showing three rotation plans of three vessels. For example, as vessels 1 and 2 visit terminal 1 first, therefore, this situation is referred to as terminal 1 is on the first segment of vessels 1 and 2's rotation plans.

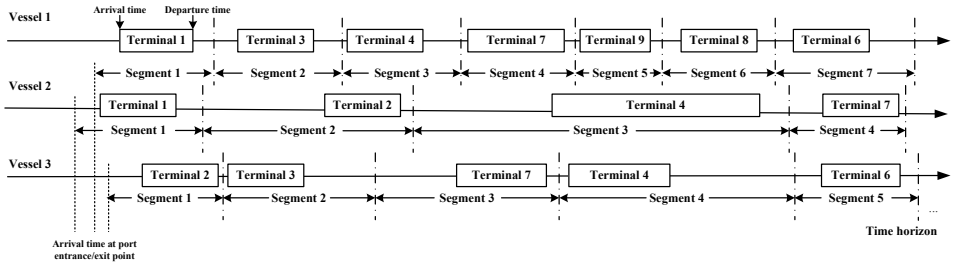

Figure 5.2: Time-segment graph for instance with 3 vessels.

Table 5.1: Parameters of problem formulation.

Symbols	Definitions
M	set of vessels entering the port;
N_m	set of terminals that vessel m needs to visit in the port;
Q_i	number of quays in terminal i.
R_{ij}	number of inter-terminal containers that need to be transported from terminal i to j;
$K_m^{\max}$	number of segments on the rotation plan of vessel m;
$t_i^{\text{load}}/t_i^{\text{unload}}$	average loading/unloading time, per loaded container at terminal i.
T_i^{entrance}	traveling time from the entrance/exit of the port to terminal i;
$T_{iq}^{\text{departure}}$	latest departure time of the vessels being served in terminal i at quay q;
T_i^{deadline}	the deadline for inter-terminal containers to be delivered to terminal i;
T_{ij}^{travel}	traveling time between terminal i to j;
l_i^m/u_i^m	number of hinterland containers that need to be loaded/unloaded by vessel m at terminal i;
C_m^{capacity}	carrying capacity of vessel m in TEU;
C_m^{original}	original number of containers on of vessel m before entering the port in TEU;
$[W_{\text{start}}^i, W_{\text{end}}^i]$	the closing time window of terminal i;
$[S_q^i, E_q^i]$	the service time window of the sea vessel at quay q of terminal i.

The parameters that will be used below in the formulations are concluded in Table 5.1. Most of the parameters have already been introduced in Chapter 4. It is referred to Table 4.2 and Table 4.4 for detailed explanations. This chapter introduce new parameters including the R_{ij}, which represents the number of inter-terminal containers that need to be transported between terminals in the port, as well as the deadline T_i^{deadline} before which the ITT containers need to be delivered to terminal i.

5.3.1 Master problem

The following constraint programming problem defines the master problem. Most of the decision variables have already been introduced in Chapter 4. It is referred to Table 4.3 and Table 4.5 for detailed explanations. For completeness, Table 5.2 concludes the decision variables that are used in the master problem. A new decision variable z_{ik}^m is introduced in this chapter, which represents whether vessel m visits terminal i on the k-th segment of its rotation. .

Table 5.2: Decision variables in the master problem.

Symbols	Definitions
z_{ik}^{m}	binary variable, $z_{ik}^{m}=1$ if terminal i is on segment k of vessel m's rotation, otherwise $z_{ik}^{m}=0$;
$\alpha_{mi}^{\text{master}}$	arrival time of vessel m at/from terminal i;
$\beta_{mi}^{\text{master}}$	departure time of vessel m at/from terminal i;
$\gamma_{mi}^{\text{master}}$	service time of vessel m at terminal i;
$\omega_{mi}^{\text{master}}$	waiting time of vessel m at terminal i;
v_{ij}^{m}	number of inter-terminal containers from terminal i to j carried by vessel m;
b_{i}^{m}	number of containers on vessel m when it leaves terminal i;
r_{mi}^{master}	ranking of the arrival time of vessel m at terminal i in the master problem.

Table 5.3: Auxiliary variables in the master problem.

Symbols	Definitions
$\varepsilon_{mm'i}$	binary variable, $\varepsilon_{mm'i}=1$ if the vessel m' arrives at terminal i earlier than vessel m, otherwise $\varepsilon_{mm'i}=0$;
$\eta_{i,r_{mi}^{\text{master}}}$	the departure time of the vessel that visits terminal i with ranking r_{mi}^{master}.

The aim of the master problem is to generate efficient rotation plans for the vessels. Therefore the objective is formulated as minimizing the sum of the times that vessels spend in the port area plus the waiting time variable W_m^{wait} from the sub-problem. The objective function is therefore as follows:

$$\min \quad \sum_{m=1}^{M}\left(\sum_{i\in N_m}\left(\beta_{mi}^{\text{master}}+T_i^{\text{entrance}}\right)z_{i,K_m^{\max}}^{m}+W_m^{\text{wait}}\right), \tag{5.1}$$

where $\beta_{mi}^{\text{master}}$ is the departure time of vessel m when it leaves the last terminal i on its rotation plan, T_i^{entrance} is the traveling time from the last terminal i to port entrance/exit point, and W_m^{wait} is the total the waiting time for vessel m, the lower bound of which is determined by the sub-problem.

To distinguish the various constraints that are introduced in this section, they are categorized into different types: (A) sequence and container-related constraints, which concern the constraints that relate to how each vessel visits different terminals, and the constraints on the number of inter-terminal containers that need to be transported and the corresponding loading and unloading time for each vessel; (B) capacity-related constraints, which concern the maximum number of containers that can be loaded on each vessel; (C) time-related constraints, which concern the relations among the sequence variables, arrival time and departure time variables; (D) terminal-related constraints, which concern the constraints on the possible start and end time of each terminal at each terminal. The corresponding constraints are introduced as follow:

A. Sequence and container-related constraints:

$$\sum_{k=1}^{K_m^{\max}} z_{ik}^{m}=1,\ \sum_{i\in N_m} z_{ik}^{m}=1 \qquad \forall i\in N_m, \forall k\in\{1,2,\ldots,K_m^{\max}\} \tag{5.2}$$

$$\sum_{m=1}^{|M|} v_{ij}^m = R_{ij} \qquad \forall i,j \in N_m, \forall m \in M \tag{5.3}$$

$$\sum_{k'=1}^{k} v_{ij}^m z_{ik}^m z_{jk'}^m = 0 \qquad \forall i,j \in N_m, \forall m \in M, \forall k' < k \leq K_m^{\max} \tag{5.4}$$

$$(T_j^{\text{deadline}} - \alpha_j^m) v_{ij}^m \geq 0 \qquad \forall i,j \in N_m, \forall m \in M \tag{5.5}$$

Constraint (5.2) ensures that each terminal will only be visited once. Constraint (5.3) ensures that the inter-terminal containers that need to be transported from terminal i to terminal j will be transported by the set of M vessels. Constraint (5.4) represents that at terminal i vessel m will not carry any inter-terminal container that needs to be transported from i to terminals that it has already visited. Constraint (5.5) represents that if there are some containers need to be transported from terminal i to j, then they must be transported to the destination before the deadline.

B. Capacity-related constraints:

$$\begin{aligned} b_i^m = &\left(C_m^{\text{original}} + l_i^m - u_i^m + \sum_{p \in N_m} v_{ip}^m \right) z_{i1}^m \\ &+ \left(l_i^m - u_i^m + \sum_{k=2}^{K_{\max}^m} \sum_{p \in N_m} \left(b_p^m z_{p,k-1}^m z_{ik}^m \right) + \sum_{p \in N_m} v_{ip}^m - \sum_{p \in N_m} v_{pi}^m \right) (1 - z_{i1}^m) \\ &\qquad \forall i \in N_m, \forall m \in M \end{aligned} \tag{5.6}$$

$$b_i^m \leq C_m^{\text{capacity}} \qquad \forall i \in N_m, \forall m \in M \tag{5.7}$$

Constraint (5.6) represents that firstly, if vessel m visits terminal i the first on its rotation ($z_{i1}^m = 1$), then the number of the number of containers on vessel m when it leaves terminal i equals the initial number of containers on-board C_m^{original}, plus the number of hinterland containers l_i^m that need to be loaded at terminal i and inter-terminal containers $\sum_{p \in N_m} v_{ip}^m$ that need to be transported from terminal i, minus the number of hinterland containers u_i^mthat need to be unloaded at terminal i; secondly, if vessel m does not visit terminal i the first ($z_{i1}^m = 0$), then the number of containers on-board when it leaves terminal i equals the number of containers on-board at the terminal that is visited just before terminal i ($\sum_{k=2}^{K_{\max}^m} \sum_{p \in N_m} \left(b_p^m z_{p,k-1}^m z_{ik}^m \right)$), plus the number of inter-terminal and hinterland containers that need to be loaded at terminal i ($\sum_{p \in N_m} v_{ip}^m$), minus the number of inter-terminal and hinterland containers that need to be unloaded at terminal i ($\sum_{p \in N_m} v_{pi}^m$). Constraint (5.7) ensures that the number of containers on-board of vessel m when it leaves terminal i will not exceed the capacity of each vessel.

C. Time-related constraints:

$$\alpha_{mi}^{\text{master}} = T_i^{\text{entrance}} z_{i1}^m + \left(\sum_{k=2}^{K_m^{\max}} z_{ik}^m \left(\sum_{p \in N_m, p \neq i} z_{p,k-1}^m (\beta_{mp}^{\text{master}} + T_{pi}^{\text{travel}}) \right) \right) (1 - z_{i1}^m) \qquad \forall i \in N_m, \forall m \in M \tag{5.8}$$

$$\beta_{mi}^{\text{master}} = \alpha_{mi}^{\text{master}} + \gamma_{mi}^{\text{master}} + \omega_{mi}^{\text{master}} \qquad \forall i \in N_m, \forall m \in M \quad (5.9)$$

$$\gamma_{mi}^{\text{master}} = l_i^m t^{\text{load}} + u_i^m t^{\text{unload}} + \sum_{p \in N_m, p \neq i} v_{ip}^m t^{\text{load}} + \sum_{p \in N_m, p \neq i} v_{pi}^m t^{\text{unload}} \qquad \forall i \in N_m, \forall m \in M \quad (5.10)$$

$$\alpha_{mi}^{\text{master}}(1 - \varepsilon_{mm'i}) < \alpha_{m'i}^{\text{master}} \qquad \forall i \in \{N_m \cap N_{m'}\}, \forall m, m' \in M \quad (5.11)$$

$$\alpha_{mi}^{\text{master}} \geq \alpha_{m'i}^{\text{master}} \varepsilon_{mm'i} \qquad \forall i \in \{N_m \cap N_{m'}\}, \forall m, m' \in M \quad (5.12)$$

$$r_{mi}^{\text{master}} = \sum_{m \in M} \varepsilon_{mm'i} \qquad \forall i \in N_m, \forall m, m' \in M \quad (5.13)$$

$$\omega_{mi}^{\text{master}} = \max\left(\alpha_{mi}^{\text{master}}, \eta_{i, r_{mi}^{\text{master}} - 1}\right) - \alpha_{mi}^{\text{master}} \qquad \forall i \in N_m, \forall m \in M \quad (5.14)$$

Constraint (5.8) ensures that firstly, if vessel m visits terminal i as the first terminal on its rotation ($z_{i1}^m = 1$), then the arrival time at i equals the traveling time from the port entrance point to terminal i; secondly, if vessel m does not visit terminal i the first ($z_{i1}^m = 0$), then the arrival time at terminal i equals the departure time from the previous terminal plus the traveling time. Constraint (5.9) represents that the departure time equals the arrival time plus the service time and waiting time. Constraint (5.10) represents that the service time of a vessel m at terminal i equals the sum of the loading and unloading time for the inter-terminal containers and hinterland containers. Constraints (5.11)–(5.13) together determine the handling order of the vessels that arrive at the same terminal, based on a first-come-first-served basis. Constraint (5.14) ensures that if there is another vessel that is being handled at a terminal, then the next upcoming vessels needs to wait. Therefore, the waiting time of vessel m at terminal i equals the difference between the earliest possible starting time of terminal i and the arrival time of vessel m at terminal i. Here the earliest possible starting time for vessel m equals the departure time $\eta_{i, r_{mi}^{\text{master}} - 1}$ of the vessel that has just been handled before m with the ranking of $r_{mi}^{\text{master}} - 1$.

D. Terminal-related constraints:

$$\eta_{i0} = T_i^{\text{departure}} \qquad \forall i \in N_m \quad (5.15)$$

$$\eta_{i, r_{mi}^{\text{master}}} = \max\left(\alpha_{mi}^{\text{master}}, \eta_{i, r_{mi}^{\text{master}} - 1}\right) + \gamma_{mi}^{\text{master}} \qquad \forall i \in N_m, \forall m \in M \quad (5.16)$$

Constraint (5.15) represents that initially, the earliest possible starting time for the first vessel arriving at terminal i equals $T_i^{\text{departure}}$, which is the latest departure time of the vessels that have already been served before the upcoming vessels. Constraint (5.16) ensures that the latest departure time of terminal i will be updated accordingly each time when a vessel has been loaded and unloaded.

5.3.2 Subproblem

Once the master problem has determined the sequence of vessels to different terminals, the waiting time of these rotation plans needs to be evaluated after considering the handling capacity of terminal, the opening/closing of terminals and the priority of sea-vessels. The solutions from the master problem include the arrival, departure time at different terminals, and the number of inter-terminal containers to load and unload at each terminal. Based on

Table 5.4: Decision variables in the sub-problem.

Symbols	Definitions
α_{mi}^{sub}	arrival time of vessel m at terminal i in the sub-problem;
β_{mi}^{sub}	departure time of vessel m at terminal i in the sub-problem;
ω_{mi}^{sub}	waiting time of vessel m at terminal i in the sub-problem;
r_{mi}^{sub}	ranking of the arrival time of vessel m at terminal i in the sub-problem.

Table 5.5: Auxiliary variables in the sub-problem.

Symbols	Definitions
$\delta_{mm'i}$	binary variable, $\delta_{mm'i} = 1$ if the vessel m' arrives at terminal i earlier than vessel m, otherwise $\delta_{mm'i} = 0$;
$\xi_{i,r_{mi}^{\text{sub}}}^{q}$	the possible starting time at quay crane q for the vessel that will visit terminal i with the ranking of r_{mi}^{sub};
$\lambda_{i,r_{mi}^{\text{sub}}}^{q}$	the departure time at quay crane q of terminal i after the vessel with the ranking of r_{mi}^{sub} has been handled;
$x_{mi}^{\text{start,terminal}}$	binary variable, $x_{mi}^{\text{start,terminal}} = 1$ if vessel m departures later than the start of closing time window of terminal i, otherwise $x_{mi}^{\text{start,terminal}} = 0$;
$x_{mi}^{\text{end,terminal}}$	binary variable, $x_{mi}^{\text{end,terminal}} = 1$ if vessel m arrives earlier than the end of closing time window of terminal i, otherwise $x_{mi}^{\text{end,terminal}} = 0$;
$x_{mi}^{\text{start,sea}}$	binary variable, $x_{mi}^{\text{start,sea}} = 1$ if vessel m departures later than the start of the handling time window of a sea-going vessel, otherwise $x_{mi}^{\text{start,sea}} = 0$;
$x_{mi}^{\text{end,sea}}$	binary variable, $x_{mi}^{\text{end,sea}} = 1$ if vessel m arrives earlier than the end of closing time window of the handling time window of a sea-going vessel, otherwise $x_{mi}^{\text{end,sea}} = 0$;
θ_{mi}^{q}	binary variable, $\theta_{mi}^{q} = 1$ if vessel m will be handled at quay crane q of terminal i, otherwise $\theta_{mi}^{q} = 0$.

the solution from the master problem, which consists of the optimal rotation plans for each vessel, the respective waiting time at each terminal for each vessel is determined. Therefore, the subproblem is a satisfaction problem using the CP formulation.

The decision variables in the subproblem are shown in Table 5.4. It is referred to Table 4.5 in Chapter 4 for detailed explanations. Auxiliary variables are also introduced to calculate decision variables r_{mi}^{sub} and ω_{mi}^{sub}, as shown in Table 5.5. Variable $\delta_{mm'i}$ is used to determine the ranking r_{mi}^{sub} of vessel m at terminal i by comparing its arrival time with the arrival time of another vessel m'. Variables $\xi_{i,r_{mi}^{\text{sub}}}^{q}$ and $\lambda_{i,r_{mi}^{\text{sub}}}^{q}$ are used to determine the earliest possible starting time for handling vessels that arrive at terminal i with the ranking r_{mi}^{sub} at quay crane q. Variables $x_{mi}^{\text{start,terminal}}$, $x_{mi}^{\text{end,terminal}}$, $x_{mi}^{\text{start,sea}}$ and $x_{mi}^{\text{end,sea}}$ are used to calculate the waiting time of vessel m at terminal i. Variable θ_{mi}^{q} is used to determine whether vessel m will be handled at quay crane q of terminal i.

The constraints in the subproblem are divided into four types: (E) vessel-related constraints; (F) terminal capacity-related constraints, which concern the relations among the different terminal capacities and the possible start and end times for each incoming ves-

sel; (G) opening/closing time-related constraints, which concern the relations among the waiting times of each vessel and the opening and closing time of each terminal; (H) sea vessel-related constraints, which concern the relations among the waiting times of each vessel and the start and end time for the sea vessels at different terminals.

E. Vessel-related constraints

$$\alpha_{mi}^{\text{sub}} = T_i^{\text{entrance}} z_{i1}^{m*} + \left(\sum_{k=2}^{K_m^{\max}} z_{ik}^{m*} \left(z_{p,k-1}^{m*} \sum_{p \in N_m, p \neq i} (\beta_{mp}^{\text{sub}} + T_{pi}^{\text{travel}}) \right) \right) (1 - z_{i1}^{m*}) \quad \forall i \in N_m, \forall m \in M \tag{5.17}$$

$$\beta_{mi}^{\text{sub}} = \alpha_{mi}^{\text{sub}} + \omega_{mi}^{\text{sub}} + \gamma_{mi}^{\text{master}*} \quad \forall i \in N_m, \forall m \in M \tag{5.18}$$

Constraints (5.17) and (5.18) determine the value of arrival and departure times of each vessel in the sub-problem, based on the solutions from the master problem including z_{ik}^{m*} and $\gamma_{mi}^{\text{master}*}$.

F. Terminal capacity-related constraints

$$\alpha_{mi}^{\text{sub}}(1 - \delta_{mm'i}) < \alpha_{m'i}^{\text{sub}} \quad \forall i \in \{N_m \cap N_{m'}\}, \forall m, m' \in M \tag{5.19}$$

$$\alpha_{m'i}^{\text{sub}} \delta_{mm'i} \leq \alpha_{mi}^{\text{sub}} \quad \forall i \in \{N_m \cap N_{m'}\}, \forall m, m' \in M \tag{5.20}$$

$$r_{mi}^{\text{sub}} = \sum_{m=1}^{M} \delta_{mm'i} \quad \forall i \in N_m, \forall m, m' \in M \tag{5.21}$$

$$\xi_{i0}^{q} = T_{iq}^{\text{departure}} \quad \forall i \in N_m, \forall q \in Q_i \tag{5.22}$$

$$\xi_{i,r_{mi}^{\text{sub}}}^{q} = \lambda_{i,r_{mi}^{\text{sub}}-1}^{q} \quad \forall i \in N_m \tag{5.23}$$

$$\left(\xi_{i,r_{mi}^{\text{sub}}}^{q} - \min\left(\xi_{i,r_{mi}^{\text{sub}}}^{1}, \ldots, \xi_{i,r_{mi}^{\text{sub}}}^{Q_i} \right) \right) \theta_{mi}^{q} = 0 \quad \forall i \in N_m, \forall m \in M \tag{5.24}$$

$$\xi_{i,r_{mi}^{\text{sub}}}^{q} > \min\left(\xi_{i,r_{mi}^{\text{sub}}}^{1}, \ldots, \xi_{i,r_{mi}^{\text{sub}}}^{Q_i} \right) (1 - \theta_{mi}^{q}) \quad \forall i \in N_m, \forall m \in M \tag{5.25}$$

Similar to Constraints (5.11)–(5.13), Constraints (5.19)–(5.21) are also used to determine the handling order of vessel m at terminal i. These constraints are the extension of Constraints (4.25)–(4.27) without logical operators. Constraint (5.22) represents that the possible starting time for the first vessel arriving at terminal i at quay crane q equals $T_{jq}^{\text{departure}}$, which is the latest departure time of the vessels already being served before the upcoming vessels. Constraint (5.23) represents that the possible starting time for the vessel arriving at quay crane q of terminal i with ranking r_{mi}^{sub} equals the latest departure time of the vessels arrived earlier at terminal i with ranking $r_{mi}^{\text{sub}} - 1$. Constraints (5.24) and (5.25) are used indicates whether quay crane q has the earliest possible starting time. They are extensions of Constraint (4.32) without logical operators. If quay crane q has the earliest starting time, which implies that $\xi_{i,r_{mi}^{\text{sub}}}^{q} = \min(\xi_{i,r_{mi}^{\text{sub}}}^{1}, \xi_{i,r_{mi}^{\text{sub}}}^{2}, \ldots, \xi_{i,r_{mi}^{\text{sub}}}^{Q_i})$, auxiliary binary variable $\theta_{mi}^{q} = 1$ and vessel m will be served at this quay. If quay crane q does not have the earliest starting time, which implies that $\xi_{i,r_{mi}^{\text{sub}}}^{q} > \min(\xi_{i,r_{mi}^{\text{sub}}}^{1}, \xi_{i,r_{mi}^{\text{sub}}}^{2}, \ldots, \xi_{i,r_{mi}^{\text{sub}}}^{Q_i})$, binary variable $\theta_{mi}^{q} = 0$ and vessel m will not be served at this quay.

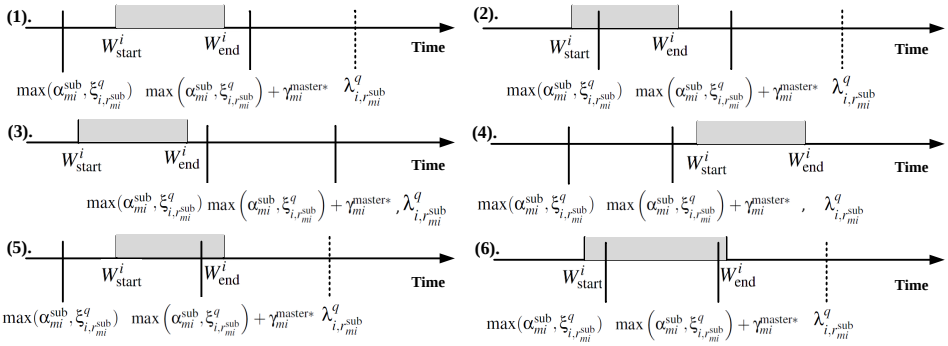

Figure 5.3: Six possible relations between a vessel's arrival, departure time window and closing of the terminal (adapted from [86]).

When $\theta^q_{mi} = 1$, vessel m will be served at the quay crane q. To determine the waiting time at quay q, the opening/closing time-related constraints and sea vessel-related constraints are further introduced in the next section.

G. Opening/closing time-related constraints

$$\left(\max\left(\alpha^{\text{sub}}_{mi}, \xi^q_{i,r^{\text{sub}}_{mi}}\right) + \gamma^{\text{master}*}_{mi} - W^i_{\text{start}} x^{\text{start,terminal}}_{mi}\right)\theta^q_{mi} \geq 0 \quad \forall m \in M, \forall i \in N_m, \forall q \in Q_i \tag{5.26}$$

$$\left(\left(\max\left(\alpha^{\text{sub}}_{mi}, \xi^q_{i,r^{\text{sub}}_{mi}}\right) + \gamma^{\text{master}*}_{mi}\right)(1 - x^{\text{start,terminal}}_{mi}) - W^i_{\text{start}}\right)\theta^q_{mi} < 0 \quad \forall m \in M, \forall i \in N_m, \forall q \in Q_i \tag{5.27}$$

$$\left(\max\left(\alpha^{\text{sub}}_{mi}, \xi^q_{i,r^{\text{sub}}_{mi}}\right)(1 - x^{\text{end,terminal}}_{mi}) - W^i_{\text{end}}\right)\theta^q_{mi} \leq 0 \quad \forall m \in M, \forall i \in N_m, \forall q \in Q_i \tag{5.28}$$

$$\left(\max\left(\alpha^{\text{sub}}_{mi}, \xi^q_{i,r^{\text{sub}}_{mi}}\right) - W^i_{\text{end}} x^{\text{end,terminal}}_{mi}\right)\theta^q_{mi} > 0 \quad \forall m \in M, \forall i \in N_m, \forall q \in Q_i \tag{5.29}$$

$$\begin{aligned}&\Big(\omega^{\text{sub}}_{mi} - \Big(\max\left(\alpha^{\text{sub}}_{mi}, \xi^q_{i,r^{\text{sub}}_{mi}}\right) - \alpha^{\text{sub}}_{mi} + \max\left(\max\left(\alpha^{\text{sub}}_{mi}, \xi^q_{i,r^{\text{sub}}_{mi}}\right), W^i_{\text{end}}\right)\\ &- \max\left(\max\left(\alpha^{\text{sub}}_{mi}, \xi^q_{i,r^{\text{sub}}_{mi}}\right), W^i_{\text{start}}\right)\Big)\Big) x^{\text{start,terminal}}_{mi} x^{\text{end,terminal}}_{mi}\theta^q_{mi} = 0\\ &\forall m \in M, \forall i \in N_m, \forall q \in Q_i\end{aligned} \tag{5.30}$$

$$\left(\omega^{\text{sub}}_{mi} - (\max\left(\alpha^{\text{sub}}_{mi}, \xi^q_{i,r^{\text{sub}}_{mi}}\right) - \alpha^{\text{sub}}_{mi})\right)(1 - x^{\text{start,terminal}}_{mi} x^{\text{end,terminal}}_{mi})\theta^q_{mi} = 0 \quad \forall m \in M, \forall i \in N_m, \forall q \in Q_i \tag{5.31}$$

$$\left(\lambda^q_{i,r^{\text{sub}}_{mi}} - \left(\alpha^{\text{sub}}_{mi} + \omega^{\text{sub}}_{mi} + \gamma^{\text{master}*}_{mi}\right)\right)\theta^q_{mi} = 0 \quad \forall m \in M, \forall i \in N_m, \forall q \in Q_i \tag{5.32}$$

To take into account opening/closing of terminals, different relations of a vessel's arrival,

departure time window and the closing of a terminal need to be identified. The same relations as shown in Figure 4.3 of Chapter 4 are considered. Figure 5.3 shows six possible relations that are considered in our formulation. When a vessel is assigned to a quay of a terminal, it has to wait until the other vessel that is being served at the quay has left. Thus, the actual starting time for vessel m with arrival time a_{mi}^{sub} is $\max(a_{mi}^{\text{sub}}, \xi_{i,kr_{mi}^{\text{sub}}}^{q})$, and the updated departure time at quay q of terminal i is $\max(a_{mi}^{\text{sub}}, \xi_{i,r_{mi}^{\text{sub}}}^{q}) + s_{mi}^{\text{sub}}$.

According to Figure 5.3, situations 1[1] and 5 show that $\max\left(\alpha_{mi}^{\text{sub}}, \xi_{i,r_{mi}^{\text{sub}}}^{q}\right) \leq W_{\text{start}}^{i} \wedge \max\left(\alpha_{mi}^{\text{sub}}, \xi_{i,r_{mi}^{\text{sub}}}^{q}\right) \leq W_{\text{end}}^{i} \wedge \max\left(\alpha_{mi}^{\text{sub}}, \xi_{i,r_{mi}^{\text{sub}}}^{q}\right) + \gamma_{mi}^{\text{master}*} \geq W_{\text{start}}^{i}$. In these two situations, the waiting time of vessel m at terminal i equals $W_{\text{end}}^{i} + \max\left(\alpha_{mi}^{\text{sub}}, \xi_{i,r_{mi}^{\text{sub}}}^{q}\right) - W_{\text{start}}^{i} - \alpha_{mi}^{\text{sub}}$. Situations 2 and 6 show that
$\max\left(\alpha_{mi}^{\text{sub}}, \xi_{i,r_{mi}^{\text{sub}}}^{q}\right) > W_{\text{start}}^{i} \wedge \max\left(\alpha_{mi}^{\text{sub}}, \xi_{i,r_{mi}^{\text{sub}}}^{q}\right) < W_{\text{end}}^{i} \wedge \max\left(\alpha_{mi}^{\text{sub}}, \xi_{i,r_{mi}^{\text{sub}}}^{q}\right) + \gamma_{mi}^{\text{master}*} > W_{\text{start}}^{i}$.

In these two situations, the waiting time of vessel m at terminal i equals $W_{\text{end}}^{i} - \alpha_{mi}^{\text{sub}}$. Situations 3[2] and 4 show that $\max\left(\alpha_{mi}^{\text{sub}}, \xi_{i,r_{mi}^{\text{sub}}}^{q}\right) > W_{\text{end}}^{i} \vee \max\left(\alpha_{mi}^{\text{sub}}, \xi_{i,r_{mi}^{\text{sub}}}^{q}\right) + \gamma_{mi}^{\text{master}*} < W_{\text{start}}^{i}$. In these two situations, the waiting time of vessel m at terminal i equals $\max\left(\alpha_{mi}^{\text{sub}}, \xi_{i,r_{mi}^{\text{sub}}}^{q}\right) - \alpha_{mi}^{\text{sub}}$.

Actually, the waiting time of vessel m at terminal i in situations 1, 2, 5 and 6 can be summarized as
$\omega_{mi}^{\text{sub}} = \max\left(\alpha_{mi}^{\text{sub}}, \xi_{i,r_{mi}^{\text{sub}}}^{q}\right) - \alpha_{mi}^{\text{sub}} + \max\left(\max\left(\alpha_{mi}^{\text{sub}}, \xi_{i,r_{mi}^{\text{sub}}}^{q}\right), W_{\text{end}}^{i}\right)$
$- \max\left(\max\left(\alpha_{mi}^{\text{sub}}, \xi_{i,r_{mi}^{\text{sub}}}^{q}\right), W_{\text{start}}^{i}\right)$. Consequently, situations 1, 2, 5 and 6 can also be summarized as $\max\left(\alpha_{mi}^{\text{sub}}, \xi_{i,r_{mi}^{\text{sub}}}^{q}\right) \leq W_{\text{end}}^{i} \wedge \max\left(\alpha_{mi}^{\text{sub}}, \xi_{i,r_{mi}^{\text{sub}}}^{q}\right) + \gamma_{mi}^{\text{master}*} \geq W_{\text{start}}^{i}$, and they are referred as Relation 1. Situations 3 and 4 are referred as Relation 2.

Binary auxiliary variables $x_{mi}^{\text{start,terminal}}$ and $x_{mi}^{\text{end,terminal}}$ are introduced to distinguish between the different relations of vessel m's arrival time $\max(\alpha_{mi}^{\text{sub}}, \xi_{i,r_{mi}^{\text{sub}}}^{q})$ to the closing time window $[W_{\text{start}}^{i}, W_{\text{end}}^{i}]$ of terminal i. Variable $x_{mi}^{\text{start,terminal}} = 1$ when $\max(\alpha_{mi}^{\text{sub}}, \xi_{i,r_{mi}^{\text{sub}}}^{q}) + \gamma_{mi}^{\text{master}*} \geq W_{\text{start}}^{i}$, which implies that vessel m departs later than the start of closing time window at terminal i, as shown in Constraint (5.26); variable $x_{mi}^{\text{start,terminal}} = 0$ when $\max(\alpha_{mi}^{\text{sub}}, \xi_{i,r_{mi}^{\text{sub}}}^{q}) + \gamma_{mi}^{\text{master}*} < W_{\text{start}}^{i}$, which implies that vessel m departs earlier than the start of closing time window at terminal i, as shown in Constraint (5.27). Similarly, variable $x_{mi}^{\text{end,terminal}} = 1$ when $\max(\alpha_{mi}^{\text{sub}}, \xi_{i,r_{mi}^{\text{sub}}}^{q}) < W_{\text{end}}^{i}$, which implies that vessel m arrives earlier than the end of closing time window at terminal i, as shown in Constraint (5.28); variable $x_{mi}^{\text{end,terminal}} = 0$ when $\max(\alpha_{mi}^{\text{sub}}, \xi_{i,r_{mi}^{\text{sub}}}^{q}) \geq W_{\text{end}}^{i}$, which implies that vessel m arrives later than the end of closing time window at terminal i, as shown in Constraint (5.29). Therefore, Relation 1 can be expressed as $x_{mi}^{\text{start,terminal}} = 1 \wedge x_{mi}^{\text{end,terminal}} = 1$; Relation 2 can be expressed as $x_{mi}^{\text{start,terminal}} = 0 \wedge x_{mi}^{\text{end,terminal}} = 0$.

After identifying the above-mentioned relations, the waiting time ω_{i}^{m} that vessel m spends at terminal i with relations 1 and 2 are represented by Constraint (5.30) and Con-

[1] $a \wedge b$: statement a and statement b are both true.
[2] $a \vee b$: either statement a or statement b is true.

straint (5.31), respectively. The departure time $\lambda^q_{i,r^{\text{sub}}_{mi}}$ at quay crane q of terminal i with rank r_{mi} are represented in Constraints (5.32). Constraints (5.26)–(5.32) are extensions of Constraints (4.33)–(4.38) in Chapter 4, in which additional auxiliary variables are introduced in order to leave out the logical operators.

H. Sea vessel-related constraints

$$\left(\max\left(\alpha^{\text{sub}}_{mi},\xi^q_{i,r^{\text{sub}}_{mi}}\right)+\gamma^{\text{master}*}_{mi}-W^i_{\text{start}}x^{\text{start,sea}}_{mi}\right)\theta^q_{mi}\geq 0 \quad \forall m\in M,\forall i\in N_m,\forall q\in Q_i \tag{5.33}$$

$$\left(\left(\max\left(\alpha^{\text{sub}}_{mi},\xi^q_{i,r^{\text{sub}}_{mi}}\right)+\gamma^{\text{master}*}_{mi}\right)(1-x^{\text{start,sea}}_{mi})-W^i_{\text{start}}\right)\theta^q_{mi}<0 \quad \forall m\in M,\forall i\in N_m,\forall q\in Q_i \tag{5.34}$$

$$\left(\max\left(\alpha^{\text{sub}}_{mi},\xi^q_{i,r^{\text{sub}}_{mi}}\right)(1-x^{\text{end,sea}}_{mi})-W^i_{\text{end}}\right)\theta^q_{mi}\leq 0 \quad \forall m\in M,\forall i\in N_m,\forall q\in Q_i \tag{5.35}$$

$$\left(\max\left(\alpha^{\text{sub}}_{mi},\xi^q_{i,r^{\text{sub}}_{mi}}\right)-W^i_{\text{end}}x^{\text{end,sea}}_{mi}\right)\theta^q_{mi}>0 \quad \forall m\in M,\forall i\in N_m,\forall q\in Q_i \tag{5.36}$$

$$\begin{aligned}&\Big(\omega^{\text{sub}}_{mi}-\Big(\max\left(\alpha^{\text{sub}}_{mi},\xi^q_{i,r^{\text{sub}}_{mi}}\right)-\alpha^{\text{sub}}_{mi}+\max\left(\max\left(\alpha^{\text{sub}}_{mi},\xi^q_{i,r^{\text{sub}}_{mi}}\right),W^i_{\text{end}}\right)\\&-\max\left(\max\left(\alpha^{\text{sub}}_{mi},\xi^q_{i,r^{\text{sub}}_{mi}}\right),W^i_{\text{start}}\right)\Big)\Big)x^{\text{start,sea}}_{mi}x^{\text{start,sea}}_{mi}\theta^q_{mi}=0\\&\qquad\forall m\in M,\forall i\in N_m,\forall q\in Q_i\end{aligned} \tag{5.37}$$

$$\left(\omega^{\text{sub}}_{mi}-(\max\left(\alpha^{\text{sub}}_{mi},\xi^q_{i,r^{\text{sub}}_{mi}}\right)-\alpha^{\text{sub}}_{mi})\right)(1-x^{\text{start,sea}}_{mi}x^{\text{end,sea}}_{mi})\theta^q_{mi}=0 \quad \forall m\in M,\forall i\in N_m,\forall q\in Q_i \tag{5.38}$$

$$\left(\lambda^q_{i,r^{\text{sub}}_{mi}}-\left(\alpha^{\text{sub}}_{mi}+\omega^{\text{sub}}_{mi}+\gamma^{\text{master}*}_{mi}\right)\right)\theta^q_{mi}=0 \quad \forall m\in M,\forall i\in N_m,\forall q\in Q_i \tag{5.39}$$

Sea-going vessels are also considered, which always have priority over inland vessels. The calculation of the waiting time caused by sea-going vessels is similar to the calculation of the waiting time caused by closing of terminals, which is represented via Constraints (5.34)–(5.39), except for that the closing time window $[W^i_{\text{start}},W^i_{\text{end}}]$ is replaced with $[S^i_q,E^i_q]$, which represents the estimated start and end time of the sea-going vessel at quay q of terminal i.

5.3.3 Dealing with disturbances

This chapter considers four types of disturbances: (1) the unavailability quay cranes; (2) the delay of sea-going vessels; (3) the delay of hinterland containers; (4) the unexpected closing of terminals. These disturbances may happen at the same time or at different times. When at least one of these disturbances happens, the previously planned rotations may not be efficient any more. Therefore, the vessels in the port will be re-planned first and then the

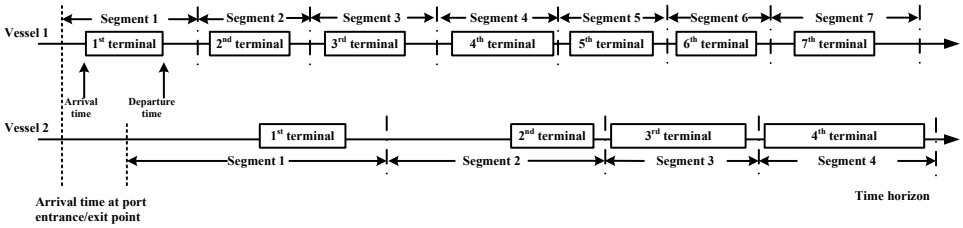

Figure 5.4: Original time-segment graph before disturbance happens.

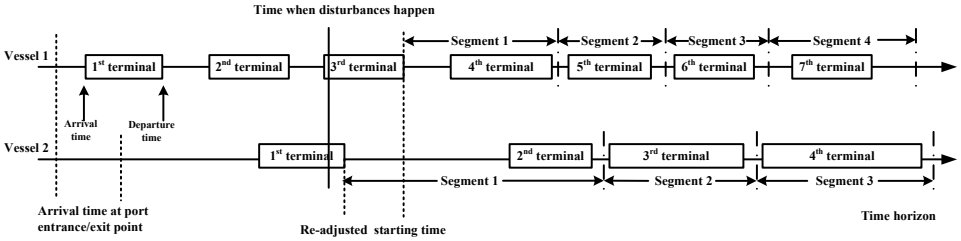

Figure 5.5: Updated time-segment graph before disturbance happens.

upcoming vessels will be planned based on the updated information of the terminals and vessels that are already in the port.

For the vessels that are already in the port area, the re-planning is based on an updated time-segment graph. Figure 5.4 and Figure 5.5 show a simple instance of an updated time-segment graph with 2 vessels. When disturbances happen, the terminals that are on the rotations of the inland vessels in the port are divided into two types: the already visited terminals and the upcoming terminals to visit. For vessel 1 and vessel 2, the already visited terminals will be removed from the set of terminals N_1 and N_2 that they have to visit. In addition, the segments will be re-divided according to the number of terminals they still need to visit. In addition, the information of the vessels and terminals will be updated based on the current status of the port. The updated information of a vessel includes: (1) the rest of terminals it needs to visit (N_m); (2) the planned arrival time at the next terminal (α_{mi}); (3) the number of inter-terminal containers it has already transported (v_{ij}^m).

The updated information of a terminal includes: (1) the currently available quay cranes (Q_i); (2) the latest departure time of the vessels that are currently being served ($T_{iq}^{\text{departure}}$); (3) the number of inter-terminal containers that still need to be transported (R_{ij}); (4) the updated closing and opening time ($[W_{\text{start}}^j, W_{\text{end}}^j]$). With the updated information, the parameters in the constraints will also be updated and new constraints will be added to the model, which will be introduced as follows.

For the upcoming vessels, they are planned based on the up-to-date information of the vessels in the port and terminals after their re-planning process are finished. Therefore, the constraints used for scheduling upcoming vessels will be the same as the vessels that are already in the port, except that the parameters reflecting the up-to-date information of terminals will be different.

Terminal equipment failure

When equipment failure happens in a particular terminal i, the constraints that are relevant to that terminal need to be updated based on the current status. This means constraints in the subproblem need to be changed based on the current available number of quay cranes Q_i. As the constraints about the quay cranes are also considered in the subproblem, the constraints in the master problem remain the same, while Constraints (5.19)-(5.24) and Constraints (5.27)–(5.32) need to be updated, the constraints for sea-going vessels will be updated as well.

Delay of sea-going vessels

When the delay of a sea-going vessel happens at a particular terminal, the sea-going vessel will schedule a new service time window with the terminal. As sea-going vessels always have priorities, during the new service time window the inland vessels cannot be scheduled at the same quay cranes as the sea-going vessel. Therefore, Constraints (5.27)–(5.32)with the old service time window $[S_q^i, E_q^i]$ will be removed. Constraints (5.27)–(5.32) with the new updated service time window $[S_q^i, E_q^i]$ for the sea-going vessel will be added in the model.

Delay of hinterland containers

When the delay of certain hinterland containers happens at a particular terminal, this mean even if the vessel has arrived at the terminal earlier, it has to wait until the hinterland containers are ready. It means that the terminal service will be available for the vessel until the required containers are ready. Therefore, similar constraints as Constraints (5.27)–(5.32) are introduced, except that the W_{start}^j is replaced with the current time, and W_{end}^j is replaced with the time when the hinterland containers are ready.

Unexpected terminal closing

When a particular terminal has to be closed due to extreme weather conditions or accidents, the inland vessels have to go to other terminals that are opened or waiting until the terminal is re-opened. This disturbance is considered by introducing constraints similar to Constraints (5.27)–(5.32). Assume that terminal i will be closed during time window $[W_{\text{start}}^{\text{unexpected}}, W_{\text{end}}^{\text{unexpected}}]$, the constraints for unexpected terminal closing will be the same as in (33)-(38), except that W_{start}^i and W_{end}^i are replaced by $W_{\text{start}}^{\text{unexpected}}$ and $W_{\text{end}}^{\text{unexpected}}$, respectively.

5.4 M3.1: Hybrid solution method

The basic steps of the solution method is illustrated in Algorithm 8 for the central coordinator and Algorithm 9 for the local agents, respectively. During one iteration, the algorithm first solves the master problem and generates initial solution $z_{ik}^{m*}, \alpha_i^{m*}, \beta_i^{m*}$, and γ_i^{m*}. Based on the initial solution, the subproblems are solved to determine the waiting time $\omega_{mi}^{\text{sub}*}$ for

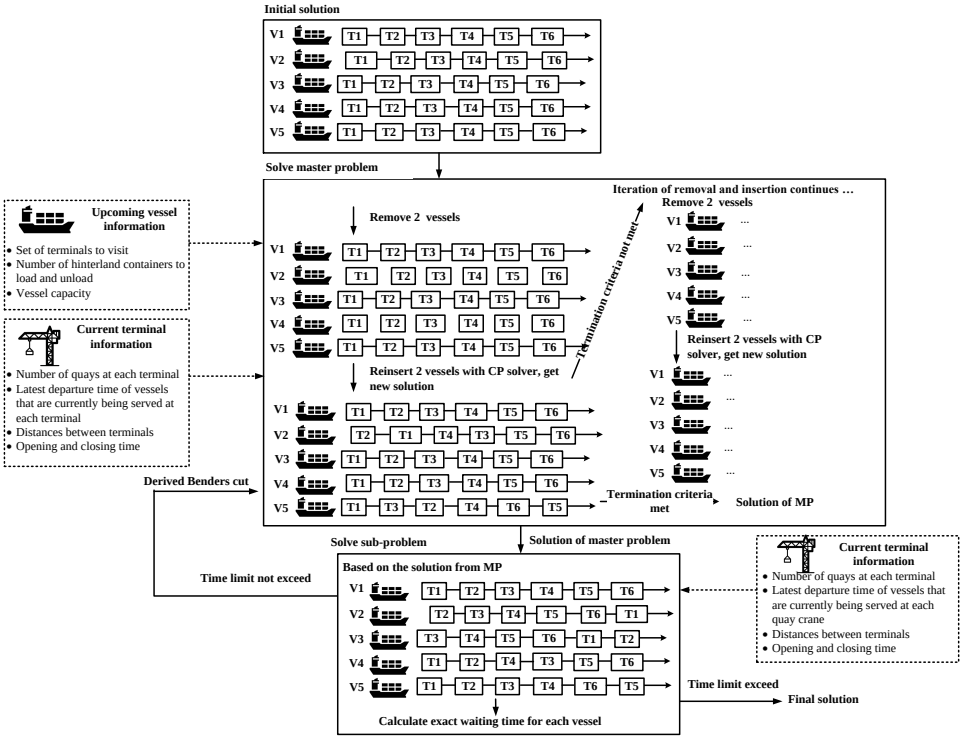

Figure 5.6: Illustration of the hybrid solution method.

Algorithm 8 Basic steps of the hybrid solution method carried out by the central coordinator

```
1: if iteration = 1 then receive information M, N_m, l_i^m, u_i^m from vessel agents, information R_ij from
   terminal agents, generate initial solution using CP solver;
2: while iteration > 1 and termination criteria not reached do
3: // Solve rotation generation master problem
4:     while time limit for solving master problem not exceeded do
5:         receive information M, N_m, l_i^m, u_i^m, v_ij^m and R_ij from vessel and terminal agents;
6:         re-solve the master problem with all Benders cut and save the current solution s;
7:     // Solve rotation generation master problem with LNS heuristic
8:         while time limit for LNS not exceeded do
9:             remove q vessels from the current solution s;
10:            keep the value of the rest of M − q vessels;
11:            reinsert q vessels to solution s with CPLEX CP solver, get new solution s';
12:            if obj(s') ≤ obj(s_best) then s_best = s, save the obj(s);
13:    return s_best and obj(s_best) to subproblem;
14: // Solve rotation evaluation subproblem based on s_best
15:    solution z_ik^m*, α_i^m*, β_i^m*, γ_i^m* ∈ s_best from master problem as the input;
       receive information Q_i, [W_start^i, W_end^i], and [S_start^i, S_end^i] from terminal agents as the input;
16:    calculate waiting time ω_mi^sub* for each vessel m;
17: // Derive Benders cut
18:    Create Benders cut with subproblem solution ω_mi^sub* and master problem variable z_ik^m;
19:    add new Benders cut to the master problem;
20: return s_best and obj(s_best) to local agents.
```

each vessel m. Then Benders cut for the master problem variable z_{ik}^m is created. The generated Benders cut will be added to the master problem. In the subsequent iteration, the master problem will be re-solved using LNS.

The procedure of LNS heuristic is similar to the one described in Algorithm 7 of Chapter 4. During the LNS, the central coordinator first removes q vessels from the current solution s'. Then the CPLEX CP solver is used to generate solutions for the already removed q vessels. At this point, a new solution s' has been found. This new solution is then compared with the current best solution s_{best} in terms of the objective value. If the new solution is better than the current best solution, the new solution is saved as the current best solution, and save the new objective value as the current best objective value. Then the removal and reinsertion process starts over again. If the objective value of the new solution is worse than the current best solution, this new solution are deleted and the removal and reinsertion iteration starts over again by removing another randomly chosen q vessels and reinserting them with the CPLEX CP solver. The whole procedure stops using given termination criteria: it stops after a total number of N iterations. When the LNS in this iteration has finished, the subproblem is solved again based on the current best solution from the master problem, and a new Benders cut is derived. Then the master problem will be re-solved again and another iteration begins.

The round-trip time of an inland vessel consists of the sum of service time, traveling time, waiting time, and the arrival time at the port entrance, in which the waiting time is dependent on the visiting sequences of each vessel. Therefore, Benders cut is derived for the variable defining the visiting sequence z_{ik}^m and waiting time variable W_m^{wait}. The cut from

Algorithm 9 Basic steps of the hybrid solution method carried out by the local agents

1: **if** iteration $= 1$ **then** vessel agents send information M, N_m, l_i^m, u_i^m, and terminal agents send R_{ij} to central coordinator;
2: **while** iteration > 1 and while termination criteria for central coordinator not reached **do**
3: // Vessel/terminal agents send information to central coordinator before solving master problem
4: vessel agents send information M, N_m, l_i^m, u_i^m, v_{ij}^m and terminal agents send R_{ij} to central coordinator;
5: // Terminal vessel agents send information to central coordinator before solving subproblem
6: terminal agents send information Q_i, $[W_{\text{start}}^i, W_{\text{end}}^i]$, and $[S_{\text{start}}^i, S_{\text{end}}^i]$ to central coordinator
7: **return** vessel/terminal agents receive solutions from central coordinator.

subproblem in iteration h is:

$$W_m^{\text{wait}} \geq \sum_{i \in N_m} \omega_{mih}^{\text{sub}*} - \sum_{i \in N_m} (1 - z_{ik}^m) \omega_{mih}^{\text{sub}*} \tag{5.40}$$

According to [64], a valid Benders cut as a logical expression should adhere to two conditions: (1) the cut removes the current solution from the master problem; (2) the cut does not eliminate any global optimal solutions.

In cut (5.40), W_m^{wait} is a variable representing the sum of vessel m's waiting times at different terminals. Solution $\omega_{mih}^{\text{sub}*}$ represents the waiting time of vessel m at terminal i, which is found in iteration h when solving the rotation evaluation sub-problem. This cut states that the future solution of the waiting time can only decreases if another sequence of a vessel's visits to terminals is given. That is, if the same assignment is given to the sub-problem, the z_{ik}^m variables that are part of this cut will all equal 1. If this is the case, then $(1 - z_{ik}^m) = 0$ for all m and the sum of waiting times in the sub-problem becomes a lower bound on W_m^{wait}. When a different visiting sequence of a vessel is chosen and at least one of the z_{ik}^m variables that previously had a value of 1 turns to 0, the waiting time of vessel m at that terminal i is removed from the sub-problem. This would result in a smaller lower bound for W_m, which consists of the sum of waiting times in which the waiting time at terminal i has been removed. This cut follows the 2 conditions defined by [64] to be a valid cut: the cut removes the current solution from the master problem and does not eliminate any global optimal solutions.

5.5 Simulation experiments

Benchmark Layout 3, and KPI 2 (quality of solutions), KPI 7 (total round-trip time), KPI 8 (round-trip time per vessel), KPI 9 (total waiting time), KPI 10 (waiting time per vessel) and KPI 11 (latest departure time) that were introduced in Chapter 2 are used to evaluate the proposed solution method M3.1. The number of incoming vessels varies from 8 vessels to 16 vessels, and each vessel visits 8 terminals in the port. It is referred to Chapter 2 for a detailed description.

In this section, firstly, the quality of solutions with different running time limits is presented. Secondly, the logistical performances of the proposed method is evaluated through the comparison of the results from scenarios with vessels' coordination and without vessels'

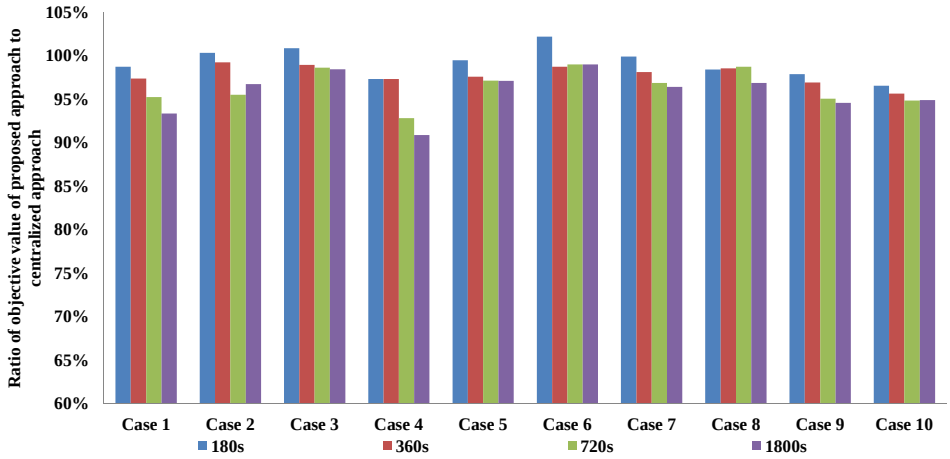

Figure 5.7: The quality of solutions (KPI 2) with different time limits.

coordination. Thirdly, the impact of extra ITT containers and extra vessel capacity on the overall performance of a set of vessels, as well as on the individual performance per vessel, is investigated. In the end, the logistical performance of the proposed method on handling four different types of disturbances is presented and analyzed. Four typical scenarios are used as examples to show the rotation plans before and after re-planning.

5.5.1 Experimental settings

CPLEX 12.6 CP solver is used as the CP solver. The proposed heuristic is implemented in C++. For the LNS-heuristics, in each iteration, 20% $\sim$ 30% of the vessels are removed from the current solution. The time limits for running the CP solver in the LNS-based heuristic is set as 30 seconds in each iteration.

To evaluate the solution quality of the proposed solution method, a centralized formulation is used for comparison, in which the master problem and the sub problem are considered together in the mathematical formulation as a large CP problem. This large CP problem (5.2)–(5.39) is solved using the CPLEX 12.6 CP solver and this method is referred as benchmark method. Moreover, to show the performance of the proposed coordination method on logistical KPIs including KPI 7, KPI 8, KPI 9, KPI 10, and KPI 11, the situation without any coordination between vessels is set as benchmark.

Four types of disturbances are considered in the experiments: (1) unavailable quay cranes; (2) delay of sea-going vessel; (3) delay of hinterland containers; (4) terminal closing due to extreme weather.

5.5.2 Quality of solutions

As it is difficult to obtain optimal solutions in a reasonable time for large problem sizes, therefore 4 different run time limits are imposed on the 10 cases in Scenario 1. This means the values reported in Figure 5.7 are generated from 40 experiments for the Benders cut

Table 5.6: Overall performance.

	KPI 7: Total Round-trip Time			KPI 9: Total Waiting Time			KPI 11: Port Departure Time		
	Max(%)	Min(%)	Avg(%)	Max(%)	Min(%)	Avg(%)	Max(%)	Min(%)	Avg(%)
Scenario 1	93.73	86.41	89.56	82.81	59.05	72.03	89.79	77.96	83.56
Scenario 2	89.47	80.97	84.31	70.21	20.54	50.71	78.56	70.94	74.35

Table 5.7: Performance of each vessel.

	KPI 8: Round-trip Time			KPI 10: Waiting Time		
	Max(%)	Min(%)	Avg(%)	Max(%)	Min(%)	Avg(%)
Scenario 1	114.60	66.17	90.01	127.95	34.13	76.43
Scenario 2	121.56	54.64	87.96	151.13	0	67.86

solution method and the benchmark method with centralized formulation, respectively. The values in Figure 5.7 are the ratio of objective values [3] between proposed method and the benchmark method. As we can see, in 99% of the experiments that are carried out for cases in Scenario 1, the Benders method generates better solutions than the benchmark method with the same runtime limits. Figure 5.7 also shows that the quality of solution improves with the increase of time. Moreover, as the Benders cut generated could effectively prevent the master problem from revisiting similar areas of the search space, it can also be seen that with the increase of run time, the Benders decomposition method has even better solutions than the benchmark method.

5.5.3 With coordination and without coordination

The overall logistical performance of the proposed solution method is reflected in KPIs 7, 9 and 11, as shown in Table 5.6. The individual logistical performance per vessel is reflected in KPIs 8 and 10, as show in Table 5.7. The percentages in Table 5.6 equal the values of KPIs 7, 9 and 11 from the proposed approach, divided by the KPIs 7, 9 and 11 obtained from situations in which there is no coordination between vessels. The percentages in Table 5.7 equal the values of KPIs 8 and 10 from the proposed approach, divided by the values of KPIs 8 and 10 from situations in which there is no coordination. In the settings, none of the inland vessels carry any ITT containers.

It can be seen from Table 5.6 and Table 5.7 that the proposed method generates rotation plans with shorter round-trip time and waiting time, as well as earlier port departure time compared with the no coordination situation. In Table 5.6, the vessels can spend on average 11% and 16% less total round-trip time in Scenario 1 and Scenario 2, respectively. In addition, they can also spend on average 28% and 50% less total waiting time in Scenario 1 and Scenario 2. Moreover, the last vessel can leave the port on average 17% and 26% earlier than no coordination situation in Scenario 1 and Scenario 2, respectively. Table 5.7 shows that each vessel can spend 10% and 13% less round-trip time, as well as 24% and 33% less waiting time in Scenario 1 and Scenario 2, respectively. This implies that inland vessels can spend less time in the port for transporting the same amount of hinterland containers using

[3] ratio of objective value $= \frac{\text{objective value from proposed method}}{\text{objective value from benchmark method}}$

Table 5.8: Impact of extra ITT containers (ITT) and extra space (ES) on the round-trip time per vessel (KPI 8) on average in Scenario 1 (S1) and Scenario 2 (S2).

	ITT=5%		ITT=10%		ITT=20%		ITT=30%	
	S1(%)	S2(%)	S1(%)	S2(%)	S1(%)	S2(%)	S1(%)	S2(%)
ES=5%	105.41	107.42	113.71	113.69	-	-	-	-
ES=10%	104.83	107.29	112.23	112.16	-	130.66	-	-
ES=20%	103.85	106.64	111.14	111.26	124.41	130.38	-	141.80
ES=30%	102.38	106.07	110.99	110.66	123.30	125.76	134.54	136.18

Table 5.9: Impact of extra ITT containers (ITT) and extra space (ES) on the total waiting time (KPI 9) on average in Scenario 1 (S1) and Scenario 2 (S2).

	ITT=5%		ITT=10%		ITT=20%		ITT=30%	
	S1(%)	S2(%)	S1(%)	S2(%)	S1(%)	S2(%)	S1(%)	S2(%)
ES=5%	98.62	110.68	119.79	135.41	-	-	-	-
ES=10%	91.54	105.66	123.52	120.20	-	147.69	-	-
ES=20%	82.49	109.77	122.66	116.16	149.06	198.20	-	224.88
ES=30%	81.37	109.56	119.66	113.06	145.40	158.26	146.11	158.39

Table 5.10: Impact of extra ITT containers (ITT) and extra space (ES) on the latest departure time of all the vessel (KPI 11) on average in Scenario 1 (S1) and Scenario 2 (S2).

	ITT=5%		ITT=10%		ITT=20%		ITT=30%	
	S1(%)	S2(%)	S1(%)	S2(%)	S1(%)	S2(%)	S1(%)	S2(%)
ES=5%	105.86	109.54	114.74	117.22	-	-	-	-
ES=10%	107.94	107.21	114.22	116.44	-	135.27	-	-
ES=20%	105.27	107.28	113.25	111.19	123.29	134.58	-	146.73
ES=30%	101.88	107.19	106.66	110.53	112.34	127.79	120.63	141.91

the proposed method.

5.5.4 Impact of extra vessel capacity and extra ITT containers

To investigate the potential of using inland vessels for transporting ITT containers, it is important to show to the vessel operators how much extra time they may spend in transporting different number of ITT containers, with different extra ship capacities. Therefore, experiments are carried out in Scenario 1 and Scenario 2. The vessels of each scenario together carry the numbers of ITT containers that equal to 5%, 10% and 20% of the sum of their mandatory hinterland container, when each vessel reserves 5%, 10% and 20% vessel capacity for these ITT containers, respectively.

Table 5.8 presents the impact on the round-trip time of each vessel (KPI 8) when it carries these ITT containers with different extra vessel capacity in both Scenario 1 and Scenario 2. Meanwhile, Table 5.9 and Table 5.10 also present the impact on the overall performance of these inland vessels, which refers to the total waiting time (KPI 9) and

the port departure time (KPI 11), when these vessels together carry different numbers of ITT containers in both scenarios. The values in these tables are the averages of all the experiments in each scenario. The percentages in Table 5.8 equal the values of the round-trip time that each vessel spends when it carries ITT containers, divided by the values of the round-trip time that it spends when it does not carry any ITT containers. The percentages in Tables 5.9 and 5.10 equal the values of the sum of waiting time and the port departure time of these vessels when they carry ITT containers, divided by the values of the sum of waiting time and the port departure time of these vessels when they do not carry any ITT containers, respectively.

Table 5.8 shows that in both scenarios, with the same amount of the ITT containers, the more extra capacities that is left on the vessels, the less time the vessels are likely to spend in the port. On the one hand, the vessels in Scenario 1 can spend on average less time than the vessels in Scenario 2 for carrying the same percentage of extra ITT containers with the same percentage of extra capacity in most of the cases. This implies that the inland vessels can spend less time for transporting the same percentage of extra ITT containers in less busy hours of the port. On the other hand, the vessels can cooperate to carry more ITT containers in Scenario 2 than in Scenario 1. As can be seen from Table 5.8, with extra 10% vessel capacity, the vessels in Scenario 1 cannot transport extra 20% ITT containers, while the vessels in Scenario 2 can carry extra 20% ITT container with additional 30% extra round-trip time spent per vessel on average. Moreover, with extra 20% vessel capacity, the vessels in Scenario 1 cannot transport extra 30% ITT containers, while the vessels in Scenario 2 can transport extra 30% ITT containers with additional 41% extra round-trip time spent per vessel on average. This implies that with more incoming vessels (busy hours), larger amount of extra ITT containers can be transported.

Table 5.9 shows that the vessels in Scenario 2 spent much longer time waiting than the vessels in Scenario 1. Table 5.10 shows the same trends with shorter latest departure time in Scenario 1 than in Scenario 2. This is also due to the fact that Scenario 2 includes more vessels, and these vessels will visit the same set of the terminals, the waiting at some terminals are inevitable.

5.5.5 Dealing with disturbances

The following 4 different disturbances are considered occurring during the transport of inland vessels between terminals:

- Disturbance 1: when the rotation plans have been executed for T hours, there is only one quay crane available at a number of J_1 terminals afterwards due to terminal strikes;
- Disturbance 2: when the rotation plans have been executed for T hours, a number of J_2 terminals will be unavailable for $T_2^{\text{unavailable}}$ hours for inland container vessels, due to delay of sea-going vessels;
- Disturbance 3: when the rotation plans have been executed for T hours, a number of J_3 terminals will be closed for T_3^{closed} due to extreme weather conditions;
- Disturbance 4: when the rotation plans have been executed for T hours, the containers from a number of J_4 terminal will only be available after $T_4^{\text{available}}$ hours due to delay of hinterland containers.

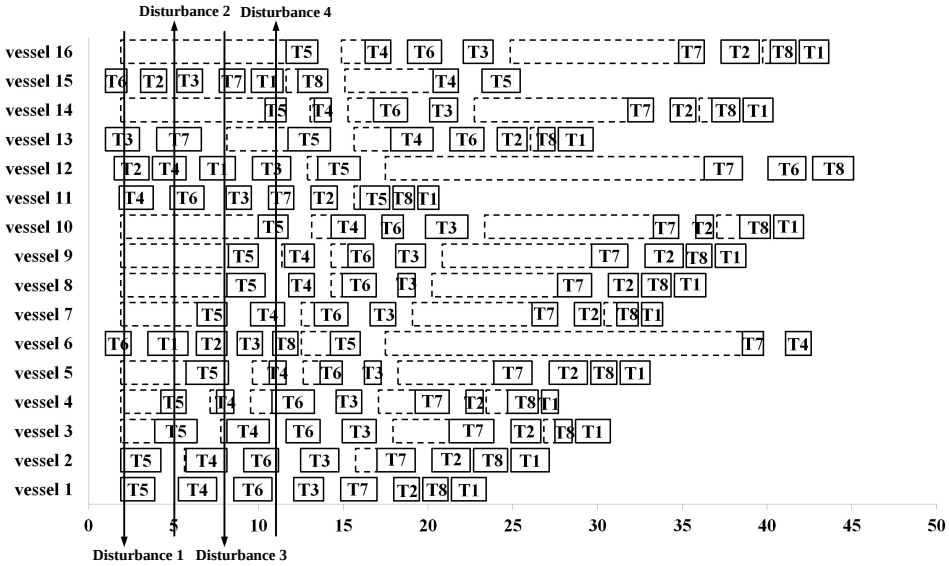

Figure 5.8: Time when disturbances happen.

To investigate the proposed approach's capability to handle these disturbances, this chapter setup 10 cases with Scenario 2 for each type of disturbance. In these cases, parameter J_1 varies from 4 to 6 terminals, $J_2 = 2$ varies from 2 to 4 terminals, J_3 varies from 3 to 4 terminals, and J_4 varies from 2 to 3 terminals. In addition, $T_2^{\text{unavailable}}$ varies from 6 to 8 hours, T_3^{closed} varies from 12 to 24 hours, and $T_4^{\text{available}}$ varies from 8 to 16 hours. In each case, these disturbances occur at the time which the rotation plans have been executed for $T = 2$, $T = 5$, $T = 8$ and $T = 11$ hours.

The overall and individual logistical performances of vessels' rotations after re-planning compared with the original chosen rotations, the results are presented in Table 5.11 and Table 5.12. The percentages in Table 5.11 equal the values of the total round-trip time, total waiting time and port departure time of the re-planned rotations, divided by the values of the total round-trip time, total waiting time and port departure time of the original chosen rotations. The percentages in Table 5.12 equal the values of the round-trip time and waiting time of each vessel with the re-planned rotations, divided by the values of the round-trip time and waiting time of each vessel with the original chosen rotations. The values in the tables are the averages of the experiments in all cases.

It can be seen from Table 5.11 and Table 5.12 that the re-planned rotations can make the vessels spend less round-trip time, waiting time and leave the port earlier, considering the four types of disturbances that happen at different times. This implies that whenever disturbances happen, which makes the previously chosen rotations no longer suitable, the proposed method is able to take consider these disturbances and re-plan the vessels' rotations accordingly.

As an example, Figure 5.8 shows the time when the four types of disturbances are occurring during the transport of inland vessels between terminals:

- Disturbance 1: when the rotation plans have been executed for 2 hours, there is only

Table 5.11: Overall performance of re-planned rotations compared with original rotations considering Disturbance 1 (D1), Disturbance 2 (D2), Disturbance 3 (D3) and Disturbance 4 (D4).

	KPI 7: Total round-trip time(%)				KPI 9: Total waiting time(%)				KPI 11: Port departure time(%)			
	T=2h	T=5h	T=8h	T=11h	T=2h	T=5h	T=8h	T=11h	T=2h	T=5h	T=8h	T=11h
D1	95.91	96.95	97.78	99.18	88.62	91.18	95.29	97.52	95.54	95.09	100.23	99.75
D2	97.13	97.49	98.16	97.75	91.88	92.92	92.94	92.91	97.31	99.34	97.21	97.34
D3	96.47	96.69	98.61	98.62	89.82	90.44	97.16	97.16	99.48	100.01	100.00	97.36
D4	94.91	96.79	97.08	99.82	86.67	89.26	91.88	99.94	93.78	97.45	97.01	98.73

Table 5.12: Performance of each vessel's re-planned rotation compared with original rotation considering Disturbance 1 (D1), Disturbance 2 (D2), Disturbance 3 (D3) and Disturbance 4 (D4).

	KPI 8: Round-trip time per vessel (%)				KPI 10: Waiting time per vessel (%)			
	T=2h	T=5h	T=8h	T=11h	T=2h	T=5h	T=8h	T=11h
D1	96.60	97.40	97.91	99.33	94.46	95.23	96.39	98.84
D2	97.37	97.68	98.32	97.86	95.64	99.75	95.58	94.27
D3	96.66	97.14	97.89	98.01	91.98	91.09	93.95	94.93
D4	97.24	97.02	97.25	100.00	88.76	90.70	93.61	100.00

one quay crane available at each terminal afterwards due to terminal strikes;

- Disturbance 2: when the rotation plans have been executed for 5 hours, from $T = 8$h to $T = 16$h terminal 3 and terminal 6 will be unavailable for inland container vessels due to delay of sea-going vessels;
- Disturbance 3: when the rotation plans have been executed for 8 hours, terminals 3, 7 and 8 will be closed from $T = 16$h to $T = 32$h due to extreme weather conditions;
- Disturbance 4: when the rotation plans have been executed for 11 hours, the containers from terminal 4 will only be available after $T = 15$h due to delay of hinterland containers.

Figures 5.9, 5.11, 5.13 and Figure 5.15 show the rotation plans before re-planning. Figures 5.10, 5.12, 5.14 and Figure 5.16 show the rotation plans after re-planning, the changes rotation plans are in gray color.

According to Figure 5.10, all vessels can spend on average 8% less time in the port after re-planning, in which vessel 6 and vessel 12 can even reduce 20% less round-trip time in the port. Figure 5.12 shows that all vessels can spend on average 6% less round-trip time in the port after re-planning. Figures 5.14 and 5.16 show that the vessels can reduce on average 4% and 3% round-trip time per vessel, respectively, after re-planning.

It can be seen that the reductions on round-trip time in Figures 5.14 and 5.16 are less than the reductions on round-trip time in Figures 5.10 and 5.12. This implies that larger improvements on rotation plans of vessels could be made if the disturbances happen earlier. The reason is that if the disturbances happens earlier, there is much room for improvements as the vessels can choose totally different rotations. For example, in Figure 5.16, the disturbance happens at $T = 11$h, and the vessels have visited most of the terminals, there are

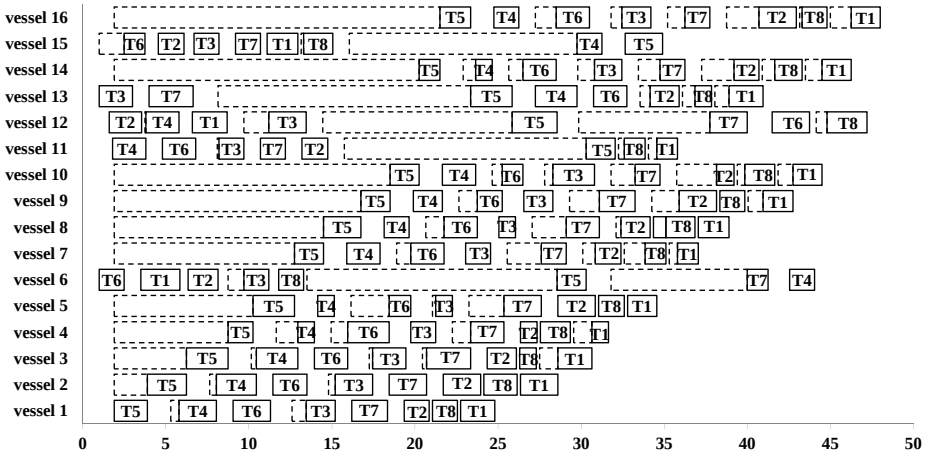

Figure 5.9: Implemented rotations with disturbance 1.

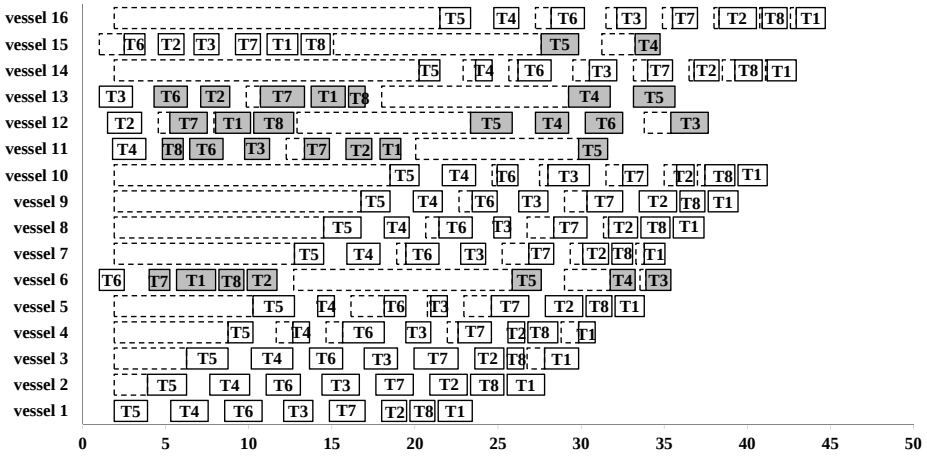

Figure 5.10: Implemented rotations with disturbance 1 after re-planning.

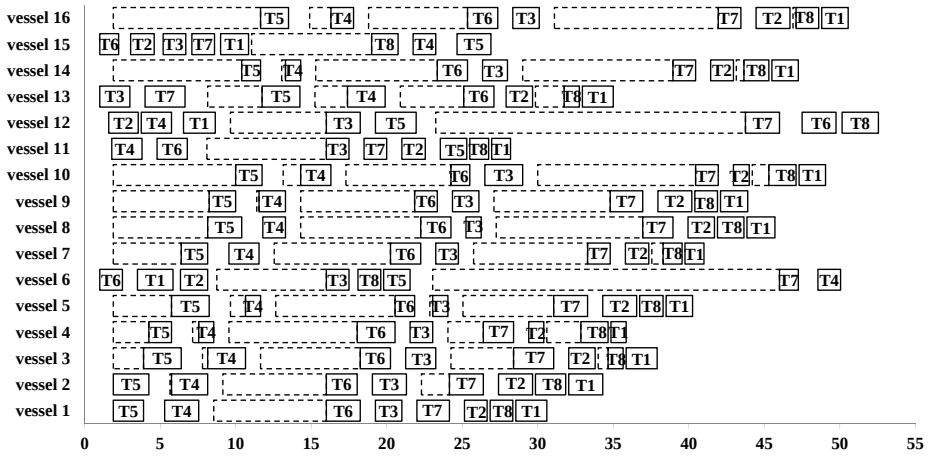

Figure 5.11: Implemented rotations with disturbance 2.

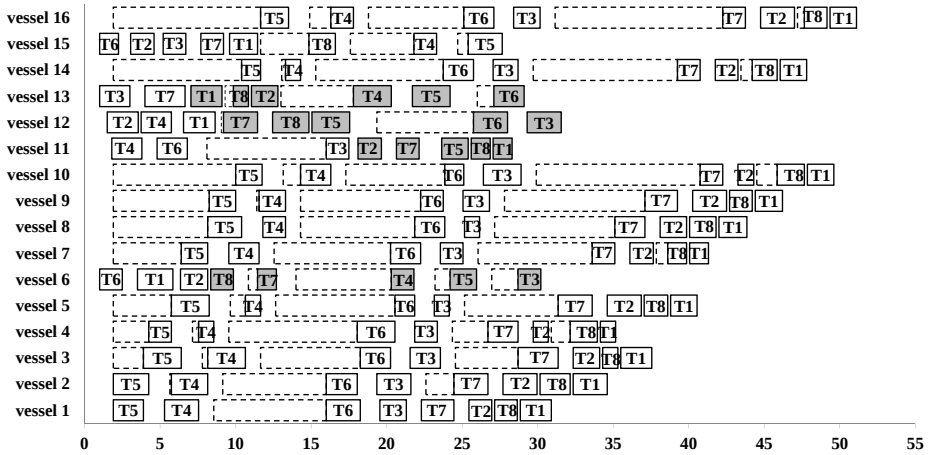

Figure 5.12: Implemented rotations with disturbance 2 after re-planning.

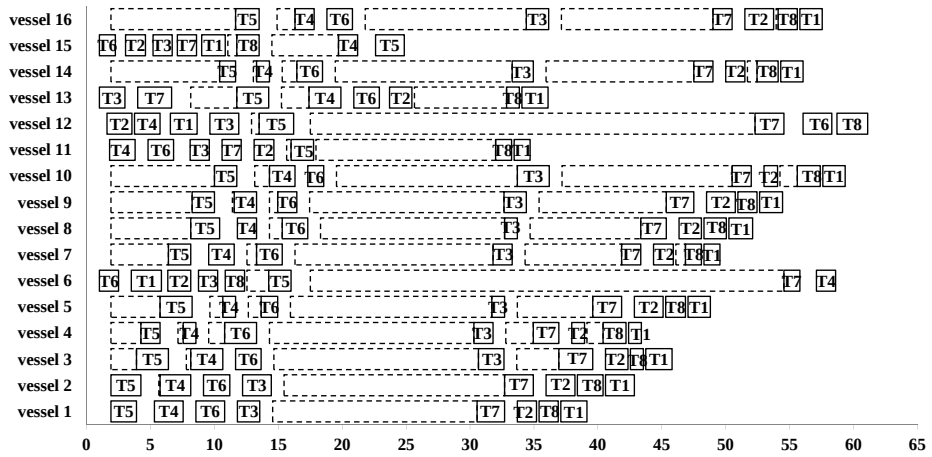

Figure 5.13: Implemented rotations with disturbance 3.

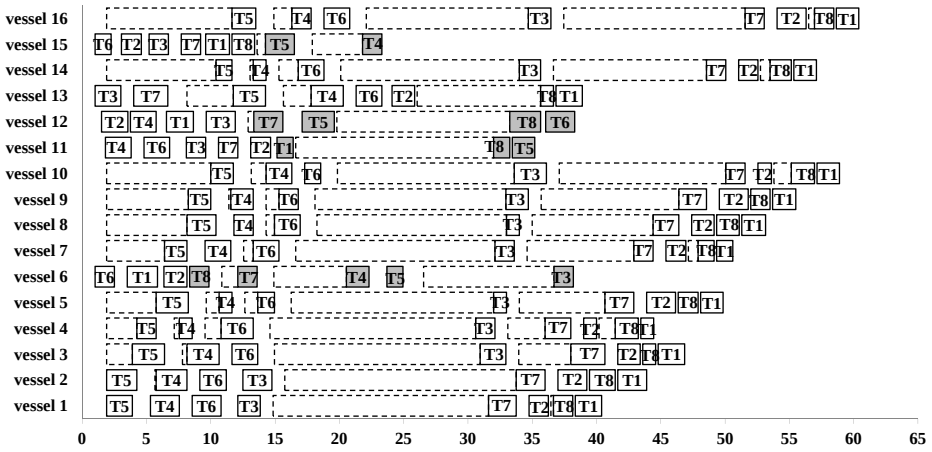

Figure 5.14: Implemented rotations with disturbance 3 after re-planning.

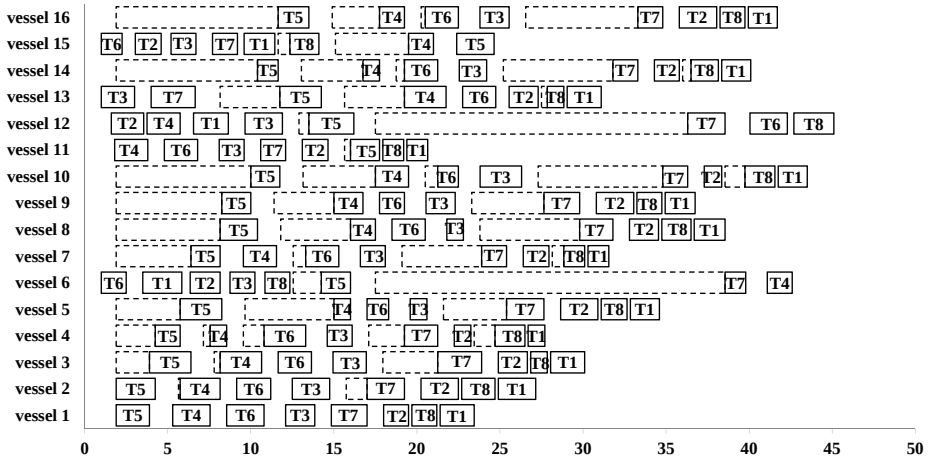

Figure 5.15: Implemented rotations with disturbance 4.

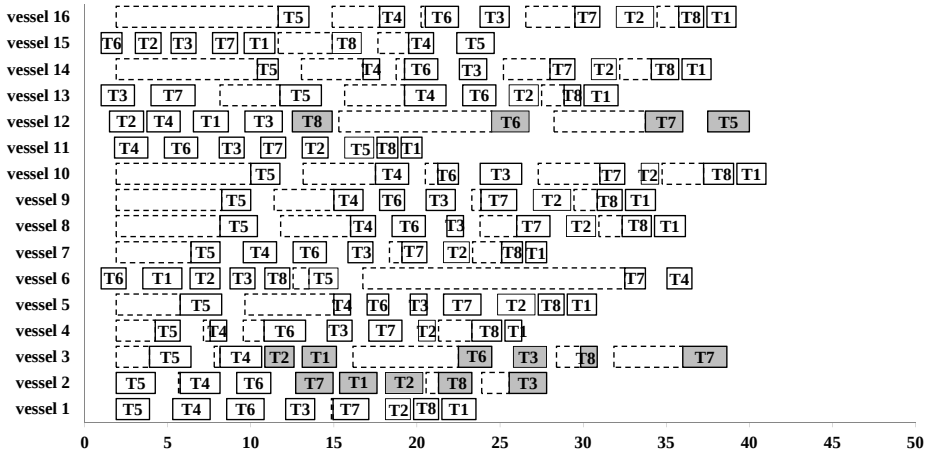

Figure 5.16: Implemented rotations with disturbance 4 after re-planning.

only 4 or 5 terminals left to visit. As we can also see from the original rotation in Figure 5.8, most of them will visit terminals 1, 2, 3, 7 and 8 in the end. Therefore, there is limited flexibility to change their rotations.

5.6 Conclusions

This chapter proposes a Benders decomposition-based method (M3.1) for tackling the vessel rotation planning in large ports, in which the problem is decomposed into a rotation generation master problem and a rotation evaluation subproblem. In this method, the vessels are coordinated in a fully-cooperative way with multi-level interactions. Comparing with the methods proposed in Chapters 3 and 4, method M3.1 can handle larger problem sizes and take into account possible disturbances that may happen in practice.

With the proposed method M3.1, vessel operators can decide whether they are willing to cooperate to take extra ITT containers based on the possible extra round-trip time and waiting time calculated. The port authority or terminal operators can also estimate how much extra ITT containers can be transported on the inland vessels by make use of their extra ship capacities. Moreover, method M3.1 is able to deal with four types of possible disturbances.

Based on the simulation results, it can be concluded that: firstly, method M3.1 generates better solutions (KPI 2) than a centralized formulation with the same computation time. This also implies that for smaller problem sizes, method M3.1 finds optimal solutions in shorter time than a centralized method. Secondly, with method M3.1, inland vessels can spend on average 12% less round-trip time (KPI 7), spend 30% less total waiting time (KPI 9), and spend 22% less port departure time (KPI 11) compared with the benchmark in which there is no coordination. Method M3.1 can also reduce on average 10% of the round-trip time and 28% of the waiting time per vessel (KPIs 8 and 10), respectively. Thirdly, whenever disturbances happen, the vessel operators could use method M3.1 to re-adjust their rotations based on the up-to-date information. This chapter answers partially the Key Research Questions 2 and 3.

Chapter 6

Conclusions and recommendations

This thesis aims to improve the reliability and efficiency of inland vessel operations in seaports. Three classes of automatic coordination methods have been proposed and investigated for use in various scenarios. This chapter first presents the main conclusions and answers the related research questions. Then remaining open questions are recommended for future research.

6.1 Conclusions

In this thesis, the following main research question was addressed: *how can different coordination methods be used to improve the reliability and efficiency of inland container vessel transport in seaports?* To answer this question, three classes of coordination methods were proposed: methods M1.1 and M1.2 were introduced in Chapter 3 to solve the coordination problem in small ports; methods M2.1, M2.2 and M2.3 were introduced in Chapter 4 to solve the problem in medium-sized ports; method M3.1 was introduced in Chapter 5 to solve the problem in large ports. In this thesis, a small port refers to the type of port that consists of 3-6 terminals; a medium-sized port refers to the type of port that consists of 6–10 terminals; a large port refers the type of port that consists of more than 10 terminals.

The characteristics of these coordination methods are categorized according to different features, as shown in Table 6. Methods M1.1 and M1.2 formulated the coordination problem using distributed constraint optimization (DCOP). There was no overall coordinator, and inland vessel agents and terminal agents were in equal positions. Therefore, the interactions among agents took place at a single-level. In addition, each agent did not reveal all information to the other agents. This implied that the vessel agents were lowly and partially cooperative. In both methods, the agents communicated with and sent information to each other, including currently chosen visiting sequences to terminals and the corresponding utility values, in a distributed way. Consequently, KPIs 1-6 were used to evaluate the communication and computation efficiency of the applied DCOP algorithms. While these methods ensure certain degree of information privacy, they also have the disadvantage that

M3.1	M2.3	M2.2	M2.1	M1.2	M1.1		
				✓	✓	Small	**Port size**
	✓	✓	✓			Medium	
✓						Large	
				✓	✓	Single level	**Interactions among agents**
✓	✓	✓	✓			Multi-levels	
	✓	✓	✓	✓	✓	Partially-cooperative	**Information sharing**
✓						Fully-cooperative	
				✓	✓	No	**Planning ITT containers**
	✓	✓	✓			Optional	
✓						Mandatory	
✓						Yes	**Disturbances**
	✓	✓	✓	✓	✓	No	
				✓	✓	Low	**Vessel cooperativeness**
	✓	✓	✓			Medium	
✓						High	
✓	✓	✓	✓	✓	✓	Currently chosen visiting sequences to terminals	**Exchanged information**
			✓	✓	✓	Utility values	
✓	✓	✓	✓			Possible arrival/departure time at each terminal	
✓						Round-trip time	
✓						Number of ITT containers	
				✓	✓	1 Computation time	**Key Performance Indicators**
✓	✓	✓	✓	✓	✓	2 Quality of solutions	
				✓	✓	3 Total number of messages	
				✓	✓	4 Number of messages sent/received per agent	
				✓	✓	5 Total amount of information	
				✓	✓	6 Amount of information sent/received per	
✓	✓	✓	✓			7 Total round-trip time	
✓						8 Round-trip time per vessel	
✓	✓	✓	✓			9 Total waiting time	
✓						10 Waiting time pre vessel	
✓	✓	✓	✓			11 Latest departure time	
					✓	Exact method	**Solution methods**
✓	✓	✓	✓	✓		Approximate method	
				✓	✓	Maximizing sum of utility values	**Optimization objective**
✓	✓	✓	✓			Minimizing the total round-trip time	
				✓	✓	Distributed constraint optimization	**Optimization techniques**
	✓	✓	✓			MIP solver	
		✓				CP solver	
✓	✓					Large neighbourhood search	
			✓			Coordination rules	
✓						Benders decomposition	

Table 6.1: Characteristics of the proposed coordination methods.

the overall solving process could be slower than a centralized scheme. This was because the distributed coordination scheme considered required a considerable amount of information exchange between terminal agents and vessel agents, and a distributed coordination scheme would involve a considerable amount of message exchange with the increase of problem sizes. For large-scale problems, a distributed scheme could cost a much longer time to find appropriate solutions. Therefore, methods M1.1 and M1.2 mainly apply for small-sized ports.

Methods M2.1, M2.2 and M2.3 consisted of two steps, including single vessel optimization and multiple vessel coordination. By splitting the solving process of the coordination problem into two steps, the problem size that can be solved was largely increased. Therefore, these methods can solve the coordination problem in larger ports than methods M1.1 and M1.2. An overall coordinator was only considered in the multiple vessel coordination, which sent and received information from local agents, and searched for global optimal solutions. Thus, the interactions among the agents took place at multi-levels. In single vessel optimization, the vessel agents locally determined their optimal solutions, and only revealed part of these solutions (possible arrival/departure time at each terminal) to the overall coordinator. Therefore, these vessel agents were coordinated in a medium and partially-cooperative way. As these methods are approximate solution methods, KPIs 2, 7, 9 and 11 were used to evaluate their solution qualities and their overall logistical performances.

A fully-cooperative coordination method M3.1 was proposed, in order to motivate the vessel operators to be more cooperative in planning their rotations in large seaports. To make this method applicable for even larger ports than methods M2.1, M2.2 and M2.3, Benders decomposition was used to decompose the coordination problem into a master problem and a sub-problem. This meant that the interactions among agents also took place at multi-levels. An overall coordinator was considered, and the vessel agents and terminal agents sent all the information this coordinator. This overall coordinator then searched for global optimal solutions for these agents. In addition, method M3.1 also considered mandatory ITT containers, as well as disturbances or accidents that may happen. As this method is also an approximate method, KPIs 2, 7, 8, 9, 10 and 11 were used to evaluate its solution quality and its overall and individual logistical performances.

Following the main research question, four sub-research questions were formulated in Chapter 1. The answers to the questions are as follows:

1. *What performance indicators should be used to evaluate the reliability and efficiency of inland vessel transport in seaports?*

   Chapter 2 defined the key performance indicators. The reliability of inland vessel transport is evaluated based on the deviations of the actually executed inland vessel schedules from their originally planned schedules. As these deviations are usually caused by the waiting times that the vessels spend at each terminal, shorter waiting time implies less deviation, which means that the vessel schedules are more reliable. Therefore, waiting time is an important performance indicator, and KPIs 9 (total waiting time) and 10 (waiting time per vessel) have been proposed to evaluate the reliability. The efficiency is evaluated based on the total time that inland vessels spend in the port for transporting containers, which is defined as the total round-trip time. The round-trip time of a vessel refers to the time it spends in the port for transporting containers. Shorter round-trip time implies that this vessel could accomplish its

transport tasks in a more efficiency way. Therefore, KPI 7 (total round-trip time), KPI 8 (round-trip time per vessel) and KPI 11 (latest departure time) have been proposed to evaluate the efficiency. To conclude, KPIs 7–11 are logistical KPIs that reflect the reliability and efficiency of inland vessel transport in seaports.

Besides these logistical KPIs, Chapter 2 also defined KPI 1 (computation time), KPI 2 (quality of solution), KPI 3 (total number of messages exchanged), KPI 4 (number of messages send/received per agent), KPI 5 (total amount of information) and KPI 6 (amount of information sent/received per agent) to evaluate the algorithmic performance of the proposed coordination methods in Chapters 3–5.

2. *To what extent can the inland vessels be better coordinated considering different levels of cooperativeness?*

   Chapter 1 identified two levels of cooperativeness based on the willingness of inland vessel operators to share information, including partially-cooperative and fully-cooperative. Partially-cooperative means that vessel operators only share part of the information with respect to the arrival and departure time at different terminals. Fully-cooperative means that vessel operators are willing to share all information to get better rotations. Chapter 3 proposed partially-cooperative coordination methods M1.1 and M1.2 with single-level interaction based on DCOP for small-sized ports. Experimental results show that methods M1.1 and M1.2 are able to find optimal rotation plans for inland vessels in small ports. For larger ports, these methods are not able to find optimal solutions within a reasonable amount of time. Chapter 4 proposed partially-cooperative coordination methods M2.1, M2.2 and M2.3 with multi-level interactions based on MIP, coordination rules, CP, and LNS-based heuristics, for medium-sized ports. Experimental results show that method M2.3 can reduce on average 8% of the total round-trip time (KPI 7), 23% of the total waiting time (KPI 9) and 10% of the port departure time (KPI 11) compared with the benchmark in which there is no coordination between vessels. Chapter 5 proposed a fully-cooperative coordination method M3.1 with multi-level interactions based on Benders decomposition and LNS for large ports. Experimental results show that this method can reduce on average 12% of the total round-trip time (KPI 7), and 30% of the total waiting time (KPI 9), and 22% of the port departure time (KPI 11) compared with the benchmark in which there is no coordination. This method can also reduce on average 10% of the round-trip time and 28% of the waiting time per vessel (KPIs 8 and 10), respectively.

3. *How can the planning of inland vessels contribute to inter-terminal transport (ITT) in large seaports?*

   As a potential solution for alleviating congestion of ITT on roads in large seaports, this thesis also investigated the possibility of using inland vessels to transport ITT containers, by making use of the available space on inland vessels when they are transporting between terminals in Chapters 4 and 5. In Chapter 4, each vessel agent was given the option to determine how many ITT containers it is willing to transport in the single vessel optimization problem by itself. This implies that each inland vessel agent determines locally how many ITT containers they are willing to transport, without cooperating with the other vessels. It is assumed that there will always be sufficient number of ITT containers to be transported. In Chapter 5, the inland vessels

were cooperating with each other to carry a certain number of ITT containers. Meanwhile, the vessel operators can also decide how much they are willing to cooperate by choosing how much space they have saved on-board specifically for carrying ITT containers. Based on this extra space that is made available on each inland vessel, the terminal operators can also estimate the number of extra ITT containers that are able to be transported by the incoming vessels.

4. *How can the proposed methods help practitioners making decisions?*

   This thesis proposed three classes of coordination methods for seaports with different sizes. For small-sized ports, Chapter 3 used four optimization algorithms in methods M1.1 and M1.2, including SyncBB, AFB, DPOP and MB-DPOP. For situations in which the vessel operators prefer good-quality rotation plans in a shorter time, layered SyncBB in method M1.2 is the best option, as it finds a good rotation plan quicker without the vessels having to spend long waiting times and a heavy communication load. For situations in which the vessel operators always prefer optimal rotation plans and long computation time is not an issue, non-layered DPOP in method M1.1 is the best option for small-sized problems, and non-layered MB-DPOP in method M1.1 is the best option for larger-sized problems with running memory limits. For situations in which a long computation time is acceptable, but the communication capacity is limited, both non-layered DPOP in M1.1 and layered DPOP in M1.2 are the best options, as these algorithms involve fewer number of messages and less amount of information exchanged.

   For medium-sized ports, Chapter 4 proposed several solution methods, and method M2.3 performs better than the other methods. Among the three types of LNS-based heuristics in method M2.3, if the vessel operators prefer to spend less time round-trip time in the port, then LNS-worst heuristic is the best option. If the vessel operators prefer to spend less waiting time in the port, then LNS-random heuristic is the best option. If the port authority prefer that the vessels can finish their transport tasks in a shorter time, then LNS-Shaw heuristic is the best option.

   For large-sized ports, Chapter 5 proposed a coordination method M3.1 in which vessel operators can decide whether they are willing to cooperate to take extra ITT containers based on the possible extra round-trip time and waiting time calculated. With this method the terminal operators can also estimate how much extra ITT containers could be transported by the incoming vessels during different times of a day, so that they can plan the ITT containers to be transported by inland vessels accordingly. Moreover, whenever real-time disturbances or accidents happen,vessel operators can adjust their rotation plans based on the up-to-date information.

To conclude, with these coordination methods, firstly, the vessel operators can choose to what extent they would like to be coordinated, either partially-cooperative (methods M1.1, M1.2, M2.1, M2.2 and M2.3) or fully-cooperative (method M3.1); secondly, terminal operators can also estimate how much time each inland vessel may spend at the terminals, in order to determine the schedules of terminal operations (methods M2.1, M2.2, M2.3 and M3.1); thirdly, whenever real-time disturbances or accidents occur, the previously determined rotations might be no longer optimal, method M3.1 can consider these disturbances

and generate new and better rotations for vessel operators based on the up-to-date information; fourthly, from the perspective of using inland vessels for inter-terminal transport, vessel operators can decide whether they are willing to cooperate to take extra ITT containers based on the possible extra round-trip times and waiting times that they may spend (methods M2.1, M2.2, M2.3 and M3.1); last but not least, terminal operators can also estimate how many extra inter-terminal containers could be transported by the incoming vessels during different times of a day, so that they can plan the inter-terminal containers that could be transported by inland vessels accordingly (method M3.1).

6.2 Recommendations for future research

To use the proposed methods for practical operations, firstly, an information platform for exchanging information among vessel operators and terminal operators is required. In addition, a decision support software also needs to be developed and installed on each vessel to send and receive messages to/from the information platform. Moreover, further investigations are also required, which are listed as future research directions. These future research directions are presented and discussed from two perspectives: the methodological perspective and the logistical application perspective.

Methodological perspective

- Extension of DCOP algorithms

  In the DCOP-based methods of Chapter 3, it is assumed that all the vessel agents stay fair and do not consider the situation of manipulation of certain agents by misreporting private information. To solve the issue of manipulation of certain agents, auction-like side payments could be introduced in the future research on the application of DCOP for VRPP. In [239], a faithful DCOP algorithm for solving efficient social choice problem with private information and self-interest is proposed. It is worth investigating how it could also be applied in VRPP. For example, payments could be added to the utility functions of each vessel agent. Then the utility function of a vessel agent is the original utility function minus the payment. The payment could be designed as consisting of the sum of the negative effect that this agent imposes on the other agents, in terms of its preferences on the solutions to the VRPP. Thus, no agent can improve its utility by misreporting its local information.

- Other potential solution methods

  This thesis mainly used solution methods from mathematical programming, constraint programming, and heuristic procedures. While these methods have achieved good performance, it is still worth studying the potential use of hybrids of metaheuristics (Tabu search, simulated annealing, ant colony optimization, ..., etc.) with other optimization techniques to further speed up the process to find optimal rotation plans for inland vessels.

- Extensive simulation study

The simulation experiments in this thesis are based on several virtual seaports, the relevant parameters of which are random but with reasonable estimations from literature. In addition, the mathematical models that are used in the thesis are simplified to some extent. To enhance the applicability of this thesis, a more detailed simulation study is important. If reliable historical data from certain seaports can be obtained, the settings of the experiments can be more practical. By implementing the proposed methods on a more practical simulation platform or system, the effectiveness of the proposed coordination methods can be verified.

Logistical application perspective

- Further enhanced coordination model

  While the inland vessels are usually full before they enter the port area, there might be some space available when they are traveling between terminals in the port. The coordination between inland vessels can be further enhanced by sharing part of the container transport volume and by making use of this type of available space on board. For example, if there is some extra space on inland vessel V_1 when it departs from terminal T_1 to terminal T_2, it can also transport the hinterland containers that are requested by vessel V_2 from terminal T_1 to terminal T_2, if vessel V_2 will visit terminal T_2 shortly after vessel V_1. In this way, vessel V_2 does not need to visit terminal T_1 but can directly travel to terminal T_2. This can increase the number of containers transported every time and decrease the number of visits of inland vessels to terminals, which can result in higher efficiency of the port and container terminals.

- More efficient inter-terminal transport

  To further investigate the potential of inland vessels for ITT, it is relevant to investigate in detail how the time-related constraints on ITT containers can be included. By keeping track of how long it takes for certain containers to travel from one terminal to another, it is easier to schedule the transport of ITT containers based on real-time requests.

- Implementation of coordination methods

  A decision support system can be an potential application of the proposed coordination methods. To transform these methods into a decision support system, an information platform is required. Assuming that each inland vessel has an installed decision support system that helps to plan its rotation, it is important to decide on the types of information needs to be sent to and received from the information platform. Moreover, in the process of information exchange, information loss or delay may happen. This would affect the execution of the proposed methods. Therefore, it is also important to investigate how to improve the tolerance of the coordination methods for information loss or delay.

- Integration with real-time control

  This thesis used the perspective of operational planning. To further improve the applicability of the proposed methods, it is recommended to integrate this operational planning level to the real-time control level. The real-time control of a inland vessel

includes speed selection, path following and tracking, collision avoidance, ..., etc. By integrating these two levels, more economic benefits can be gained. For example, the proposed coordination methods can give out the exact arrival time that a inland vessel should be at each terminal. Using this arrival time as a known information for the controller of a vessel, the vessel can schedule the most cost-efficient way to reach the destination terminal through speed selection, at the real-time control level.

- Integration with synchromodality

  This thesis mainly focused on the inland vessels in the port area, and the number of hinterland containers that they transported are assumed to be known in advance. To promote synchromodal transport application, it is also critical to integrate the planning of the inland vessels into the planning of synchromodal transport. This implies that the hinterland containers that are transported by inland vessels should be planned in combination with the containers that are transported by the other transportation modes.

Bibliography

[1] Abdel-Rahman, E. M., Nayfeh, A. H., and Masoud, Z. N. Dynamics and control of cranes: a review. *Journal of Vibration and Control*, 9(7):863–908, 2003.

[2] Agarwal, Y., Mathur, K., and Salkin, H. M. A set-partitioning-based exact algorithm for the vehicle routing problem. *Networks*, 19(7):731–749, 1989.

[3] Ai, T. J. and Kachitvichyanukul, V. A particle swarm optimisation for vehicle routing problem with time windows. *International Journal of Operational Research*, 6(4): 519–537, 2009.

[4] Ai, T. J. and Kachitvichyanukul, V. A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, 36(5):1693–1702, 2009.

[5] Alba, E. and Dorronsoro, B. Solving the vehicle routing problem by using cellular genetic algorithms. In *Evolutionary Computation in Combinatorial Optimization*, pages 11–20. Springer, 2004.

[6] Alba, E. and Dorronsoro, B. Computing nine new best-so-far solutions for capacitated VRP with a cellular genetic algorithm. *Information Processing Letters*, 98(6): 225–230, 2006.

[7] Alvarenga, G. B., Mateus, G. R., and De Tomi, G. A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows. *Computers & Operations Research*, 34(6):1561–1584, 2007.

[8] Aminu, U. and Eglese, R. W. A constraint programming approach to the chinese postman problem with time windows. *Computers & Operations Research*, 33(12): 3423–3431, 2006.

[9] Apt, K. *Principles of constraint programming*. Cambridge University Press, 2003.

[10] Archetti, C., Bianchessi, N., and Speranza, M. G. A column generation approach for the split delivery vehicle routing problem. *Networks*, 58(4):241–254, 2011.

[11] Archetti, C., Bouchard, M., and Desaulniers, G. Enhanced branch and price and cut for vehicle routing with split deliveries and time windows. *Transportation Science*, 45(3):285–298, 2011.

[12] Augerat, P., Belenguer, J., Benavent, E., Corberan, A., Naddef, D., and Rinaldi, G. Computational results with a branch and cut code for the capacitated vehicle routing problem. *Rapport de recherche- IMAG*, 1995.

[13] Bačkalov, I., Kalajdžić, M., and Hofman, M. Inland vessel rolling due to severe beam wind: a step towards a realistic model. *Probabilistic Engineering Mechanics*, 25(1): 18–25, 2010.

[14] Badeau, P., Guertin, F., Gendreau, M., Potvin, J.-Y., and Taillard, E. A parallel tabu search heuristic for the vehicle routing problem with time windows. *Transportation Research Part C: Emerging Technologies*, 5(2):109–122, 1997.

[15] Baker, B. M. and Ayechew, M. A genetic algorithm for the vehicle routing problem. *Computers & Operations Research*, 30(5):787–800, 2003.

[16] Baldacci, R., Hadjiconstantinou, E., and Mingozzi, A. An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. *Operations research*, 52(5):723–738, 2004.

[17] Baldacci, R., Toth, P., and Vigo, D. Recent advances in vehicle routing exact algorithms. *4OR*, 5(4):269–298, 2007.

[18] Baldacci, R., Bartolini, E., and Mingozzi, A. An exact algorithm for the pickup and delivery problem with time windows. *Operations research*, 59(2):414–426, 2011.

[19] Baldacci, R., Mingozzi, A., and Roberti, R. New route relaxation and pricing strategies for the vehicle routing problem. *Operations research*, 59(5):1269–1283, 2011.

[20] Baldacci, R., Mingozzi, A., and Roberti, R. Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal of Operational Research*, 218(1):1–6, 2012.

[21] Baños, R., Ortega, J., Gil, C., Fernández, A., and De Toro, F. A simulated annealing-based parallel multi-objective approach to vehicle routing problems with time windows. *Expert Systems with Applications*, 40(5):1696–1707, 2013.

[22] Barán, B. and Schaerer, M. A multi-objective ant colony system for vehicle routing problem with time windows. In *Applied Informatics*, pages 97–102, 2003.

[23] Barbarosoglu, G. and Ozgur, D. A tabu search algorithm for the vehicle routing problem. *Computers & Operations Research*, 26(3):255–270, 1999.

[24] Bard, J. F., Kontoravdis, G., and Yu, G. A branch-and-cut procedure for the vehicle routing problem with time windows. *Transportation Science*, 36(2):250–269, 2002.

[25] Beasley, J. E. Route-first-cluster-second methods for vehicle routing. *Omega*, 11(4): 403–408, 1983.

[26] Belmecheri, F., Prins, C., Yalaoui, F., and Amodeo, L. Particle swarm optimization algorithm for a vehicle routing problem with heterogeneous fleet, mixed backhauls, and time windows. *Journal of Intelligent Manufacturing*, 24(4):775–789, 2013.

[27] Benders, J. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962.

[28] Bent, R. and Van Hentenryck, P. A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. *Computers & Operations Research*, 33(4):875–893, 2006.

[29] Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., and Laporte, G. Static pickup and delivery problems: a classification scheme and survey. *TOP*, 15(1):1–31, 2007.

[30] Berger, J. and Barkaoui, M. A hybrid genetic algorithm for the capacitated vehicle routing problem. In *Genetic and Evolutionary Computation–GECCO 2003*, pages 646–656. Springer, 2003.

[31] Berger, J. and Barkaoui, M. A parallel hybrid genetic algorithm for the vehicle routing problem with time windows. *Computers & Operations Research*, 31(12):2037–2053, 2004.

[32] Bettinelli, A., Ceselli, A., and Righini, G. A branch-and-cut-and-price algorithm for the multi-depot heterogeneous vehicle routing problem with time windows. *Transportation Research Part C: Emerging Technologies*, 19(5):723–740, 2011.

[33] Bianchessi, N. and Righini, G. Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. *Computers & Operations Research*, 34(2): 578–594, 2007.

[34] Bierwirth, C. and Meisel, F. A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 202(3): 615–627, 2010.

[35] Bierwirth, C. and Meisel, F. A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 244(3):675 – 689, 2015.

[36] Bodin, L. D. and Berman, L. Routing and scheduling of school buses by computer. *Transportation Science*, 13(2):113–129, 1979.

[37] Bortfeldt, A. and Wäscher, G. Constraints in container loading a state-of-the-art review. *European Journal of Operational Research*, 229(1):1 – 20, 2013.

[38] Bradley, S., Hax, A., and Magnanti, T. *Applied Mathematical Programming*. Addison-Wesley Publishing Company, 1977.

[39] Brandão, J. Metaheuristic for the vehicle routing problem with time windows. In *Meta-Heuristics*, pages 19–36. Springer, 1999.

[40] Bräysy, O. A reactive variable neighborhood search for the vehicle-routing problem with time windows. *INFORMS Journal on Computing*, 15(4):347–368, 2003.

[41] Brito, J., Martínez, F. J., Moreno, J. A., and Verdegay, J. L. A GRASP–VNS hybrid for the fuzzy vehicle routing problem with time windows. In *Computer Aided Systems Theory-EUROCAST 2009*, pages 825–832. Springer, 2009.

[42] Brønmo, G., Christiansen, M., Fagerholt, K., and Nygreen, B. A multi-start local search heuristic for ship scheduling - a computational study. *Computers & Operations Research*, 34(3):900–917, 2007.

[43] Brønmo, G., Nygreen, B., and Lysgaard, J. Column generation approaches to ship scheduling with flexible cargo sizes. *European Journal of Operational Research*, 200 (1):139–150, 2010.

[44] Bullnheimer, B., Hartl, R. F., and Strauss, C. Applying the ant system to the vehicle routing problem. In *Meta-Heuristics*, pages 285–296. Springer, 1999.

[45] Bullnheimer, B., Hartl, R. F., and Strauss, C. An improved ant system algorithm for the vehicle routing problem. *Annals of Operations Research*, 89:319–328, 1999.

[46] Caricato, P., Ghiani, G., Grieco, A., and Guerriero, E. Parallel tabu search for a pickup and delivery problem under track contention. *Parallel Computing*, 29(5): 631–639, 2003.

[47] Caris, A. *Simulation and optimisation of intermodal barge transport networks*. PhD thesis, Hasselt University, May 2010.

[48] Carlo, H. J., Vis, I. F., and Roodbergen, K. J. Storage yard operations in container terminals: literature overview, trends, and research directions. *European Journal of Operational Research*, 235(2):412 – 430, 2014.

[49] Carpenter, C. J., Dugan, C. J., Kopena, J. B., Lass, R. N., Naik, G., Nguyen, D. N., Sultanik, E., Modi, P. J., and Regli, W. C. Intelligent systems demonstration: disaster evacuation support. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1964–1965, Vancouver, Canada, 2007.

[50] Carrabs, F., Cordeau, J.-F., and Laporte, G. Variable neighborhood search for the pickup and delivery traveling salesman problem with LIFO loading. *INFORMS Journal on Computing*, 19(4):618–632, 2007.

[51] Çatay, B. A new saving-based ant algorithm for the vehicle routing problem with simultaneous pickup and delivery. *Expert Systems with Applications*, 37(10):6809–6817, 2010.

[52] Chaovalitwongse, W., Kim, D., and Pardalos, P. M. GRASP with a new local search scheme for vehicle routing problems with time windows. *Journal of Combinatorial Optimization*, 7(2):179–207, 2003.

[53] Chen, A.-l., Yang, G.-k., and Wu, Z.-m. Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem. *Journal of Zhejiang University Science A*, 7(4):607–614, 2006.

[54] Chen, J.-F. and Wu, T.-H. Vehicle routing problem with simultaneous deliveries and pickups. *Journal of the Operational Research Society*, 57(5):579–587, 2006.

[55] Chen, P., Huang, H.-k., and Dong, X.-Y. Iterated variable neighborhood descent algorithm for the capacitated vehicle routing problem. *Expert Systems with Applications*, 37(2):1620–1627, 2010.

[56] Chen, Z.-L. and Xu, H. Dynamic column generation for dynamic vehicle routing with time windows. *Transportation Science*, 40(1):74–88, 2006.

[57] Cheng, C.-B. and Wang, K.-P. Solving a vehicle routing problem with time windows by a decomposition technique and a genetic algorithm. *Expert Systems with Applications*, 36(4):7758–7763, 2009.

[58] Chiang, W.-C. and Russell, R. A. Simulated annealing metaheuristics for the vehicle routing problem with time windows. *Annals of Operations Research*, 63(1):3–27, 1996.

[59] Christiansen, C. H. and Lysgaard, J. A branch-and-price algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research Letters*, 35 (6):773–781, 2007.

[60] Christiansen, M., Fagerholt, K., and Ronen, D. Ship routing and scheduling: Status and perspectives. *Transportation Science*, 38(1):1–18, 2004.

[61] Christiansen, M., Fagerholt, K., Nygreen, B., and Ronen, D. Maritime transportation. *Handbooks in Operations Research and Management Science*, 14:189–284, 2007.

[62] Christiansen, M., Fagerholt, K., Nygreen, B., and Ronen, D. Ship routing and scheduling in the new millennium. *European Journal of Operational Research*, 228 (3):467 – 483, 2013.

[63] Christofides, N. and Beasley, J. E. The period routing problem. *Networks*, 14(2): 237–256, 1984.

[64] Chu, Y. and Xia, Q. Generating Benders Cuts for a General Class of Integer Programming Problems. In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, volume 3011 of *Lecture Notes in Computer Science*, pages 127–141. Springer Berlin Heidelberg, 2004.

[65] Cordeau, J.-F., Laporte, G., Mercier, A., et al. A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational research society*, 52(8):928–936, 2001.

[66] Cortés, C. E., Matamala, M., and Contardo, C. The pickup and delivery problem with transfers: formulation and a branch-and-cut solution method. *European Journal of Operational Research*, 200(3):711–724, 2010.

[67] Créput, J.-C. and Koukam, A. The memetic self-organizing map approach to the vehicle routing problem. *Soft Computing*, 12(11):1125–1141, 2008.

[68] Czech, Z. J. and Czarnas, P. Parallel simulated annealing for the vehicle routing problem with time windows. In *The 10th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*, page 0376, Canary Islands, Spain, 2002. IEEE Computer Society.

[69] Dabia, S., Ropke, S., Van Woensel, T., and De Kok, T. Branch and price for the time-dependent vehicle routing problem with time windows. *Transportation Science*, 47 (3):380–396, 2013.

[70] Danna, E. and Le Pape, C. Branch-and-price heuristics: a case study on the vehicle routing problem with time windows. In *Column Generation*, pages 99–129. Springer, 2005.

[71] Dantzig, G. B. and Ramser, J. H. The truck dispatching problem. *Management Science*, 6(1):80–91, 1959.

[72] De Franceschi, R., Fischetti, M., and Toth, P. A new ILP-based refinement heuristic for vehicle routing problems. *Mathematical Programming*, 105(2-3):471–499, 2006.

[73] Debudaj-Grabysz, A. and Czech, Z. J. A concurrent implementation of simulated annealing and its application to the VRPTW optimization problem. In *Distributed and Parallel Systems*, pages 201–209. Springer, 2005.

[74] Dekker, R., Voogd, P., and van Asperen, E. Advanced methods for container stacking. In *Container Terminals and Cargo Systems*, pages 131–154. Springer, 2007.

[75] Dell'Amico, M., Righini, G., and Salani, M. A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection. *Transportation Science*, 40(2):235–247, 2006.

[76] DENG, A.-m., Chao, M., and ZHOU, Y.-t. Optimizing research of an improved simulated annealing algorithm to soft time windows vehicle routing problem with pick-up and delivery. *Systems Engineering-Theory & Practice*, 29(5):186–192, 2009.

[77] Derigs, U. and Döhmer, T. Indirect search for the vehicle routing problem with pickup and delivery and time windows. *OR Spectrum*, 30(1):149–165, 2008.

[78] Desaulniers, G. Branch-and-price-and-cut for the split-delivery vehicle routing problem with time windows. *Operations Research*, 58(1):179–192, 2010.

[79] Desaulniers, G., Lessard, F., and Hadjar, A. Tabu search, partial elementarity, and generalized k-path inequalities for the vehicle routing problem with time windows. *Transportation Science*, 42(3):387–404, 2008.

[80] Desrochers, M., Desrosiers, J., and Solomon, M. A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, 40(2):342–354, 1992.

[81] Desrosiers, J. and Dumas, Y. The shortest path problem for the construction of vehicle routes with pick-up, delivery and time constraints. In *Advances in Optimization and Control*, pages 144–157. Springer, 1988.

[82] Dethloff, J. Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up. *OR-Spektrum*, 23(1):79–96, 2001.

[83] Ding, Q., Hu, X., Sun, L., and Wang, Y. An improved ant colony optimization and its application to vehicle routing problem with time windows. *Neurocomputing*, 98: 101–107, 2012.

[84] Doerner, K. F., Hartl, R. F., Kiechle, G., Lucka, M., and Reimann, M. Parallel ant systems for the capacitated vehicle routing problem. In *Evolutionary Computation in Combinatorial Optimization*, pages 72–83. Springer, 2004.

[85] Donati, A. V., Montemanni, R., Casagrande, N., Rizzoli, A. E., and Gambardella, L. M. Time-dependent vehicle routing problem with a multi ant colony system. *European Journal of Operational Research*, 185(3):1174–1191, 2008.

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

[87] Douma, A., Schuur, P., and Schutten, J. Aligning barge and terminal operations using service-time profiles. *Flexible Services and Manufacturing Journal*, 23(4):385–421, 2011.

[88] Douma, A., van Hillegersberg, J., and Schuur, P. Design and evaluation of a simulation game to introduce a multi-agent system for barge handling in a seaport. *Decision Support Systems*, 53(3):465–472, 2012.

[89] Douma, A. M. *Aligning the operations of barges and terminals through distributed planning*. PhD thesis, University of Twente, Enschede, The Netherlands, Dec. 2008.

[90] Dror, M. Note on the complexity of the shortest path models for column generation in vrptw. *Operations Research*, 42(5):977–978, 1994.

[91] Dueck, G. New optimization heuristics: the great deluge algorithm and the record-to-record travel. *Journal of Computational Physics*, 104(1):86–92, 1993.

[92] Dueck, G. and Scheuer, T. Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing. *Journal of Computational Physics*, 90(1):161–175, 1990.

[93] Duhamel, C., Potvin, J.-Y., and Rousseau, J.-M. A tabu search heuristic for the vehicle routing problem with backhauls and time windows. *Transportation Science*, 31(1):49–59, 1997.

[94] Duhamel, C., Lacomme, P., Prins, C., and Prodhon, C. A GRASP× ELS approach for the capacitated location-routing problem. *Computers & Operations Research*, 37 (11):1912–1923, 2010.

[95] Duinkerken, M., Dekker, R., Kurstjens, S., Ottjes, J., and Dellaert, N. Comparing transportation systems for inter-terminal transport at the Maasvlakte container terminals. In Kim, K. and Gunther, H.-O., editors, *Container Terminals and Cargo Systems*, pages 37–61. Springer Berlin Heidelberg, 2007.

[96] Dumas, Y., Desrosiers, J., and Soumis, F. The pickup and delivery problem with time windows. *European Journal of Operational Research*, 54(1):7–22, 1991.

[97] Dumitrescu, I., Ropke, S., Cordeau, J.-F., and Laporte, G. The traveling salesman problem with pickup and delivery: polyhedral results and a branch-and-cut algorithm. *Mathematical Programming*, 121(2):269–305, 2010.

[98] European Commission. White paper on transport. *Roadmap to a single European Transport Area–towards a competitive and resource efficient transport system*, 2011.

[99] European Commission. Inland waterways. 2016. URL https://ec.europa.eu/transport/modes/inland/index_en.htm.

[100] European Commission. Strategic Research Agenda for Inland Waterway Transport. Technical report, European Commission, 2016. URL https://ec.europa.eu/programmes/horizon2020/en/smart-green-and-integrated-transport--work-programme-2016-2017.

[101] Fagerholt, K., Johnsen, T. A., and Lindstad, H. Fleet deployment in liner shipping: a case study. *Maritime Policy & Management*, 36(5):397–409, 2009.

[102] Farinelli, A., Rogers, A., Petcu, A., and Jennings, N. R. Decentralised coordination of low-power embedded devices using the max-sum algorithm. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multi-agent Systems-Volume 2*, pages 639–646. International Foundation for Autonomous Agents and Multiagent Systems, 2008.

[103] Favaretto, D., Moretti, E., and Pellegrini, P. Ant colony system for a vrp with multiple time windows and multiple visits. *Journal of Interdisciplinary Mathematics*, 10(2): 263–284, 2007.

[104] Fisher, M. L. and Jaikumar, R. A generalized assignment heuristic for vehicle routing. *Networks*, 11(2):109–124, 1981.

[105] Fisher, M. L., Jörnsten, K. O., and Madsen, O. B. Vehicle routing with time windows: Two optimization algorithms. *Operations Research*, 45(3):488–492, 1997.

[106] Fitzpatrick, S. and Meertens, L. An experimental assessment of a stochastic, anytime, decentralized, soft colourer for sparse graphs. In *Stochastic Algorithms: Foundations and Applications*, pages 49–64. Springer, 2001.

[107] Fleszar, K., Osman, I. H., and Hindi, K. S. A variable neighbourhood search algorithm for the open vehicle routing problem. *European Journal of Operational Research*, 195(3):803–809, 2009.

[108] Flood, M. M. The traveling-salesman problem. *Operations Research*, 4(1):61–75, 1956.

[109] Fukasawa, R., Longo, H., Lysgaard, J., de Aragão, M. P., Reis, M., Uchoa, E., and Werneck, R. F. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical Programming*, 106(3):491–511, 2006.

[110] Gajpal, Y. and Abad, P. An ant colony system for vehicle routing problem with simultaneous delivery and pickup. *Computers & Operations Research*, 36(12):3215–3223, 2009.

[111] Gajpal, Y. and Abad, P. L. Multi-ant colony system for a vehicle routing problem with backhauls. *European Journal of Operational Research*, 196(1):102–117, 2009.

[112] Gambardella, L. M., Taillard, É., and Agazzi, G. MACS-VRPTW: a multiple ant colony system for vehicle routing problems with time windows. 1999.

[113] Gelareh, S. and Meng, Q. A novel modeling approach for the fleet deployment problem within a short-term planning horizon. *Transportation Research Part E: Logistics and Transportation Review*, 46(1):76–89, 2010.

[114] Gendreau, M., Hertz, A., and Laporte, G. A tabu search heuristic for the vehicle routing problem. *Management Science*, 40(10):1276–1290, 1994.

[115] Gershman, A., Meisels, A., and Zivan, R. Asynchronous forward bounding for distributed COPs. *Journal of Artificial Intelligence Research*, 34(1):61–88, 2009.

[116] Ghaziri, H. Supervision in the self-organizing feature map: application to the vehicle routing problem. In *Meta-Heuristics*, pages 651–660. Springer, 1996.

[117] Ghoseiri, K. and Ghannadpour, S. F. Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm. *Applied Soft Computing*, 10(4):1096–1107, 2010.

[118] Goel, A. *Fleet telematics: real-time management and planning of commercial vehicle operations*, volume 40. Springer Science & Business Media, 2007.

[119] Goksal, F. P., Karaoglan, I., and Altiparmak, F. A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery. *Computers & Industrial Engineering*, 65(1):39–53, 2013.

[120] Gomes, L. and Von Zuben, F. J. A heuristic method based on unsupervised learning and fuzzy inference for the vehicle routing problem. In *Proceedings of the 2002 International Joint Conference on Neural Networks*, volume 2, pages 1930–1935. IEEE, 2002.

[121] Greenstadt, R., Grosz, B., and Smith, M. D. SSDPOP: improving the privacy of DCOP with secret sharing. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multi-agent systems*, pages 171–173, New York, USA, 2007. ACM.

[122] Gribkovskaia, I., Halskau, Ø., Laporte, G., and Vlček, M. General solutions to the single vehicle routing problem with pickups and deliveries. *European Journal of Operational Research*, 180(2):568–584, 2007.

[123] Guimarans, D., Herrero, R., Riera, D., Juan, A. A., and Ramos, J. J. Combining probabilistic algorithms, constraint programming and lagrangian relaxation to solve the vehicle routing problem. *Annals of Mathematics and Artificial Intelligence*, 62 (3-4):299–315, 2011.

[124] Gutiérrez-Jarpa, G., Desaulniers, G., Laporte, G., and Marianov, V. A branch-and-price algorithm for the vehicle routing problem with deliveries, selective pickups and time windows. *European Journal of Operational Research*, 206(2):341–349, 2010.

[125] Harati Mokhtari, A., Wall, A., Brooks, P., and Wang, J. Automatic Identification System (AIS): data reliability and human error implications. *Journal of Navigation*, 60(03):373–389, 2007.

[126] Harris, I., Wang, Y., and Wang, H. ICT in multimodal transport and technological trends: unleashing potential for the future. *International Journal of Production Economics*, 159:88 – 103, 2015.

[127] Hennig, F., Nygreen, B., and Lübbecke, M. E. Nested column generation applied to the crude oil tanker routing and scheduling problem with split pickup and split delivery. *Naval Research Logistics (NRL)*, 59(3-4):298–310, 2012.

[128] Hernández-Pérez, H., Rodríguez-Martín, I., and Salazar-González, J. J. A hybrid GRASP/VND heuristic for the one-commodity pickup-and-delivery traveling salesman problem. *Computers & Operations Research*, 36(5):1639–1645, 2009.

[129] Hirayama, K. and Yokoo, M. Distributed partial constraint satisfaction problem. In *Proceedings of the 3rd International Conference on Principles and Practice of Constraint Programming*, pages 222–236. Springer, Linz, Austria, 1997.

[130] Hooker, J. *Logic-based methods for optimization:combining optimization and constraint satisfaction*. John Wiley & Sons, 2000.

[131] Hooker, J. and Ottosson, G. Logic-based benders decomposition. *Mathematical Programming*, 96(1):33–60, 2003.

[132] Hooker, J. N. *Integrated methods for optimization*. Springer-Verlag New York, Secaucus, NJ, USA, 2006.

[133] Horst, M. R. and Langen, P. W. *Port Management*, chapter Coordination in hinterland transport chains: a major challenge for the seaport community, pages 57–83. Palgrave Macmillan UK, London, 2015.

[134] Hosny, M. I. and Mumford, C. L. The single vehicle pickup and delivery problem with time windows: intelligent operators for heuristic and metaheuristic algorithms. *Journal of Heuristics*, 16(3):417–439, 2010.

[135] Hosseini, S., Samaneh, and Basir, O. A. Target to sensor allocation: A hierarchical dynamic distributed constraint optimization approach. *Computer Communications*, 36(9):1024–1038, 2013.

[136] Hwang, H.-S. An improved model for vehicle routing problem with time constraint based on genetic algorithm. *Computers & Industrial Engineering*, 42(2):361–369, 2002.

[137] Hwang, H.-S., Visoldilokpun, S., and Rosenberger, J. M. A branch-and-price-and-cut method for ship scheduling with limited risk. *Transportation Science*, 42(3): 336–351, 2008.

[138] Imai, A., Nishimura, E., and Papadimitriou, S. The dynamic berth allocation problem for a container port. *Transportation Research Part B: Methodological*, 35(4):401–417, 2001.

[139] Imai, A., Sun, X., Nishimura, E., and Papadimitriou, S. Berth allocation in a container port: using a continuous location space approach. *Transportation Research Part B: Methodological*, 39(3):199–221, 2005.

[140] Ioachim, I., Gelinas, S., Soumis, F., and Desrosiers, J. A dynamic programming algorithm for the shortest path problem with time windows and linear node costs. *Networks*, 31(3):193–204, 1998.

[141] Jarl Eirik Korsvik, Kjetil Fagerholt, and Gilbert Laporte. A large neighbourhood search heuristic for ship routing and scheduling with split loads. *Computers & Operations Research*, 38(2):474 – 483, 2011.

[142] Jaszkiewicz, A. and Kominek, P. Genetic local search with distance preserving recombination operator for a vehicle routing problem. *European Journal of Operational Research*, 151(2):352–364, 2003.

[143] Jepsen, M., Petersen, B., Spoorendonk, S., and Pisinger, D. Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research*, 56 (2):497–511, 2008.

[144] Jepsen, M., Spoorendonk, S., and Ropke, S. A branch-and-cut algorithm for the symmetric two-echelon capacitated vehicle routing problem. *Transportation Science*, 47(1):23–37, 2013.

[145] Jung, S. and Haghani, A. Genetic algorithm for a pickup and delivery problem with time windows. *Transportation Research Record: Journal of the Transportation Research Board*, (1733):1–7, 2000.

[146] Junges, R. and Bazzan, A. L. Evaluating the performance of DCOP algorithms in a real world, dynamic problem. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multi-agent Systems*, pages 599–606, Estoril, Portugal, 2008.

[147] Kalantari, B., Hill, A. V., and Arora, S. R. An algorithm for the traveling salesman problem with pickup and delivery customers. *European Journal of Operational Research*, 22(3):377–386, 1985.

[148] Kallehauge, B., Larsen, J., and Madsen, O. B. Lagrangian duality applied to the vehicle routing problem with time windows. *Computers & Operations Research*, 33 (5):1464–1487, 2006.

[149] Karaoglan, I., Altiparmak, F., Kara, I., and Dengiz, B. The location-routing problem with simultaneous pickup and delivery: formulations and a heuristic approach. *Omega*, 40(4):465–477, 2012.

[150] Karimi, H. and Seifi, A. Acceleration of lagrangian method for the vehicle routing problem with time windows. *International Journal of Industrial Engineering*, 23(4): 309–315, 2012.

[151] Kim, K. and Günther, H.-O. *Container terminals and cargo systems*. Springer, 2006.

[152] Kobayashi, K. and Kubo, M. Optimization of oil tanker schedules by decomposition, column generation, and time-space network techniques. *Japan Journal of Industrial and Applied Mathematics*, 27(1):161–173, 2010.

[153] Kohl, N. and Madsen, O. B. An optimization algorithm for the vehicle routing problem with time windows based on lagrangian relaxation. *Operations Research*, 45(3): 395–406, 1997.

[154] Kohl, N., Desrosiers, J., Madsen, O. B., Solomon, M. M., and Soumis, F. 2-path cuts for the vehicle routing problem with time windows. *Transportation Science*, 33(1): 101–116, 1999.

[155] Kok, A. L., Meyer, C. M., Kopfer, H., and Schutten, J. M. J. A dynamic programming heuristic for the vehicle routing problem with time windows and European community social legislation. *Transportation Science*, 44(4):442–454, 2010.

[156] Kolen, A. W., Rinnooy Kan, A., and Trienekens, H. Vehicle routing with time windows. *Operations Research*, 35(2):266–273, 1987.

[157] Kolkman, J. Binnenvaart en containerlogistiek. Technical report, Ministerie van Verkeer en Waterstaat, 2009.

[158] Konings, R., Kreutzberger, E., and Maraš, V. Major considerations in developing a hub-and-spoke network to improve the cost performance of container barge transport in the hinterland: the case of the port of Rotterdam. *Journal of Transport Geography*, 29:63 – 73, 2013.

[159] Kontoravdis, G. and Bard, J. F. A GRASP for the vehicle routing problem with time windows. *ORSA journal on Computing*, 7(1):10–23, 1995.

[160] Kopetz, H. *Real-Time Systems: Design Principles for Distributed Embedded Applications*, chapter Internet of Things, pages 307–323. Springer US, Boston, MA, 2011.

[161] Korsvik, J. E. and Fagerholt, K. A tabu search heuristic for ship routing and scheduling with flexible cargo quantities. *Journal of Heuristics*, 16(2):117–137, 2010.

[162] Korsvik, J. E., Fagerholt, K., and Laporte, G. A tabu search heuristic for ship routing and scheduling. *Journal of the Operational Research Society*, 61(4):594–603, 2010.

[163] Kumar, A., Faltings, B., and Petcu, A. Distributed constraint optimization with structured resource constraints. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems*, pages 923–930, Budapest, Hungary, 2009.

[164] Kytöjoki, J., Nuortio, T., Bräysy, O., and Gendreau, M. An efficient variable neighborhood search heuristic for very large scale vehicle routing problems. *Computers & Operations Research*, 34(9):2743–2757, 2007.

[165] Landrieu, A., Mati, Y., and Binder, Z. A tabu search heuristic for the single vehicle pickup and delivery problem with time windows. *Journal of Intelligent Manufacturing*, 12(5-6):497–508, 2001.

[166] Lass, R. N., Kopena, J. B., Sultanik, E. A., Nguyen, D. N., Dugan, C. P., Modi, P. J., and Regli, W. C. Coordination of first responders under communication and resource constraints. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1409–1412, Estoril, Portugal, 2008.

[167] Léauté, T. *Distributed constraint optimization: privacy guarantees and stochastic uncertainty*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, Nov. 2011.

[168] Léauté, T. and Faltings, B. Coordinating logistics operations with privacy guarantees. In *Proceedings of the 22th International Joint Conference on Artificial Intelligence*, pages 2482–2487, Barcelona, Spain, 2011.

[169] Léauté, T., Ottens, B., and Szymanek, R. FRODO 2.0: An Open-Source Framework for Distributed Constraint Optimization. In *Proceedings of the 21th International Joint Conference on Artificial Intelligence*, pages 160–164, Pasadena, California, USA, 2009.

[170] Lee, C.-Y., Lee, Z.-J., Lin, S.-W., and Ying, K.-C. An enhanced ant colony optimization (EACO) applied to capacitated vehicle routing problem. *Applied Intelligence*, 32(1):88–95, 2010.

[171] Lee, L. H., Chew, E. P., Tan, K. C., and Han, Y. An optimization model for storage yard management in transshipment hubs. *Container Terminals and Cargo Systems*, pages 107–129, 2007.

[172] Lehmann, M., Grunow, M., and Günther, H.-O. Deadlock handling for real-time control of agvs at automated container terminals. *OR Spectrum*, 28(4):631–657, 2006.

[173] Lei, H., Laporte, G., and Guo, B. A generalized variable neighborhood search heuristic for the capacitated vehicle routing problem with stochastic service times. *Top*, 20 (1):99–118, 2012.

[174] Lesser, V., Ortiz, C. L., and Tambe, M. *Distributed sensor networks: A multiagent perspective*, volume 9. Kluwer Academic Pub, 2003.

[175] Letchford, A. N., Lysgaard, J., and Eglese, R. W. A branch-and-cut algorithm for the capacitated open vehicle routing problem. *Journal of the Operational Research Society*, 58(12):1642–1651, 2007.

[176] Li, C.-L. and Pang, K.-W. An integrated model for ship routing and berth allocation. *International Journal of Shipping and Transport Logistics*, 3(3):245–260, 2011.

[177] Li, H. and Lim, A. Local search with annealing-like restarts to solve the VRPTW. *European Journal of Operational Research*, 150(1):115–127, 2003.

[178] Li, S., Negenborn, R. R., and Lodewijks, G. A distributed constraint optimization approach for vessel rotation planning. In *Proceedings of the 5th International Conference on Computational Logistics*, pages 61–80, Valparaìso, Chile, Sept. 2014.

[179] Li, S., Negenborn, R. R., and Lodewijks, G. Vessel rotation planning: A layered distributed constraint optimization approach. In *Proceedings of the 7th International Conference on Agents and Artificial Intelligence*, pages 197–204, Lisbon, Portugal, Jan. 2015.

[180] Li, S., Negenborn, R. R., and Lodewijks, G. A two phase approach for inter-terminal transport of inland vessels using preference-based and utility-based coordination rules. In *Proceedings of the 6th International Conference on Computational Logistics*, pages 281–297, Delft, The Netherlands, September 2015.

[181] Li, S., Negenborn, R. R., and Lodewijks, G. Approach integrating mixed-integer programming and constraint programming for planning rotations of inland vessels in a large seaport. *Transportation Research Record: Journal of the Transportation Research Board*, (2549):1–8, 2016.

[182] Li, S., Negenborn, R. R., and Lodewijks, G. Planning inter-terminal transport for inland vessels in large seaports using a two-phase approach. *Submitted to a journal*, 2016.

[183] Li, S., Negenborn, R. R., and Lodewijks, G. Distributed constraint optimization for addressing vessel rotation planning problems. *Engineering Applications of Artificial Intelligence*, 48:159–172, 2016.

[184] Li, S., Negenborn, R. R., and Lodewijks, G. A logic-based benders decomposition approach to improve coordination of inland vessels for inter-terminal transport. In *Proceedings of the 7th International Conference on Computational Logistics*, page To appear, Lisbon, Portugal, September 2016.

[185] Li, S., Negenborn, R. R., and Lodewijks, G. Integrated MIP-CP approach for planning rotations of inland vessels in a large seaport. In *Proceedings of the Transportation Research Board 95th Annual Meeting*, Washington, D.C., United States, January 2016.

[186] Li, S., Negenborn, R. R., and Lodewijks, G. Closed-loop coordination of inland vessels operations in large seaports using hybrid logic-based benders decomposition. *Transportation Research Part E: Logistics and Transportation Review*, 97:1–21, 2017.

[187] Li, Z., Guo, S., Wang, F., and Lim, A. Improved GRASP with tabu search for vehicle routing with both time window and limited number of vehicles. In *Innovations in Applied Artificial Intelligence*, pages 552–561. Springer, 2004.

[188] Liberatore, F., Righini, G., and Salani, M. A column generation algorithm for the vehicle routing problem with soft time windows. *4OR*, 9(1):49–82, 2011.

[189] Lim, A. and Wang, F. A smoothed dynamic tabu search embedded grasp for m-VRPTW. In *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on*, pages 704–708. IEEE, 2004.

[190] Lin, C. A cooperative strategy for a vehicle routing problem with pickup and delivery time windows. *Computers & Industrial Engineering*, 55(4):766–782, 2008.

[191] Lin, D.-Y. and Liu, H.-Y. Combined ship allocation, routing and freight assignment in tramp shipping. *Transportation Research Part E: Logistics and Transportation Review*, 47(4):414–431, 2011.

[192] Lin, S.-W., Lee, Z.-J., Ying, K.-C., and Lee, C.-Y. Applying hybrid meta-heuristics for capacitated vehicle routing problem. *Expert Systems with Applications*, 36(2): 1505–1512, 2009.

[193] Lin, S.-W., Vincent, F. Y., and Lu, C.-C. A simulated annealing heuristic for the truck and trailer routing problem with time windows. *Expert Systems with Applications*, 38(12):15244–15252, 2011.

[194] Ling, Y., Jin, M., Hilliard, M. R., and Usher, J. M. A study of real-time identification and monitoring of barge-carried hazardous commodities. In *Proceedings of the 17th International Conference onGeoinformatics,*, pages 1–4. IEEE, 2009.

[195] Liu, X., Ye, H.-Q., and Yuan, X.-M. Tactical planning models for managing container flow and ship deployment. *Maritime Policy & Management*, 38(5):487–508, 2011.

[196] Lu, Q. and Dessouky, M. An exact algorithm for the multiple vehicle pickup and delivery problem. *Transportation Science*, 38(4):503–514, 2004.

[197] Lysgaard, J. and Wøhlk, S. A branch-and-cut-and-price algorithm for the cumulative capacitated vehicle routing problem. *European Journal of Operational Research*, 236(3):800–810, 2014.

[198] Lysgaard, J., Letchford, A. N., and Eglese, R. W. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100(2): 423–445, 2004.

[199] Maheswaran, R. T., Pearce, J. P., and Tambe, M. Distributed algorithms for DCOP: a graphical-game-based approach. In *Proceedings of the ISCA 17th International Conference on Parallel and Distributed Computing Systems*, pages 432–439, Francisco, California, USA, 2004.

[200] Maheswaran, R. T., Pearce, J. P., and Tambe, M. A family of graphical-game-based algorithms for distributed constraint optimization problems. In *Coordination of Large-scale Multi-agent Systems*, pages 127–146. Springer, 2006.

[201] Malliappi, F., Bennell, J. A., and Potts, C. N. A variable neighborhood search heuristic for tramp ship scheduling. In *International Conference on Computational Logistics*, pages 273–285. Springer, 2011.

[202] Marinakis, Y. Multiple phase neighborhood search-GRASP for the capacitated vehicle routing problem. *Expert Systems with Applications*, 39(8):6807–6815, 2012.

[203] Marinakis, Y. and Marinaki, M. Bumble bees mating optimization algorithm for the vehicle routing problem. In *Handbook of Swarm Intelligence*, pages 347–369. Springer, 2011.

[204] Marinakis, Y., Migdalas, A., and Pardalos, P. M. A new bilevel formulation for the vehicle routing problem and a solution method using a genetic algorithm. *Journal of Global Optimization*, 38(4):555–580, 2007.

[205] Marinakis, Y., Marinaki, M., and Dounias, G. A hybrid particle swarm optimization algorithm for the vehicle routing problem. *Engineering Applications of Artificial Intelligence*, 23(4):463–472, 2010.

[206] Masson, R., Lehuédé, F., and Péton, O. An adaptive large neighborhood search for the pickup and delivery problem with transfers. *Transportation Science*, 47(3):344–355, 2013.

[207] Mathematicl Programming Society. What is "mathematical programming"? 2016. URL http://www.mathprog.org/mps_whatis.htm.

[208] Mazzeo, S. and Loiseau, I. An ant colony algorithm for the capacitated vehicle routing. *Electronic Notes in Discrete Mathematics*, 18:181–186, 2004.

[209] Melis, M., Miller, I., Kentrop, M., Van Eck, B., Leenaarts, M., Schut, M., and Treur, J. Distributed rotation planning for container barges in the port of rotterdam. *Intelligent Logistics Concepts*, pages 101–116, 2003.

[210] Meng, Q. and Wang, T. A chance constrained programming model for short-term liner ship fleet planning problems. *Maritime Policy and Management*, 37(4):329–346, 2010.

[211] Mes, M. R. K. and Iacob, M.-E. *Logistics and supply chain innovation: bridging the gap between theory and practice*, chapter Synchromodal Transport Planning at a Logistics Service Provider, pages 23–36. Springer International Publishing, 2016.

[212] Mester, D., Bräysy, O., and Dullaert, W. A multi-parametric evolution strategies algorithm for vehicle routing problems. *Expert Systems with Applications*, 32(2): 508–517, 2007.

[213] Mladenović, N., Urošević, D., Ilić, A., et al. A general variable neighborhood search for the one-commodity pickup-and-delivery travelling salesman problem. *European Journal of Operational Research*, 220(1):270–285, 2012.

[214] Modares, A., Somhom, S., and Enkawa, T. A self-organizing neural network approach for multiple traveling salesman and vehicle routing problems. *International Transactions in Operational Research*, 6(6):591–606, 1999.

[215] Modi, P. J., Shen, W.-M., Tambe, M., and Yokoo, M. ADOPT: Asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence*, 161 (1):149–180, 2005.

[216] Mohammad M. Fazel Zarandi and Christopher Beck, J. Using logic-based benders decomposition to solve the capacity- and distance-constrained plant location problem. *INFORMS Journal on Computing*, 24(3):387–398, 2012.

[217] Monaco, M. F., Sammarra, M., and Sorrentino, G. The terminal-oriented ship stowage planning problem. *European Journal of Operational Research*, 239(1):256 – 265, 2014.

[218] Montané, F. A. T. and Galvao, R. D. A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers & Operations Research*, 33(3):595–619, 2006.

[219] Moonen, H., Van de Rakt, B., Miller, I., Van Nunen, J., and Van Hillegersberg, J. Agent technology supports inter-organizational planning in the port. Technical report, ERIM Report Series Reference No. ERS-2005-027-LIS, 2005.

[220] Mou, J. M., Van der Tak, C., and Ligteringen, H. Study on collision avoidance in busy waterways by using ais data. *Ocean Engineering*, 37(5):483–490, 2010.

[221] Moura, A., Oliveira, J., and Pimentel, C. A mathematical model for the container stowage and ship routing problem. *Journal of Mathematical Modelling and Algorithms in Operations Research*, 12(3):217–231, 2012.

[222] Naddef, D. and Rinaldi, G. Branch-and-cut algorithms for the capacitated vrp. In *The vehicle routing problem*, volume 9, pages 53–84. SIAM Monographs on Discrete Mathematics and Applications, 2001.

[223] Nanry, W. P. and Barnes, J. W. Solving the pickup and delivery problem with time windows using reactive tabu search. *Transportation Research Part B: Methodological*, 34(2):107–121, 2000.

[224] Negenborn, R. R. and Ocampo Martinez, C. *Transport of water versus transport over water: exploring the dynamic interplay of transport and water*, chapter Perspectives on Transport of Water versus Transport over Water, pages 1–10. Springer International Publishing, 2015.

[225] Newton, R. M. and Thomas, W. H. Bus routing in a multi-school system. *Computers & Operations Research*, 1(2):213–222, 1974.

[226] Nextlogic. Chain optimimalisation container barging. 2012. URL http://http://www.nextlogic.nl/.

[227] Notteboom, T. Chapter 2 Strategic challenges to container ports in a changing market environment. *Research in Transportation Economics*, 17:29 – 52, 2006.

[228] Notteboom, T. E. and Rodrigue, J.-P. Port regionalization: towards a new phase in port development. *Maritime Policy & Management*, 32(3):297–313, 2005.

[229] Osman, I. H. Meta-strategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, 41(4):421–451, 1993.

[230] Osman, I. H. and Kelly, J. P. *Meta-heuristics: theory and applications*. Springer Science & Business Media, 2012.

[231] Pankratz, G. A grouping genetic algorithm for the pickup and delivery problem with time windows. *OR Spectrum*, 27(1):21–41, 2005.

[232] Parragh, S. N., Doerner, K. F., and Hartl, R. F. A survey on pickup and delivery problems. *Journal für Betriebswirtschaft*, 58(2):81–117, 2008.

[233] Pearce, J. P. and Tambe, M. Quality guarantees on k-optimal solutions for distributed constraint optimization problems. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1446–1451, Hyderabad, India, 2007.

[234] Pecin, D., Pessoa, A., Poggi, M., and Uchoa, E. Improved branch-cut-and-price for capacitated vehicle routing. In *Integer Programming and Combinatorial Optimization*, pages 393–403. Springer, 2014.

[235] Pessoa, A., De Aragão, M. P., and Uchoa, E. Robust branch-cut-and-price algorithms for vehicle routing problems. In *The Vehicle Routing Problem: Latest Advances and New Challenges*, pages 297–325. Springer, 2008.

[236] Petcu, A. *A class of algorithms for distributed constraint optimization*. PhD thesis, École Polytechnique Fédérale de Lausanne, Lausanne, Switherland, 2009.

[237] Petcu, A. and Faltings, B. A scalable method for multiagent constraint optimization. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, volume 5, pages 266–271, Edingurgh, Scotland, 2005.

[238] Petcu, A. and Faltings, B. MB-DPOP: A new memory-bounded algorithm for distributed optimization. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, volume 7, pages 1452–1457, Hyderabad,India, 2007.

[239] Petcu, A., Faltings, B., and Parkes, D. C. M-DPOP: Faithful distributed implementation of efficient social choice problems. *Journal of Artificial Intelligence Research*, pages 705–755, 2008.

[240] Petering, M. E., Wu, Y., Li, W., Goh, M., and de Souza, R. Development and simulation analysis of real-time yard crane control systems for seaport container transshipment terminals. *OR spectrum*, 31(4):801–835, 2009.

[241] Pielage, B. J., Konings, R., Rijsenbrij, J., and Schuylenburg, M. Barge hub terminals: a perspective for more efficient hinterland container transport for the port of Rotterdam. In *Proceedings of the 48th Transportation Research Annual Forum*, pages 15–17, 2007.

[242] Pillac, V., Gendreau, M., Guéret, C., and Medaglia, A. L. A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1):1 – 11, 2013.

[243] Pisinger, D. and Ropke, S. Large neighborhood search. In *Handbook of Metaheuristics*, volume 146, pages 399–419. Springer US, 2010.

[244] Polacek, M., Hartl, R. F., Doerner, K., and Reimann, M. A variable neighborhood search for the multi depot vehicle routing problem with time windows. *Journal of Heuristics*, 10(6):613–627, 2004.

[245] Port of Antwerp Authority. Barge traffic system. 2014. URL http://www.portofantwerp.com/en/publications/product-sheets/barge-traffic-system-bts.

[246] Port of Rotterdam Authority. Container terminals and depots in port of Rotterdam. 2011. URL http://www.portofrotterdam.com/nl/Business/containers/Documents.

[247] Port of Rotterdam Authority. Container series. 2015. URL https://www.portofrotterdam.com/en/the-port/port-facts-and-figures/vessels.

[248] Port of Rotterdam Authority. Ship arrivals. 2015. URL https://www.portofrotterdam.com/en/the-port/port-facts-and-figures/vessels.

[249] Port of Rotterdam Authority. VTS services and VHF communication procedure port of Rotterdam. 2016. URL https://www.portofrotterdam.com/en/shipping/port-services/vts-services-and-vhf-communication-procedure-port-of-rotterdam.

[250] Port of Rotterdam Authority. InlandLinks. 2016. URL https://www.portofrotterdam.com/en/news-and-press-releases/inlandlinks.

[251] Potvin, J.-Y., Kervahut, T., Garcia, B.-L., and Rousseau, J.-M. The vehicle routing problem with time windows part I: tabu search. *INFORMS Journal on Computing*, 8 (2):158–164, 1996.

[252] Prescott-Gagnon, E., Desaulniers, G., and Rousseau, L.-M. A branch-and-price-based large neighborhood search algorithm for the vehicle routing problem with time windows. *Networks*, 54(4):190–204, 2009.

[253] Prins, C. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31(12):1985–2002, 2004.

[254] Prins, C. A GRASP× evolutionary local search hybrid for the vehicle routing problem. In *Bio-inspired algorithms for the vehicle routing problem*, pages 35–53. Springer, 2009.

[255] Qu, Y. and Bard, J. F. A GRASP with adaptive large neighborhood search for pickup and delivery problems with transshipment. *Computers & Operations Research*, 39 (10):2439–2456, 2012.

[256] Ralphs, T. K., Kopman, L., Pulleyblank, W. R., and Trotter, L. E. On the capacitated vehicle routing problem. *Mathematical Programming*, 94(2-3):343–359, 2003.

[257] Reghioui, M., Prins, C., and Labadi, N. GRASP with path relinking for the capacitated arc routing problem with time windows. In *Applications of Evolutionary Computing*, pages 722–731. Springer, 2007.

[258] Rego, C. Node-ejection chains for the vehicle routing problem: Sequential and parallel algorithms. *Parallel Computing*, 27(3):201–222, 2001.

[259] Rego, C. and Roucairol, C. A parallel tabu search algorithm using ejection chains for the vehicle routing problem. In *Meta-Heuristics*, pages 661–675. Springer, 1996.

[260] Reimann, M., Doerner, K., and Hartl, R. F. Analyzing a unified ant system for the VRP and some of its variants. In *Applications of Evolutionary Computing*, pages 300–310. Springer, 2003.

[261] Reimann, M., Doerner, K., and Hartl, R. F. D-ants: savings based ants divide and conquer the vehicle routing problem. *Computers & Operations Research*, 31(4): 563–591, 2004.

[262] Renaud, J., Boctor, F. F., and Ouenniche, J. A heuristic for the pickup and delivery traveling salesman problem. *Computers & Operations Research*, 27(9):905–916, 2000.

[263] Renaud, J., Boctor, F. F., and Laporte, G. Perturbation heuristics for the pickup and delivery traveling salesman problem. *Computers & Operations Research*, 29(9): 1129–1141, 2002.

[264] Ribeiro, G. M. and Laporte, G. An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. *Computers & Operations Research*, 39(3):728–735, 2012.

[265] Ronen, D. Cargo ships routing and scheduling: Survey of models and problems. *European Journal of Operational Research*, 12(2):119 – 126, 1983.

[266] Ropke, S. and Cordeau, J.-F. Branch-and-cut-and-price for the pickup and delivery problem with time windows. *Transportation Science*, 43(3):267–286, 2009.

[267] Ropke, S. and Pisinger, D. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4): 455–472, 2006.

[268] Ropke, S., Cordeau, J.-F., and Laporte, G. Models and branch-and-cut algorithms for pickup and delivery problems with time windows. *Networks*, 49(4):258–272, 2007.

[269] Rossi, F., Van Beek, P., and Walsh, T. *Handbook of constraint programming*. Elsevier, 2006.

[270] Rousseau, L.-M., Gendreau, M., Pesant, G., and Focacci, F. Solving VRPTWs with constraint programming based column generation. *Annals of Operations Research*, 130(1-4):199–216, 2004.

[271] Salani, M. and Vacca, I. Branch-and-price for the vehicle routing problem with discrete split deliveries and time windows. *European Journal of Operational Research*, 213(3):470–477, 2011.

[272] Santos, F. A., da Cunha, A. S., and Mateus, G. R. Branch-and-price algorithms for the two-echelon capacitated vehicle routing problem. *Optimization Letters*, 7(7): 1537–1547, 2013.

[273] Schroer, H., Corman, F., Duinkerken, M., Negenborn, R., and Lodewijks, G. Evaluation of inter terminal transport configurations at Rotterdam Maasvlakte using discrete event simulation. In *Proceedings of the Winter Simulation Conference 2014 (WSC'14)*, pages 1771–1782, Savannah, Georgia, Dec. 2014.

[274] Schulz, C. Efficient local search on the gpuinvestigations on the vehicle routing problem. *Journal of Parallel and Distributed Computing*, 73(1):14–31, 2013.

[275] Schulze, J. and Fahle, T. A parallel algorithm for the vehicle routing problem with time window constraints. *Annals of Operations Research*, 86:585–607, 1999.

[276] Schut, M., Kentrop, M., Leenaarts, M., Melis, M., and Miller, I. Approach: decentralised rotation planning for container barges. In *Proceedings of the 16th European Conference on Artificial Intelligence*, pages 755–759, Valencia, Spain, 2004.

[277] Sexton, T. R. and Choi, Y.-M. Pickup and delivery of partial loads with soft time windows. *American Journal of Mathematical and Management Sciences*, 6(3-4): 369–398, 1986.

[278] Shaw, P. Using constraint programming and local search methods to solve vehicle routing problems. In *Proceedings of the 4th International Conference on Principles and Practice of Constraint Programming*, pages 417–431, Pisa, Italy, 1998.

[279] Shih, C.-H., Huang, P.-H., Yamamura, S., and Chen, C.-Y. Design optimal control of ship maneuver patterns for collision avoidance: a review. *Journal of Marine Science and Technology*, 20(2):111–121, 2012.

[280] Sigurd, M., Pisinger, D., and Sig, M. Scheduling transportation of live animals to avoid the spread of diseases. *Transportation Science*, 38(2):197–209, 2004.

[281] Song, D. and Panayides, P. *Maritime Logistics: A Complete Guide to Effective Shipping and Port Management*. Kogan Page, 2012.

[282] Song, D. and Panayides, P. *Maritime Logistics: Contemporary Issues*. Emerald, 2012.

[283] Souza, M. J. F., Mine, M. T., Silva, M. d. S. A., Ochi, L. S., and Subramanian, A. A hybrid heuristic, based on iterated local search and genius, for the vehicle routing problem with simultaneous pickup and delivery. *International Journal of Logistics Systems and Management*, 10(2):142–157, 2011.

[284] Stahlbock, R. and Voß, S. Operations research at container terminals: a literature update. *OR Spectrum*, 30(1):1–52, 2008.

[285] Stahlbock, R. and Voß, S. Vehicle routing problems and container terminal operationsAn update of research. *The Vehicle Routing Problem: Latest Advances and New Challenges*, pages 551–589, 2008.

[286] Steenken, D., Voß, S., and Stahlbock, R. Container terminal operation and operations research–a classification and literature review. *OR Spectrum*, 26(1):3–49, 2004.

[287] Suarez, J. G. and Anticona, M. T. Two grasp metaheuristic for the capacitated vehicle routing problem considering split delivery and solving the three dimensional bin packing problem. *AISS*, 2(2):42–50, 2010.

[288] Subramanian, A., Drummond, L. M. d. A., Bentes, C., Ochi, L. S., and Farias, R. A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, 37(11):1899–1911, 2010.

[289] Subramanian, A., Uchoa, E., Pessoa, A. A., and Ochi, L. S. Branch-and-cut with lazy separation for the vehicle routing problem with simultaneous pickup and delivery. *Operations Research Letters*, 39(5):338–341, 2011.

[290] Subramanian, A., Uchoa, E., Pessoa, A. A., and Ochi, L. S. Branch-cut-and-price for the vehicle routing problem with simultaneous pickup and delivery. *Optimization Letters*, 7(7):1569–1581, 2013.

[291] Szeto, W., Wu, Y., and Ho, S. C. An artificial bee colony algorithm for the capacitated vehicle routing problem. *European Journal of Operational Research*, 215(1):126–135, 2011.

[292] Taillard, É., Badeau, P., Gendreau, M., Guertin, F., and Potvin, J.-Y. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science*, 31(2):170–186, 1997.

[293] Taillard, É. D., Gambardella, L. M., Gendreau, M., and Potvin, J.-Y. Adaptive memory programming: a unified view of metaheuristics. *European Journal of Operational Research*, 135(1):1–16, 2001.

[294] Tan, K., Lee, T., Ou, K., and Lee, L. A messy genetic algorithm for the vehicle routing problem with time window constraints. In *The proceedings of the 2001 Congress on Evolutionary Computation*, volume 1, pages 679–686. IEEE, 2001.

[295] Tan, X., Zhuo, X., and Zhang, J. Ant colony system for optimizing vehicle routing problem with time windows (VRPTW). In *Computational Intelligence and Bioinformatics*, pages 33–38. Springer, 2006.

[296] Tarantilis, C., Zachariadis, E., and Kiranoudis, C. A guided tabu search for the heterogeneous vehicle routeing problem. *Journal of the Operational Research Society*, pages 1659–1699, 2008.

[297] Tarantilis, C. D. Solving the vehicle routing problem with adaptive memory programming methodology. *Computers & Operations Research*, 32(9):2309–2327, 2005.

[298] Tarantilis, C. D. and Kiranoudis, C. T. Boneroute: an adaptive memory-based method for effective fleet management. *Annals of Operations Research*, 115(1-4):227–241, 2002.

[299] Tarantilis, C. D., Kiranoudis, C. T., and Vassiliadis, V. S. A list based threshold accepting algorithm for the capacitated vehicle routing problem. *International Journal of Computer Mathematics*, 79(5):537–553, 2002.

[300] Taş, D., Gendreau, M., Dellaert, N., Van Woensel, T., and De Kok, A. Vehicle routing with soft time windows and stochastic travel times: a column generation and branch-and-price solution approach. *European Journal of Operational Research*, 236(3): 789–799, 2014.

[301] Tavakkoli-Moghaddam, R., Safaei, N., and Gholipour, Y. A hybrid simulated annealing for capacitated vehicle routing problems with the independent route length. *Applied Mathematics and Computation*, 176(2):445–454, 2006.

[302] Tavakkoli-Moghaddam, R., Safaei, N., Kah, M., and Rabbani, M. A new capacitated vehicle routing problem with split service for minimizing fleet cost by simulated annealing. *Journal of the Franklin Institute*, 344(5):406–425, 2007.

[303] Tavakkoli-Moghaddam, R., Gazanfari, M., Alinaghian, M., Salamatbakhsh, A., and Norouzi, N. A new mathematical model for a competitive vehicle routing problem with time windows solved by simulated annealing. *Journal of Manufacturing Systems*, 30(2):83–92, 2011.

[304] Terminals, E. C. The future of freight transport – ECT's vision on sustainable and reliable European transport. Technical report, Europe Container Terminals, 2011. URL http://www.informatie.binnenvaart.nl/.

[305] Thangiah, S. R. A hybrid genetic algorithms, simulated annealing and tabu search heuristic for vehicle routing problems with time windows. *Practical Handbook of Genetic Algorithms*, 3:347–381, 1999.

[306] Thangiah, S. R., Nygard, K. E., and Juell, P. L. Gideon: a genetic algorithm system for vehicle routing with time windows. In *Proceedings of the 7th IEEE Conference on Artificial Intelligence Applications*, volume 1, pages 322–328. IEEE, 1991.

[307] Tierney, K., Voß, S., and Stahlbock, R. A mathematical model of inter-terminal transportation. *European Journal of Operational Research*, 235(2):448–460, 2014.

[308] Ting, C.-J. and Chen, C.-H. Combination of multiple ant colony system and simulated annealing for the multidepot vehicle-routing problem with time windows. *Transportation Research Record: Journal of the Transportation Research Board*, (2089):85–92, 2008.

[309] Toth, P. and Tramontani, A. An integer linear programming local search for capacitated vehicle routing problems. In *The Vehicle Routing Problem: Latest Advances and New Challenges*, pages 275–295. Springer, 2008.

[310] Toth, P. and Vigo, D. The granular tabu search and its application to the vehicle-routing problem. *Informs Journal on Computing*, 15(4):333–346, 2003.

[311] Vakhutinsky, A. I. and Golden, B. L. Solving vehicle routing problems using elastic nets. In *1994 IEEE International Conference on Neural Networks*, volume 7, pages 4535–4540. IEEE World Congress on Computational Intelligence, 1994.

[312] Van Breedam, A. Comparing descent heuristics and metaheuristics for the vehicle routing problem. *Computers & Operations Research*, 28(4):289–315, 2001.

[313] Van der Bruggen, L., Lenstra, J. K., and Schuur, P. Variable-depth search for the single-vehicle pickup and delivery problem with time windows. *Transportation Science*, 27(3):298–311, 1993.

[314] Verdonck, L., Caris, A., Ramaekers, K., and Janssens, G. K. Analysis of the operations of an intermodal barge terminal. *International Journal of Simulation and Process Modelling*, 9(1-2):3–15, 2014.

[315] Vincent, F. Y. and Lin, S.-W. Multi-start simulated annealing heuristic for the location routing problem with simultaneous pickup and delivery. *Applied Soft Computing*, 24: 284–290, 2014.

[316] Wang, H.-F. and Chen, Y.-Y. A genetic algorithm for the simultaneous delivery and pickup problems with time window. *Computers & Industrial Engineering*, 62(1): 84–95, 2012.

[317] Weiss, G. *Multi-agent Systems (2nd edition)*. MIT Press, 2013.

[318] Wheatley, D., Fatma, G., and Jewkes, E. Logic-based Benders decomposition for an inventory-location problem with service constraints. *Omega*, 55:10–23, 2015.

[319] Williams, H. P. *Model building in mathematical programming*. John Wiley & Sons, 2013.

[320] Wooldridge, M. *An introduction to multi-agent systems*. John Wiley & Sons, 2009.

[321] Xu, J. and Kelly, J. P. A network flow-based tabu search heuristic for the vehicle routing problem. *Transportation Science*, 30(4):379–393, 1996.

[322] Xu, J., Yan, F., and Li, S. Vehicle routing optimization with soft time windows in a fuzzy random environment. *Transportation Research Part E: Logistics and Transportation Review*, 47(6):1075–1091, 2011.

[323] Yeo, G.-T. and Song, D.-W. An application of the hierarchical fuzzy process to container port competition: policy and strategic implications. *Transportation*, 33(4): 409–422, 2006.

[324] Yu, B. and Yang, Z. Z. An ant colony optimization model: The period vehicle routing problem with time windows. *Transportation Research Part E: Logistics and Transportation Review*, 47(2):166–181, 2011.

[325] Yu, B., Yang, Z.-Z., and Yao, B. An improved ant colony optimization for vehicle routing problem. *European Journal of Operational Research*, 196(1):171–176, 2009.

[326] Zachariadis, E. E. and Kiranoudis, C. T. A local search metaheuristic algorithm for the vehicle routing problem with simultaneous pick-ups and deliveries. *Expert Systems with Applications*, 38(3):2717–2726, 2011.

[327] Zhang, T., TIAN, W.-x., ZHANG, Y.-j., and LIU, S.-x. Improved ant colony system for VRPSPD with maximum distance constraint. *Systems Engineering-Theory & Practice*, 28(1):132–140, 2008.

[328] Zhang, T., Chaovalitwongse, W. A., and Zhang, Y. Integrated ant colony and tabu search approach for time dependent vehicle routing problems with simultaneous pickup and delivery. *Journal of Combinatorial Optimization*, 28(1):288–309, 2014.

[329] Zhao, F., Li, S., Sun, J., and Mei, D. Genetic algorithm for the one-commodity pickup-and-delivery traveling salesman problem. *Computers & Industrial Engineering*, 56(4):1642–1648, 2009.

[330] Zhao, Y. W., Wu, B., Wang, W., Ma, Y. L., Wang, W., and Sun, H. Particle swarm optimization for vehicle routing problem with time windows. In *Materials Science Forum*, volume 471, pages 801–805. Trans Tech Publ, 2004.

[331] Zhen, L., Jiang, X., Lee, L. H., and Chew, E. P. A review on yard management in container terminals. *Industrial Engineering & Management Systems*, 12(4):289–304, 2013.

[332] Zhu, Q., Qian, L., Li, Y., and Zhu, S. An improved particle swarm optimization algorithm for vehicle routing problem with time windows. In *IEEE Congress on Evolutionary Computation, 2006.*, pages 1386–1390. IEEE, 2006.

[333] Zivan, R., Glinton, R., and Sycara, K. Distributed constraint optimization for large teams of mobile sensing agents. In *Proceedings of the 2009 International Conference on Web Intelligence and Intelligent Agent Technology*, pages 347–354, Milano, Italy, 2009.

Glossary

List of symbols and notations

Below follows a list of the most frequently used symbols and notations in this thesis.

a_i^m	arrival time steps of vessel m at terminal i
a_{mk}	arrival time of vessel m on segment k after coordination
$\mathcal{A}$	set of vessel agents and terminal agents
A_m	vessel agent m
A_m^{as}	vessel agent m in the assignment problem
A_m^{sc}	vessel agent m in the scheduling problem
A_{kp}^m	arrival time of vessel m on segment k for its rotation plan p
b_i^m	number of containers on-board of vessel m when it departs from terminal i
B_i	terminal agent i
B_i^{as}	terminal agent i in the assignment problem
B_i^{sc}	terminal agent i in the scheduling problem
C_i	number of vessels that can be handled by terminal i simultaneously
C_m^{capacity}	carrying capacity of vessel m in TEU
C_m^{original}	initial number of containers on vessel m before entering the port in TEU
d_i^m	departure steps of vessel m at terminal i
d_{mk}	departure time of vessel m on segment k after coordination
D	set of discrete time slots
D_s	set of discrete time steps
$\mathcal{D}_m$	set of finite variable domains
$\mathcal{D}_m^{\text{as}}$	set of finite variable domains in assignment problem
$\mathcal{D}_m^{\text{sc}}$	set of finite variable domains in the scheduling problem
D_{kp}^m	departure time of vessel m on segment k for its rotation plan p
$e_{mm'k}$	binary variable, $e_{mm'k} = 1$ if the vessel m' arrives at terminal j earlier than vessel m; otherwise $e_{mm'k} = 0$
f_{jkr}^q	departure time at quay q of terminal j on segment k of the vessel with rank r

g^q_{jkr}	possible starting time at quay q for vessels that will visit terminal j on segment k with the ranking of r
h^q_{jk}	latest departure time at q of terminal j on segment k
I_m	rotation plan that vessel m chooses
k^m_i	the segment at which vessel m visits terminal i
$K^{\max}_m$	number of segments in the rotation plan of vessel m
l^m_i	number of hinterland containers that need to be loaded by vessel m at terminal i
M	set of vessels entering the port
$M^{+\infty}$	a very large positive number
N_m	set of terminal that vessel m needs to visit in the port
P_m	set of candidate rotation plans in the solution pool of vessel m
Q_i	set of quays in terminal i
r_{i0}	utility function of terminal i
$r_{m1} - r_{m7}$	utility functions of vessel m
r^{ia}_1	inter-agent utility function
r_{mk}	ranking of the arrival time of vessel m on segment k
r^{master}_{mi}	ranking of the arrival time of vessel m at terminal i in the master problem
r^{sub}_{mi}	ranking of the arrival time of vessel m at terminal i in the subproblem
R^m_{ij}	number of inter-terminal containers that need to be transported from terminal i to j by vessel m
$\mathcal{R}_m$	set of utility functions
$\mathcal{R}^{\text{ia}}$	set of inter-agent utility functions
$\mathcal{R}^{\text{as}}_m$	set of utility functions in the assignment problem
$\mathcal{R}^{\text{sc}}_m$	set of utility functions in the scheduling problem
s^m_i	service time of vessel m at terminal i
s_{mk}	service time of vessel m on segment k
$[S^j_q, E^j_q]$	service time window of the sea vessel at quay q of terminal j
t^{load}_i	average loading time, per loaded container at terminal i
t^{unload}_i	average unloading time, per unloaded container at terminal i
t_{mk}	terminal vessel m visits on segment k after coordination
T^{entrance}_i	traveling time from the entrance of the port to terminal i
T^{travel}_{ij}	traveling time for a vessel from terminal i to j
$[T^{k-1}_i, T^k_i]$	a fixed time window during which vessels can visit terminal i

	during time slot k
T_{kp}^{m}	terminal that is on segment k of vessel m's rotation plan p
$T_{jq}^{\text{departure}}$	latest departure time of the vessels being served at quay q of terminal j
T_{i}^{deadline}	deadline for inter-terminal containers to be delivered to terminal i
u_{i}^{m}	number of hinterland containers that need to be unloaded by vessel m at terminal i
U_{ik}^{m}	preference of vessel m of being at terminal i during time slot k
v_{ij}^{m}	number of inter-terminal containers directly transported from terminal i to terminal j by vessel m
w_{i}^{m}	number of time steps vessel m has waited at terminal i
w_{mk}	waiting time of vessel m on segment k after coordination
W_{ik}^{m}	utility value of waiting time w_{i}^{m} during time slot k
$[W_{\text{start}}^{j}, W_{\text{end}}^{j}]$	closing time window of terminal j
x_{i}^{m}	time slot at which vessel m is at terminal i
$x_{mi}^{\text{start,terminal}}$	binary variable, $x_{mi}^{\text{start,terminal}} = 1$ if vessel m departures later than the start of closing time of terminal i; otherwise $x_{mi}^{\text{start,terminal}} = 0$
$x_{mi}^{\text{end,terminal}}$	binary variable, $x_{mi}^{\text{end,terminal}} = 1$ if vessel m arrives earlier than the end of closing time of terminal i; otherwise $x_{mi}^{\text{end,terminal}} = 0$
$x_{mi}^{\text{start,sea}}$	binary variable, $x_{mi}^{\text{start,sea}} = 1$ if vessel m departures later than the start of the handling time window of a sea-going vessel; otherwise $x_{mi}^{\text{start,sea}} = 0$
$x_{mi}^{\text{end,sea}}$	binary variable, $x_{mi}^{\text{end,sea}} = 1$ if vessel m arrives earlier than the end of the handling time window of a sea-going vessel; otherwise $x_{mi}^{\text{end,sea}} = 0$
$\mathcal{X}_{m}$	set of decision variables
$\mathcal{X}_{m}^{\text{as}}$	set of variables in the assignment problem
$\mathcal{X}_{m}^{\text{sc}}$	set of decision variables in the scheduling problem
y_{ij}^{m}	binary variable, $y_{ij}^{m} = 1$ if vessel m will travel from terminal i to j; otherwise $y_{ij}^{m} = 0$
z_{ik}^{m}	binary variable, $z_{ik}^{m} = 1$ if terminal i is on segment k of vessel m's rotation; otherwise $z_{ik}^{m} = 0$
α_{i}^{m}	arrival time of vessel m at terminal i
$\alpha_{i}^{\text{master}}$	arrival time of vessel m at terminal i in the master problem
α_{i}^{sub}	arrival time of vessel m at terminal i in the sub-problem
β_{i}^{m}	departure time of vessel m from terminal i
$\beta_{i}^{\text{master}}$	departure time of vessel m from terminal i in the master problem
β_{i}^{sub}	departure time of vessel m from terminal i in the sub-problem
γ_{i}^{m}	service time of vessel m at terminal i

γ_i^{master}	service time of vessel m at terminal i in the master problem
γ_i^{sub}	service time of vessel m at terminal i in the sub-problem
$\delta_{mm'i}$	binary variable, $\delta_{mm'i} = 1$ if the vessel m' arrives at terminal i earlier than vessel m; otherwise $\delta_{mm'i} = 0$
$\varepsilon_{mm'i}$	binary variable, $\varepsilon_{mm'i} = 1$ if the vessel m' arrives at terminal i earlier than vessel m; otherwise $\varepsilon_{mm'i} = 0$
$\eta_{i,r}$	departure time of the vessel that visits terminal j with rank r
θ_{mk}^q	binary variable, $\theta_{mk}^q = 1$ if vessel m will be handled at quay q; otherwise $\theta_{mk}^q = 0$
$\lambda_{i,r_{mi}^{\text{sub}}}^q$	departure time at quay crane q of terminal j after the vessel with the ranking of r_{mi}^{sub} has been handled
μ_i^m	binary variable, $\mu_i^m = 1$ if vessel m visits terminal i in the first
$\xi_{i,r_{mi}^{\text{sub}}}^q$	possible starting time at quay crane q for the vessel that will visit terminal j with the ranking of r_{mi}^{sub}
σ_i^m	binary variable, $\sigma_i^m = 1$ if vessel m visits terminal i in the end
$\tau_{k-1,k}^m$	traveling time between the terminal that vessel m visits at segment $k-1$ and the terminal that vessel m visits at segment k
ω_i^{master}	waiting time of vessel m at terminal i in the master problem
ω_i^{sub}	waiting time of vessel m at terminal i in the sub-problem

List of abbreviations

The following abbreviations are used in this thesis:

ADOPT	Asynchronous Distributed Optimization
AFB	Asynchronous Forward Bounding
AGV	Automated Guided Vehicle
AIS	Automatic Identification System
BTS	Barge Traffic System
CC	Cycle-cut
CP	Constraint Programming
CPA	Current Partial Assignment
CR	Cluster Root
CVRP	Capacitated Vehicle Routing Problem
DCOP	Distributed Constraint Optimization
DPOP	Dynamic Programming Optimization Protocol
DSA	Distributed Stochastic Algorithm
EDI	Electronic Data Interchange
ICT	Information and Communication Technology
IoT	Internet of Things

IP	Integer Programming
ITT	Inter Terminal Transport
KPI	Key Performance Indicators
LB	Lower bound
LNS	Large Neighborhood Search
LP	Linear Programming
LSP	Logistic service provider
MAS	Multi-agent System
MB-DPOP	Memory Bounded Dynamic Programming Optimization Protocol
MGM	Maximum Gain Message
MIP	Mixed Integer Programming
MIR	Maritime Inventory Routing Problem
MP	Mathematical Programming
NLP	Non-linear Programming
SRSP	Ship Routing and Scheduling
SyncBB	Synchronous Branch and Bound
TEN	Trans European Network
TEU	Twenty-foot Equivalent Unit
TSP	Traveling Salesman Problem
UB	Upper bound
VHF	Very High Frequency
VRP	Vehicle Routing Problem
VRPP	Vessel Rotation Planning Problem
VRPPD	Vehicle Routing Problem with Pickup and Delivery
VRPTW	Vehicle Routing Problem with Time Windows
VTS	Vessel Traffic Services

List of methods

The following methods are proposed in this thesis:

M1.1	Non-layered distributed constraint optimization method in Chapter 3
M1.2	Layered distributed constraint optimization method in Chapter 3
M2.1	The two-phase approach based on mixed-integer programming and coordination rules in Chapter 4
M2.2	The two-phase approach based on mixed-integer programming and CP solver in Chapter 4

M2.3	The two-phase approach based on mixed-integer programming and large neighborhood search-based heuristics in Chapter 4
M3.1	The hybrid method based on Benders decomposition and large neighborhood search in Chapter 5

TRAIL Thesis Series

The following list contains the most recent dissertations in the TRAIL Thesis Series. For a complete overview of more than 150 titles see the TRAIL website: www.rsTRAIL.nl.

The TRAIL Thesis Series is a series of the Netherlands TRAIL Research School on transport, infrastructure and logistics.

Li, S., *Coordinated Planning of Inland Vessels for Large Seaports*, T2016/23, December 2016, TRAIL Thesis Series, the Netherlands

Berg, M. van den, *The Influence of Herding on Departure Choice in Case of Evacuation: Design and analysis of a serious gaming experimental set-up*, T2016/22, December 2016, TRAIL Thesis Series, the Netherlands

Luo, R., *Multi-Agent Control of urban Transportation Networks and of Hybrid Systems with Limited Information Sharing*, T2016/21, November 2016, TRAIL Thesis Series, the Netherlands

Campanella, M., *Microscopic Modelling of Walking Behavior*, T2016/20, November 2016, TRAIL Thesis Series, the Netherlands

Horst, M. van der, *Coordination in Hinterland Chains: An institutional analysis of port-related transport*, T2016/19, November 2016, TRAIL Thesis Series, the Netherlands

Beukenkamp, *Securing Safety: Resilience time as a hidden critical factor*, T2016/18, October 2016, TRAIL Thesis Series, the Netherlands

Mingardo, G., *Articles on Parking Policy*, T2016/17, October 2016, TRAIL Thesis Series, the Netherlands

Duives, D.C., *Analysis and Modelling of Pedestrian Movement Dynamics at Large-scale Events*, T2016/16, October 2016, TRAIL Thesis Series, the Netherlands

Wan Ahmad, W.N.K., *Contextual Factors of Sustainable Supply Chain Management Practices in the Oil and Gas Industry*, T2016/15, September 2016, TRAIL Thesis Series, the Netherlands

Liu, X., *Prediction of Belt Conveyor Idler Performance*, T2016/14, September 2016, TRAIL Thesis Series, the Netherlands

Gaast, J.P. van der, *Stochastic Models for Order Picking Systems*, T2016/13, September 2016, TRAIL Thesis Series, the Netherlands

Wagenaar, J.C., *Practice Oriented Algorithmic Disruption Management in Passenger Rail-*

ways, T2016/12, September 2016, TRAIL Thesis Series, the Netherlands

Psarra, I., *A Bounded Rationality Model of Short and Long-Term Dynamics of Activity-Travel Behavior*, T2016/11, June 2016, TRAIL Thesis Series, the Netherlands

Ma, Y., *The Use of Advanced Transportation Monitoring Data for Official Statistics*, T2016/10, June 2016, TRAIL Thesis Series, the Netherlands

Li, L., *Coordinated Model Predictive Control of Synchromodal Freight Transport Systems*, T2016/9, June 2016, TRAIL Thesis Series, the Netherlands

Vonk Noordegraaf, D.M., *Road Pricing Policy Implementation*, T2016/8, June 2016, TRAIL Thesis Series, the Netherlands

Liu, S., Modeling, *Robust and Distributed Model Predictive Control for Freeway Networks*, T2016/7, May 2016, TRAIL Thesis Series, the Netherlands

Calvert, S.C., *Stochastic Macroscopic Analysis and Modelling for Traffic Management*, T2016/6, May 2016, TRAIL Thesis Series, the Netherlands

Sparing, D., *Reliable Timetable Design for Railways and Connecting Public Transport Services*, T2016/5, May 2016, TRAIL Thesis Series, the Netherlands

Rasouli, S, *Uncertainty in Modeling Activity-Travel Demand in Complex Urban Systems*, T2016/4, March 2016, TRAIL Thesis Series, the Netherlands

Vries, J. de, *Behavioral Operations in Logistics*, T2016/3, February 2016, TRAIL Thesis Series, the Netherlands

Goi-Ros, B., *Traffic Flow at Sags: Theory, Modeling and Control*, T2016/2, March 2016, TRAIL Thesis Series, the Netherlands

Khademi, E., *Effects of Pricing Strategies on Dynamic Repertoires of Activity-Travel Behaviour*, T2016/1, February 2016, TRAIL Thesis Series, the Netherlands

Cong, Z., *Efficient Optimization Methods for Freeway Management and Control*, T2015/17, November 2015, TRAIL Thesis Series, the Netherlands

Kersbergen, B., *Modeling and Control of Switching Max-Plus-Linear Systems: Rescheduling of railway traffic and changing gaits in legged locomotion*, T2015/16, October 2015, TRAIL Thesis Series, the Netherlands

Brands, T., *Multi-Objective Optimisation of Multimodal Passenger Transportation Networks*, T2015/15, October 2015, TRAIL Thesis Series, the Netherlands

Ardic, O., *Road Pricing Policy Process: The interplay between policy actors, the media and public*, T2015/14, September 2015, TRAIL Thesis Series, the Netherlands

Xin, J., *Control and Coordination for Automated Container Terminals*, T2015/13, September 2015, TRAIL Thesis Series, the Netherlands

Anand, N., *An Agent Based Modelling Approach for Multi-Stakeholder Analysis of City Logistics Solutions*, T2015/12, September 2015, TRAIL Thesis Series, the Netherlands

Samenvatting

Zeehavens zijn cruciale knooppunten in de internationale handel en transport. Sommige van de ladingen die aankomen op zeehavens worden overgeladen voor transport naar andere havens, terwijl andere ladingen worden vervoerd naar bestemmingen in het achterland. Iedere keer dat een binnenvaartschip een haven binnenkomt, bezoekt het verschillende terminals verspreid over het havengebied. Er zijn twee belangrijke problemen in de planning van binnenvaartschepen in grote zeehavens: (i) het lange verblijf van de binnenvaartschepen in de haven; en (ii) de ontoereikende terminal en kadeplanning met betrekking tot de vaarschema's van zeeschepen en binnenvaartschepen.

Om deze problemen op te lossen heeft dit proefschrift als doel de operaties van binnenvaartschepen in de zeehavens te verbeteren. Dit kan vervolgens de efficintie en betrouwbaarheid van de binnenvaartschepen van zeehavens naar het achterland en vice versa verbeteren. Tegelijkertijd kan dit ook bijdragen aan de verbetering van het inter-terminal transport in grote zeehavens, als een mogelijke manier om congestie op wegen te verlichten. Bovendien kan efficinte afhandeling van binnenvaartschepen flexibele planning van het vervoer over water bevorderen, zodat deze vorm van vervoer beter kan worden gentegreerd in synchromodale vervoersketens.

Drie klassen van automatische cordinatiemethoden worden in dit proefschrift voorgesteld voor toepassing in zeehavens van verschillend formaat: een gedeeltelijk coperatieve cordinatiemethode met enkel-gelaagde interactie gebaseerd op gedistribueerde beperkingsoptimalisatie wordt voorgesteld voor kleine zeehavens; een gedeeltelijk coperatieve cordinatiemethode met meervoudig-gelaagde interacties op basis van MIP, cordinatie regels en constraint programming wordt voorgesteld voor middelgrote zeehavens; een volledig coperatieve cordinatiemethode met meervoudig-gelaagde interacties op basis van Benders decompositie en large neighborhood search wordt voorgesteld voor grote zeehavens.

Met de voorgestelde methoden kunnen scheepsbeheerders ten eerste bepalen in welke mate zij acties willen cordineren, ofwel gedeeltelijk ofwel volledig; ten tweede kunnen terminal operatoren deze methoden gebruiken om in te schatten hoeveel tijd elk binnenvaartschip bij de terminal zal besteden, wat mee kan worden genomen bij het bepalen van planningen; ten derde kunnen de voorgestelde methoden nieuwe en betere rotaties genereren op basis van up-to-date informatie, hierbij rekening houdend met real-time verstoringen (bijvoorbeeld door ongelukken of niet-functionerende apparatuur), die eerder gegenereerde rotaties mogelijk suboptimaal maken. Simulatieresultaten tonen aan dat de voorgestelde aanpak de round-trip tijd en wachttijden van binnenvaartschepen aanzienlijk kunnen verminderen.

Vanuit het perspectief van het gebruik van binnenvaartschepen voor inter-terminal trans-

port (ITT) in zeehavens kunnen scheepsoperators de voorgestelde methoden daarnaast gebruiken om op basis van besteedbare extra round-trip tijd en wachttijd te beslissen of ze samen willen werken om extra ITT containers te transporteren. Bovendien kunnen terminal operators ook inschatten hoeveel extra ITT containers door de binnenkomende schepen vervoerd kunnen worden tijdens verschillende momenten van de dag, zodat zij de ITT containers overeenkomstig kunnen inplannen.

Samenvattend onderzoekt dit proefschrift de operationele planning van binnenvaartschepen in zeehavens. Dit proefschrift toont het potentieel van de voorgestelde nieuwe benaderingen ter verbetering van de efficintie en betrouwbaarheid van het binnenvaarttransport in zeehavens.

Summary

Seaports are crucial nodes in international trade and transport. Some of the cargoes arriving at seaports are transshipped to other ports, while others are transported to inland destinations. Every time an inland container vessel enters the port, it calls at many different terminals spread over the port area. Two coordination problems exist in the planning of inland vessels in large seaports: firstly, the long stay in the port and secondly, the insufficient terminal and quay planning with respect to the sailing schedules of sea-going vessels and inland vessels. To solve these problems, this thesis aims to improve the reliability and efficiency of inland vessel transport in seaports. To achieve this, efficient handling of inland container vessels in large seaports is required. This could improve the efficiency and reliability of inland waterway transport from seaports to hinterland and vice versa. Meanwhile, this could also contribute to enhancing of the inter-terminal transport in large seaports, as a potential solution for alleviating congestion on roads. Moreover, efficient handling of inland vessels could facilitate flexible planning of transport over water, so that this transport mode can be better integrated into the synchromodal transport chain.

Therefore, three classes of automatic coordination methods are proposed in this thesis for seaports with different sizes: for small seaports, a partially-cooperative coordination method with single-level interaction based on distributed constraint optimization is proposed; for medium-sized seaports, a partially-cooperative coordination method with multi-level interactions based on MIP, coordination rules and constraint programming, is proposed; for large seaports, a fully-cooperative coordination method with multi-level interactions based on Benders decomposition and large neighborhood search is proposed.

With the proposed methods, firstly, the vessel operators can decide to what extent they would like to be coordinated, either partially-cooperative or fully-cooperative; secondly, terminal operators can also use these methods to estimate how much time each inland vessels spends at the terminal, in order to determine the schedules of terminal operations; thirdly, whenever real-time disturbances or accidents happen, the previously generated rotations might be no longer optimal, the proposed methods can take into account these disturbances and generate new and better rotations for vessel operators based on the up-to-date information. Simulation results demonstrate that the proposed approach can significantly reduce the round-trip time the inland vessels spend in the port, as well as reducing the time they spend waiting at container terminals.

Moreover, from the perspective of using inland vessels also for inter-terminal transport (ITT) in seaports, vessel operators can use the proposed methods to decide whether they are willing to cooperate to take extra ITT containers based on the possible extra round-trip time

and waiting time that they may spend. Moreover, terminal operators can also estimate how much extra ITT containers could be transported by the incoming vessels during different times of a day, so that they can plan the ITT containers to be transported by inland vessels accordingly.

To conclude, this thesis investigates the operational planning of inland vessels in seaports. This thesis shows the potential of the proposed new approaches for improving the efficiency and reliability of inland vessel transport in seaports.

Curriculum vitae

Shijie Li was born in December 3rd, 1988 in Jingmen, Hubei, China. In 2010, she obtained her B.Sc degree in Detection, Guidance and Control from the College of Automation, Harbin Engineering University. After this, she was recommended to continue her master education in the same college. She received her M.Sc degree in Control Theory and Control Engineering in 2013.

In October 2012, she was sponsored by the Chinese Scholarship Council to become a Ph.D candidate at the Department of Maritime and Transport Technology. In her Ph.D project, she worked on the coordinate planning of inland vessels in large seaports. Her research interests include artificial intelligence, constraint programming, operations research, and transport and logistic systems.

Publications

1. Shijie Li, Rudy R. Negenborn, Gabriel Lodewijks (2017). Closed-loop coordination of inland vessels operations in large seaports using hybrid logic-based benders decomposition. *Transportation Research Part E: Logistics and Transportation Review*, vol. 97, pp. 1–21, 2017.

2. Shijie Li, Rudy R. Negenborn, Gabriel Lodewijks (2016). Approach integrating mixed-integer programming and constraint programming for planning rotations of inland vessels in a large seaport. *Transportation Research Record: Journal of the Transportation Research Board*, vol.2549, pp. 1–8, 2016.

3. Shijie Li, Rudy R. Negenborn, Gabriel Lodewijks (2016). Distributed constraint optimization for addressing vessel rotation planning problems. *Engineering Applications of Artificial Intelligence*, vol. 48, pp. 159–172, 2016.

4. Shijie Li, Rudy R. Negenborn, Gabriel Lodewijks (2016). Planning inland vessels operations in large seaports using a two-phase approach. *Submitted to a journal*.

5. Shijie Li, Rudy R. Negenborn, Gabriel Lodewijks (2016). A logic-based benders decomposition approach to improve coordination of inland vessels for inter-terminal transport. In *Proceedings of the 7th International Conference on Computational Logistics (ICCL 2016)*, pp. 96–115, Lisbon, Portugal, September 2016.

6. Shijie Li, Rudy R. Negenborn, Gabriel Lodewijks (2016). Integrated MIP-CP approach for planning rotations of inland vessels in a large seaport (2016). In *Proceedings*

of the Transportation Research Board 95th Annual Meeting (TRB 2016), Washington, D.C., United States, January 2016.

7. Shijie Li, Rudy R. Negenborn, Gabriel Lodewijks (2015). A two phase approach for inter-terminal transport of inland vessels using preference-based and utility-based coordination rules. In *Proceedings of the 6th International Conference on Computational Logistics (ICCL 2015)*, pp. 281–297, Delft, The Netherlands, September 2015.

8. Shijie Li, Huarong Zheng, Rudy R. Negenborn, Gabriel Lodewijks (2015). Coordination for efficient transport over water. In *Proceedings of the 19th IEEE International Conference on Computer Supported Cooperative Work in Design (IEEE CSCWD15)*, pp. 389–394, Calabria, Italy, May 2015.

9. Shijie Li, Rudy R. Negenborn, Gabriel Lodewijks (2015). Vessel rotation planning: A layered distributed constraint optimization approach. In *Proceedings of the 7th International Conference on Agents and Artificial Intelligence (ICAART 2015)*, pp. 197–204, Lisbon, Portugal, January 2015.

10. Shijie Li, Rudy R. Negenborn, Gabriel Lodewijks (2014). A distributed constraint optimization approach for vessel rotation planning. In *Proceedings of the 5th International Conference on Computational Logistics (ICCL 2014)*, Valparaiso, Chile, pp. 61–80, September 2014.

11. Shijie Li, Rudy R. Negenborn, Gabriel Lodewijks (2013). Survey on planning problems in inland waterway transport: Current status and future perspectives. In *Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems (IEEE ITSC 2013)*, The Hague, The Netherlands, pp. 1231–1237, October 2013.