JUDITH MULDER

Network Design and Robust Scheduling in Liner Shipping

Network Design and Robust Scheduling in Liner Shipping

Network Design and Robust Scheduling in Liner Shipping

Het ontwerpen van netwerken en robuuste dienstregelingen voor lijnschepen

Thesis

to obtain the degree of Doctor from the
Erasmus University Rotterdam
by command of the
rector magnificus

Prof.dr. H.A.P. Pols

and in accordance with the decision of the Doctorate Board

The public defense shall be held on

Friday 1 July 2016 at 09:30 hours

by

JUDITH MULDER
born in Delft, the Netherlands.

Erasmus University Rotterdam

Erasmus

Doctoral Committee

Promotor:	Prof.dr.ir. R. Dekker
Other members:	Prof.dr. R. Zuidwijk
	Prof.dr. L.G. Kroon
	Prof.dr. H.N. Psaraftis
Copromotor:	Dr. W.L. van Jaarsveld

Erasmus Research Institute of Management - ERIM
The joint research institute of the Rotterdam School of Management (RSM)
and the Erasmus School of Economics (ESE) at the Erasmus University Rotterdam
Internet: http://www.erim.eur.nl

ERIM Electronic Series Portal: http://hdl.handle.net/1765/1

ERIM PhD Series in Research in Management, 384
ERIM reference number: EPS-2016-384-LIS
ISBN 978-90-5892-446-9

Design: B&T Ontwerp en advies www.b-en-t.nl

This publication (cover and interior) is printed by haveka.nl on recycled paper, Revive®.
The ink used is produced from renewable resources and alcohol free fountain solution.
Certifications for the paper and the printing production process: Recycle, EU Flower, FSC, ISO14001.
More info: http://www.haveka.nl/greening

Acknowledgments

At the end of my PhD trajectory, I have the opportunity to thank the people that contributed to the process of writing this thesis.

First, I woud like to thank my promotor Rommert Dekker. You have already introduced me to my research topic when I was writing my bachelor's and master's theses and made me enthousiastic to continue this work during a PhD project. You have always found time in your busy schedule to plan a meeting with me to discuss my research and share your many ideas. I am also very grateful to my copromotor Willem van Jaarsveld. Although our official collaboration only started late in my PhD trajectory, you were already interested in my research when I just started. Your enthousiasm helped me to overcome the difficult moments in my PhD trajectory. I really enjoyed working together with you.

I also want to thank Leo Kroon, Harilaos Psaraftis and Rob Zuidwijk for being part of my inner committee and evaluating my thesis. Leo, thank you for your valuable feedback on one of my chapters. Harilaos, you were always actively participating in the many conferences where we were together. Your questions about my presentations provided me with valuable feedback on my research. Rob, I really had a nice time during the conferences we attended together. Dennis Huisman and Albert Veenstra, I would like to thank you for being part of my committee. Furthermore, I want to thank SmartPort for providing a platform to meet others and discuss ideas. I have met many people and learned a lot during the events.

I want to thank Ilse, Kristiaan, Lanah, Mathijn, Remy, Twan, Willem and Zahra for making me feel welcome when I just started and for the many enjoyable breaks we had together. Adriana, Bart, Bert, Charlie, Chiel, Dennis, Evelot, Gertjan, Gönül, Harwin, Iris, Jessica, Kevin, Koen, Mustafa, Nick, Niels, Paul, Pieter, Rutger, Sha, Thomas, Thomas and Wilco, thanks for maintaining this good atmosphere during the last years of my PhD trajectory. I really enjoyed my years as a PhD candidate and am very grateful to have had you all as my colleagues. Thank you for always being there for me when I wanted to discuss my research or just wanted to have a break. I have had many roommates during the last five years, but only three of them had to endure my presence every day: Agapi, Mustafa and Twan, thanks for providing a pleasant environment to work in. I also want to thank my colleagues from RSM, in particular Alp, Amir, Clint, Evelien, Evşen, Jelmer, Joris, Luuk, Marie, Panos, Paul, Tim and Xishu, for the nice conversations we had. Ilse, I want to thank you for bringing me the idea to start this process and answering all the questions I had when I just started. Charlie, thanks for introducing and organizing the monthly game nights, providing a nice distraction from our work.

Finally, I want to thank my family and friends for their support in the last years. Anneke and Robert, Ilse and Koen, Marchella and Jelmer, Tanny and Ruben, and Yvonne, thank you for the distraction you offered me during my work. Marchella, I am really happy that you agreed to be one of my paranymphs. Furthermore, I want to thank my parents and my sisters, Marlies and Lotte. I want to thank my father for introducing me to the field of mathematics and econometrics. Marlies, thanks for helping me with the important side issues during the process. I want to thank my mother, Lotte and Suzan for baking many cakes and muffins during the last years, which successfully initiated many coffee breaks with my colleagues. Mathijn, when I decided to start this project, I didn't know that it would bring me so much. Thank you for standing by my side as my paranymph and for your patience and support during the last years.

Judith Mulder
Delft, April 2016

Table of Contents

Chapter 1

Introduction[1]

Seaborne shipping is the most important mode of transport in international trade. In comparison to other modes of freight transport, like truck, aircraft, train and pipeline, ships are preferred for moving large amounts of cargo over long distances, because shipping is more cost efficient and environmentally friendly (Rodrigue et al. 2013). Reviews of maritime transport provided by the United Nations Conference on Trade And Development (UNCTAD 2014) show that about 80% of international trade is transported (at least partly) by sea. Sea transport can be separated into dry bulk (e.g. steel, coal and grain), liquid bulk (e.g. oil and gas) and containerized cargo. In 2013, containerized cargo is with a total of 1.5 billion tons responsible for over 15% of all seaborne trade, which resulted in a world container port throughput of more than 650 million twenty-foot equivalent units (TEUs).

The shipping market comprises three types of operations: tramp shipping, industrial shipping and liner shipping (Lawrence 1972). Tramp ships have no fixed route, but ensure an immediate delivery for any type of cargo from any port to any port, resulting in irregular activities. The behaviour of tramp ships is thus comparable to taxi services. In industrial shipping the cargo owner also controls the ships used to transport the freight. The objective of industrial operators is to minimize the cost of shipping the owned cargoes. Liner ships follow a fixed route within a fixed time schedule and serve many smaller customers. The schedules are usually published online and demand depends on the operated schedules. Hence, liner shipping services are comparable to train and bus services.

In the next section, we will discuss a variety of liner shipping problems on the strategic, tactical and operational planning levels, while Section 1.2 introduces some problems

[1]This chapter is based on the online report of the Econometric Institute Report Series with small modifications: Mulder, J. and Dekker, R. Optimization in container liner shipping, report number: EI 2016-05.

related to terminal operations. Sections 1.3 and 1.4 discuss respectively the influence of geographical bottlenecks and the importance of operations research in solving these problems. This overview is based on the following overview articles: Ronen (1983, 1993), Christiansen et al. (2004, 2007, 2013), Meng et al. (2014). In Section 1.5, a case study is performed for six Indonesian ports to provide insight into the different problems. Finally, Section 1.6 provides an overview of the remainder of this dissertation and Section 1.7 clarifies the contribution of each of the chapters.

1.1 Container liner shipping

In this thesis, we will focus on the liner shipping operations concerned with the transport of containers. Liner shipping operators face a wide variety of decision problems in operating a liner shipping network. First, at the strategic planning level, the fleet size and mix problem and the market and trade selection problem need to be solved. In the fleet size and mix problem, operators decide on the fleet composition and in the market and trade selection problem on which trade route to serve. At the tactical planning level the network needs to be designed, prices need to be set and empty containers have to be repositioned. Finally, at the operational level, operators need to determine the cargo routing through the network and how to deal with disruptions. Furthermore, they can make adjustments to the earlier set prices and need to determine a plan to store all the containers on the ship during the loading process. These problems are considered in respectively the cargo routing, disruption management, revenue management and stowage planning problems. Some problems have to be considered at both the tactical and the operational planning level, such as setting the sailing speed and optimizing the bunkering decision and designing a (robust) schedule. In this section, we will introduce all these decision problems. In these problems we will make use of the following terminology. Liner shipping operators will also be referred to as liner shipping companies, liner companies or liners. Liner ships follow fixed routes, which are sequences of port calls to be made by the ship. Route networks consist of a set of services, which are routes to which a ship is allocated. Besides publishing their route networks, liner companies also publish the exact arrival and departure days at each port of call. When we refer to the route together with the arrival and departure days, we will talk about a schedule. Finally, a round tour refers to one traversal of a route and a (sea) leg refers to the sailing between two consecutive ports.

1.1.1 Strategic planning level

The strategic planning level consists of long term decision problems. Generally, these problems are only solved at most once a year. Examples of long term decision problems in container liner shipping are: the fleet size and mix problem and the market and trade selection problem. In this thesis, we only consider problems in which the markets and trade lanes are already selected. The fleet size and mix problem is only considered in Chapter 2; in the other chapters the fleet composition is already known.

Fleet size and mix

In the fleet size and mix problem, a liner company decides on how many ships of each type to keep in its fleet. Container ship sizes have increased substantially because of the growth in container trade and because of competitive reasons. For example, the Emma Maersk (introduced in 2006) has an estimated capacity of more than 14,500 TEU. Before the introduction of the Emma Maersk, the capacity of the largest container ship in the world was less than 10,000 TEU. In 2013, Maersk introduced a series of ships belonging to the Triple E class with capacities of over 18,000 TEU, while both MSC and CSCL introduced container ships with a capacity of more than 19,000 TEU in 2015. Ships benefit from economies of scale when they are sailing at sea, but they might suffer diseconomies of scale when berthing in ports. However, the effect of the economies of scale at sea is much larger than the effect of the diseconomies of scale in ports (Cullinane and Khanna 1999). Hence, economies of scale in larger container ships can lead to substantial savings if the capacity of the ship is adequately used. However, if the demand decreases and the liner company is not able to fill these large ships any more, higher operational costs are incurred by these large ships. Therefore, fleet size and mix problems are used to balance the possible benefits from economies of scale with the risk of not being able to use the full capacity of the ships. Since building a new container ship may take about one year and ships usually have life expectancies of 25-30 years, future demand and availability of ships play an important role in the fleet size and mix problem. Pantuso et al. (2014) present an overview of research conducted on the fleet size and mix problem. Most of these works incorporate ship routing and/or deployment decisions in order to ensure feasibility of demand satisfaction and capacity constraints.

Market and trade selection

Before a liner container shipping company starts building a network and operating the routes, it has to decide which trade lanes to participate in. The Asia-Europe trade lane is an example of a popular trade lane. Clearly, the selected trade lanes influence the type and number of ships required. For example, trade lanes serving the US will usually not use vessels from the Maersk Triple E class, since they can not sail through the Panama Canal and most ports in the US are not capable of handling these large vessels. Furthermore, the type and amount of cargoes that have to be transported and the required sailing frequency may influence the ship types used on the trade lane.

1.1.2 Tactical planning level

Medium-term decision problems belong to the tactical planning level. Liner companies usually change their service networks every 6-12 months, but more often in case of world-wide disruptions. Problems that have to be solved again each time the service network is adjusted are considered to belong to the tactical planning level. The examples that will be discussed next are: the network design problem, the pricing problem and the empty container repositioning problem. In this thesis, we are mainly interested in the network design problem out of these problems. We consider prices to be fixed and do not include the repositioning of empty containers in deciding upon a liner network.

Network design

The network design problem in liner shipping can be split into two subproblems. The first subproblem is the routing and scheduling problem, which is concerned with determining which ports will be visited on each route, in which order the ports will be called at and what the arrival and departure times at each port will be. Many studies only consider the routing decisions in the network design problem and do not address the scheduling problem of determining the actual arrival and departure time. The second subproblem considers the fleet deployment and frequency. Here, the liner company determines which ships will be used to sail each route and with what frequency the ships will call at the ports along the route. In general, a weekly frequency is imposed, which facilitates planning by shippers, but this can be relaxed to a biweekly frequency for low demand routes or multiple port calls per week for high demand routes. Sometimes sailing speed optimization is considered as a third subproblem of the network design problem, but in most studies the sailing speed is either assumed to be fixed and known, or will follow directly from the imposed frequency. Usually, the cargo routing problem is already included (using expected demand as input) in the network design problem in order to evaluate the profitability of a service network. The cargo routing problem will be discussed in more detail with the operational planning level problems.

The structure of the routes in a network can be divided into several types, like non-stop services or end-to-end connections, hub and spoke systems, hub and feeder systems, circular routes, butterfly routes, pendulum routes and nonsimple routes (Notteboom 2004, Brouer et al. 2014a). A non-stop service or end-to-end connection provides a direct connection between two ports: a ship sails from one port to the other and immediately back to the first port; sometimes this is called a shuttle service, although that also requires a high frequency. In a hub and spoke system, usually one port is identified as the main or hub port. All other ports (also called feeder ports) are served using direct services from the hub port. However, it is also possible that multiple hubs are applied, which are connected with each other and used as transshipment ports to satisfy demand between different feeder ports, in which case they might also be referred to as main ports. In the hub and feeder system, feeder ports might also be visited on routes with multiple port calls. Circular routes are cyclic and visit each port exactly once, while butterfly routes allow for multiple stops at the same port in one cycle. Pendulum routes visit the same port in both directions, only in reverse order. Finally, ports can be visited multiple times on nonsimple routes. Examples of some of these route types are provided in the case study in Section 1.5.

The liner shipping network design problem has attained quite some attention in the literature. We will briefly describe some of the recent publications on this problem. Plum et al. (2014b) consider a subproblem of the network design problem. They develop a branch-and-cut-and-price algorithm to find a single vessel round trip. Each port has to be visited exactly once and the best paying demand pairs are accepted and transported. Polat et al. (2014) consider an adapted neighbourhood search method to solve a hub and feeder system with one single hub. Finally, Zheng et al. (2015) propose a genetic algorithm to solve the same problem with multiple hubs.

Wang and Meng (2014) propose a column-generation heuristic approach to find the best liner shipping network. Each port can be visited twice during each route: once on the inbound direction and once on the outbound direction. Brouer et al. (2014a) provide both a base mixed integer programming formulation for the network design problem and benchmark data instances. They propose a column generation approach to generate butterfly and pendulum routes. Plum et al. (2014c) extend the butterfly routes as used in the benchmark model to routes with multiple butterfly ports. Brouer et al. (2014b) propose a matheuristic to solve the base network design problem with nonsimple routes. Although their assumptions are a bit more restrictive than in the benchmark paper (Brouer et al. 2014a), they are able to construct a more profitable route network using this approach.

Liu et al. (2014), Mulder and Dekker (2014) and Wang et al. (2015) consider slightly different network design problems. Liu et al. (2014) consider a problem in which the port-to-port demand is combined with the inland transportation. They start with an initial liner network and try to improve it while also including the transportation between the ports and the real origin and destination of the demand. Mulder and Dekker (2014) consider the strategic liner shipping network design problem, including the fleet size and mix problem, using a hub and feeder network structure. Wang et al. (2015) consider the liner shipping network alteration problem. In this problem, an initial liner network is given and this network is modified to become more profitable.

Pricing

The goal of liner companies is to maximize profit by transporting containers from one port to another. The revenue of the company is determined by the amount of containers that are transported and the price that will be charged for each container. The pricing problem is concerned with which price to charge for each possible demand pair. Factors that influence the price are for example: distance, trade direction, expected demand and expected capacity. The pricing problem is more a marketing, micro-economic problem than an operations research problem. Although it is an interesting problem, it has hardly been touched. Two approaches exist: cost-plus and what the market can pay. Yet, even determining the cost is a difficult allocation problem.

Empty container repositioning

Containers delivering import products in a region can be re-used to transport export goods to another region. However, most regions face an imbalance between import and export containers. This trade imbalance results in an excess of empty containers in regions with more import than export and a shortage of containers for high export regions. The empty container repositioning problem tries to reallocate the empty containers in order to solve the imbalance, where costs are associated with transporting a container from one region to another. The repositioning of empty containers is considered to be very costly, since there is no clear revenue associated with it.

Some recent papers dealing with the empty container repositioning problem are the following. Both Di Francesco et al. (2014) and Long et al. (2015) consider the empty container repositioning problem under uncertain container demand and use a stochastic optimization approach to solve it. Zhang and Facanha (2014) consider the problem of repositioning empty containers to the location of demand in the US. Empty containers are transported using trucks or trains to the location where they can be loaded. If containers can not be allocated to a loading location, they are transported to a West Coast port to be shipped to Asia. Huang et al. (2015) consider the network design problem with both laden and empty container repositioning. Multiple hub ports are identified, where transshipment from feeder ports might take place. They select the best routes from a candidate set of routes, which is used as input in the model.

1.1.3 Operational planning level

The operational planning level captures the problems that occur during the execution of the routes in the service network. In order to solve operational level problems, reliable information about the actual situation is needed. Hence, operational problems usually need to be solved relatively shortly before the solutions have to be implemented. Next we will discuss the cargo routing problem, the disruption management problem, the revenue management problem and the stowage planning problem. Although all these four problems are important to liners, we focus in this thesis only on the cargo routing problem. The cargo routing problem is jointly considered with the network design problem at the tactical level.

Cargo routing

The cargo routing problem takes the liner shipping network and container demand as an input. The goal of this problem is to find a cargo flow over the network, satisfying the capacity constraints imposed by the allocated container ships, that maximizes the profit of transporting the containers. Costs are associated to (un)loading and transshipment operations. A transshipment occurs when a container has to be unloaded from a ship and loaded to another ship in order to arrive at its destination. Additionally, penalties can be imposed for demand that is not met. It is also possible to include transit time constraints to guarantee that containers will arrive on time at their destination.

Formulations of the cargo routing problem can be distinguished in OD-based link flow formulations, origin/destination-based link flow formulations, segment-based flow formulations and path-based formulations (Meng et al. 2014). All flow formulations consider the amount of flow at a link or segment of the route as decision variables in the model. Flow balance constraints ensure that all flow starts at the origin port and arrives at the destination port, but the exact route followed by a container might not be immediately clear from the model. In the OD-based link flow formulation, both the origin and the destination of the container are stored for each link in the network, while the origin/destination-based link flow formulations only store the origin or destination of the container. In this way, the number of decision variables can be reduced significantly. In a segment-based flow formulation, consecutive links of a route are already combined into segments before building the model. Segment-based flow formulations reduce the number of decision variables even more, but limit the possibility of transshipment operations to the ports at the beginning and end of the predefined segments. Finally, in path-based formulations complete container paths from origin to destination are generated beforehand and used as decision variable in the model. These paths might also include transshipment operations. The disadvantage of path-based formulations is that the number of paths might explode, such that more complex methods, like column generation, are needed to solve the problem. However, path-based formulations can usually be solved faster than flow-based formulations. Furthermore, transit time constraints are easily incorporated in path-based formulations, while this is generally much more troublesome in flow-based formulations. Little research is performed on the separate cargo routing problem: usually it is considered as a subproblem of the network design problem. Recently, Karsten et al. (2015) considered the cargo routing problem with transit time constraints. They propose a path-based formulation exploiting the ease to include transit time in this type of model. Their findings show that including transit time constraints in the cargo routing model is essential to find practically acceptable container paths and does not necessary increase computational times.

Disruption management

During the execution of the route schedules, ships may encounter delays. The disruption management problem focuses on which actions should be taken in order to get back on schedule after a disruption has occurred. Examples of actions that might be performed are: changing the sailing speed, swap port calls, cut and go (leave the port before all containers are (un)loaded) or skip a port. Usually, the goal of disruption management is to find a sequence of actions with minimum cost such that the ship will be back on schedule at a predetermined time. Brouer et al. (2013) propose a mixed integer programming formulation to solve this problem and prove that the problem is NP-hard. However, experimental results show that the model is able to solve standard disruption scenarios within ten seconds to optimality.

Revenue management

At the operational level, more information about the demand and available capacity of a ship is available. Therefore, it might be profitable for liner companies to vary their prices based on the available capacity between a port pair. Liners will probably charge higher prices related to low capacity pairs, while they might reduce the prices on legs where the available capacity is high.

Stowage planning

The stowage planning problem determines at which location containers are stored on the ship during the loading process. The stowage planning is a very complicated process with many constraints. Essential constraints are for example the stability of the ship both during the next sea leg and during the (un)loading process. Furthermore, containers may have to be stored at specific locations on the ship, like reefer containers. However, the storage of the containers also influence the (un)loading process in the next ports. Ideally, all containers with destination in the next port are stored on top of the stack, but this may take too many movements in the current port. Hence, a trade-off between the number of moves required to store and to discharge a container has to be made. Tierney et al. (2014) prove that the container stowage planning problem is a NP-complete problem.

1.1.4 Both tactical and operational level

Finally, some problems can either be considered at two different planning levels or have to be considered at two planning levels at the same time. For example, sailing speed optimization and bunkering optimization can be considered at the moment a new service network is designed, but the solutions to these problems can be reconsidered during the execution of the routes. Furthermore, robust schedule design is an example of a decision problem that combines decisions to be taken at the tactical and at the operational level. In this thesis, we do not consider the bunkering problem. Further, the sailing speed on a route is chosen in such a way that weekly route durations are guaranteed, but no real optimization is performed. The robust schedule design problem is studied extensively in Chapters 4 and 5.

Sailing speed and bunkering optimization

Both sailing speed optimization and bunkering optimization are typical problems that can be considered at two different planning levels. At the tactical planning level, the environmental aspects of sailing speed are usually considered. At the operational level, sailing speed is mostly used as an instrument to reduce delays incurred by the ship. The bunkering optimization problem is concerned with deciding at which ports ships are going to be refuelled. Initially, a bunker refuelling plan is made given estimates of the bunker price at the moment the ship will be at a bunkering station. Shipping lines also regularly make bunkering contracts, containing the ports where bunker can be purchased, the amount to be purchased, the price to be paid and the validity duration of the contract (Pedrielli et al. 2015). However, due to fluctuations in prices or fuel consumption, this initial plan might have to be adjusted at the operational planning level. At this stage, more accurate information about the fuel prices and availability at the ports and the bunker level of the ship is available. Sailing speed plays an important role in the bunker fuel consumption of ships and hence sailing speed optimization is often included in the bunkering optimization problem (Yao et al. 2012).

Recently, the sailing speed and bunkering optimization problems have received increasing attention. Psaraftis and Kontovas (2014) consider the speed optimization problem at the operational planning level. They include fuel prices, freight rates, cargo inventory costs and fuel consumption dependencies on payload into their model. Kontovas (2014) and Du et al. (2015) consider the influence of sailing speed on fuel emissions. Wang et al. (2013) provide a literature review on bunker consumption optimization problems. Bunker consumption is an important input for bunkering optimization. Yao et al. (2012) study the bunker fuel management strategy for a single liner shipping route. The strategy consists of the selection of the bunkering ports, the determination of the bunkering amounts and the adjustments in the sailing speeds. They consider a deterministic situation in which all parameters, including bunker costs, are fixed and known. Plum et al. (2014a) and Pedrielli et al. (2015) study the problem to determine the optimal bunkering contracts. Plum et al. (2014a) propose a mixed integer programming model, which is solved using column generation. Also, the possibility to purchase bunker on the spot market is included in their model. Pedrielli et al. (2015) use a game theoretical approach to design the contracts. Wang et al. (2014) propose a fuzzy approach to include uncertainties in the bunkering port selection problem. Their method returns a ranking of ports based on the profitability to bunker in these ports. Sheng et al. (2015) implement an (s, S) policy to jointly optimize the bunkering and speed optimization problem taking into account both bunker price and consumption uncertainty. Finally, Wang and Meng (2015) consider the robust bunker management problem, taking into account that the real sailing speed might differ from the planned sailing speed.

Robust schedule design

Robust schedule design can be seen as a combination of the scheduling problem at the tactical level and disruption management or sailing speed optimization at the operational level. The order in which ports are visited is considered to be an input of this problem. The goal is to jointly determine the planned arrival and departure times in each port and the actions that will be performed during the execution of the route when delays are incurred. The difficulty of this problem is that the tactical and operational planning level problems can not be solved separately, but have to be considered simultaneously.

The problem to determine the scheduled arrival and departure times under uncertainty in port times and a predetermined sailing speed policy is considered in Qi and Song (2012) and Wang and Meng (2012b). Qi and Song (2012) provide some useful insights in the optimal schedule under 100% service level constraints. Wang and Meng (2012b) formulate the problem as a two-stage stochastic programming problem and solve it using sample average approximation. They are able to find solutions with an objective value within 1.5% of the optimal solution in less than one hour.

1.2 Terminal operations

Liner shipping operations are closely related to terminal operations and decisions about ships cannot be taken while disregarding their effects on terminals. In fact, terminals are the largest bottleneck for shipping. It is important to have the right berth slots and to be loaded and unloaded quickly and in a predictable manner. There are many ports all over the world with a large number of ships waiting in front of the harbour to be allowed to berth. Accordingly we will discuss those terminals operations aspects which directly affect shipping, viz. berth scheduling, crane allocation and container stacking.

1.2.1 Berth scheduling

Both at a tactical as well as at an operational level, liner shipping schedules are made while taking berth availability into account. On a tactical level, when designing the liner shipping routes, agreements are made with terminals on berth availability and productivity (how many cranes and crane teams will be employed and how many container moves will be done per hour). This enables the shipping line to calculate the port time of his ships and to complete the ship scheduling. Naturally buffer times are incorporated in the schedule and in the berth schedule in order to take care of schedule deviations. Quite often agreements are made on demurrage charges (penalties related to delayed cargoes) if terminals need more time or if the shipping line arrives too late at the terminal. At the operational level the berth schedule is adjusted according to actual information. Quite often liner ships are too late. In 2015 Drewry shipping reports that ships are on average one day late. So the berth schedule is updated at a relatively short term (2 weeks) to take care of changing circumstances, while taking the tactical berth planning as a start.

1.2.2 Crane allocation

One level deeper than berth scheduling is the crane allocation. Cranes are used to move a container from the quay to the location where it will stored on the ship and vice versa. The storage locations on the ships are called locks. As container ships typically visit many ports, the cargo destined for a particular port will be distributed over many holds in the ship. After unloading, a ship will load cargo for several destinations which all have to be put in different ship holds. As a result the scheduling of the cranes is a difficult stochastic problem (handling times of containers are quite variable). The last crane to finish determines the moment when the ship can leave and hence the port time. A good balancing of the workload between the cranes is therefore necessary, but also very difficult to achieve. Another complication comes from the fact that port workers often work in shifts with fixed starting and end times, and a terminal will have to accommodate these restrictions.

1.2.3 Container stacking

A final aspect we like to mention is the stacking of the containers in the yard. The issue is not only that containers are stacked on top of each other, which complicates the retrieval of a bottom container, it is also the location on the yard of the containers to be loaded. If a ship moors right before the place where its (to be loaded) containers are located, then travel distances to the quay cranes are short and no bottlenecks are likely to occur. However, if a ship (due to delay or congestion at the berths) berths somewhere else, or if containers are spread out over the yard, then the terminal has to transport the containers over longer distances by which the loading could potentially be delayed. Container stacking is closely related to stowage planning, as the latter determines the order in which containers are to be loaded. In a perfect world one can take the order in which containers are stacked into account while making the load planning, but that creates a very complex problem, which also suffer from the variations in the loading. Hence costly reshuffles, where top containers are placed somewhere else to retrieve bottom containers are needed in large quantities.

1.3 Geographical bottlenecks

Canal restrictions form the main geographical bottlenecks for container ships. The Suez Canal and the Panama Canal are two well known canals imposing restrictions on container ships. The type of restrictions may differ between different canals. The main restriction imposed by the Suez Canal is for example the compulsory convoy passage through the canal. This results in long waiting times if a container ship misses the planned convoy. The Panama Canal on the other hand, imposes limits on the size of ships that want to sail through the canal.

Two other examples of geographical bottlenecks are the Strait of Malacca and the Gulf of Aden. These waterways are narrow, but are strategically important locations for the world trade, making them vulnerable to piracy.

Finally, ports may also impose geographical bottlenecks. Large ships might not be able to access certain ports, because the access ways have tight draft restrictions.

1.4 Operations research in liner shipping

In 1983, Ronen provided the first overview paper on the contribution of operations research methods in ship routing and scheduling. Since this first paper, every decade a follow-up overview paper appeared reviewing new research conducted in that decade (Ronen 1993, Christiansen et al. 2004, 2013). Initially, the reviews were mainly focused on the ship routing and scheduling problem, but more and more other shipping problems are included in these reviews. Furthermore, Christiansen et al. (2007) provides an extensive overview of maritime transportation problems. Finally, Meng et al. (2014) give an overview of research related to container routing and scheduling in the liner shipping industry in the last thirty years. The number of citations in these reviews has increased fast, showing the increasing interest in operations research in liner shipping problems.

Liners usually face complex problems, because the above discussed decision problems cannot be seen separately from each other and because problem instances are usually large. For example, when a liner company wants to determine its service network, it has to consider which effect the included routes will have on the cargo routing problem. The solution to the cargo routing problem depends on the underlying network and will influence the profit of that network. This increases the complexity of the problems faced by liners, since they need to solve multiple decision problems simultaneously. Furthermore, the number of ports that need to be included in a network is usually large (it can easily contain over 100 ports). The Indonesian case study in the next section will show that designing a network for only six ports is already quite difficult. Liner companies used to solve these problems manually, but in the last years computerized decision support systems became available. A well-known example of a successful decision support system is TurboRouter, a tool for optimizing vessel fleet scheduling (Fagerholt and Lindstad 2007).

1.5 Case study: Indonesia

Shipping is an important mode of transport in Indonesia because the country consists of many islands. Figure 1.1 shows six main ports in Indonesia. The six ports are located on five different islands of Indonesia, hence transport over land is only possible between Jakarta and Surabaya. Transportation between all other combinations of these cities is only possible by sea or air.

Figure 1.1: Location of six main ports in Indonesia

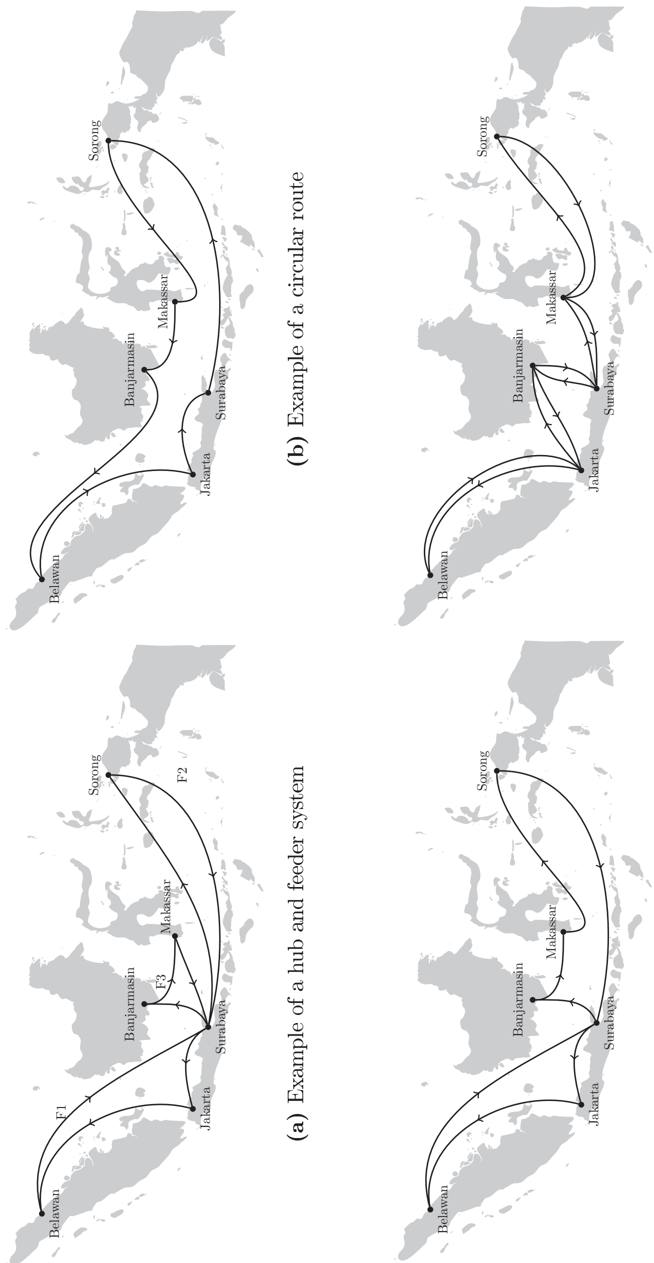
We will use the Indonesian case to illustrate some of the decision problems introduced in Section 1.1. The case considers six main ports in Indonesia, which is a very low number compared to the amount of ports considered in the problems in the remainder of this thesis. However, it allows us to illustrate some of the important properties of liner shipping, such as economies of scale and the efficiency of different route types. Furthermore, this illustrative example shows that it is already difficult to obtain an optimal route network for six ports, justifying the heuristics used in the later chapters.

We will assume that Table 1.1 gives the expected weekly demand in TEUs between the Indonesian ports. The last column and row give the row and column sums, denoting respectively the total supply from and demand to a port. The supply and demand values denote the number of containers leaving and arriving in the port respectively. The difference between demand and supply indicates how many empty containers have to be repositioned from or to the port. For the six ports in Indonesia, the empty container repositioning problem is of limited importance, since there are no large differences between supply and demand.

	Belawan	Jakarta	Surabaya	Banjarmasin	Makassar	Sorong	**Supply**
Belawan	-	6,500	1,000	100	75	25	**7,700**
Jakarta	6,750	-	2,000	4,000	2,800	450	**16,000**
Surabaya	1,000	2,500	-	3,750	4,800	2,150	**14,200**
Banjarmasin	100	3,600	3,500	-	10	0	**7,210**
Makassar	100	3,500	4,000	75	-	0	**7,675**
Sorong	50	650	2,000	0	0	-	**2,700**
Demand	**8,000**	**16,750**	**12,500**	**7,925**	**7,685**	**2,625**	**55,485**

Table 1.1: Expected weekly demand in TEU between the Indonesian ports (Source: own calculations)

Table 1.1 shows that Jakarta and Surabaya are the two ports with the largest container throughput, while trade with Sorong is relatively small. In this specific case, this might lead to problems, since Sorong is also located relatively far away from the other ports. Liner shipping companies prefer to offer services calling at the ports of Jakarta and Surabaya and consider it too costly to call at Sorong. By charging higher prices for containers that have to be transported from or to the port of Sorong, liners can make stops at Sorong more attractive. Hence, the liner company may use the pricing strategy to ensure that services calling at Sorong will also be beneficial. However, to determine exactly which prices they have to charge in order to maximize their profit, the liner company needs more details on the cost structure of the network they will provide.



(a) Example of a hub and feeder system

(b) Example of a circular route

(c) Example of a butterfly route

(d) Example of a pendulum route

Figure 1.2: Route examples for the Indonesian ports

Figure 1.2 shows examples of a hub and feeder system, a circular route, a butterfly route and a pendulum route calling at the six Indonesian ports. In the hub and feeder system of Figure 1.2a the port of Surabaya is the hub port, while Belawan, Jakarta, Banjarmasin, Makassar and Sorong are feeder ports. The route Surabaya - Jakarta - Belawan - Surabaya is referred to as F1. F2 is a direct feeder route between Surabaya and Sorong. The third feeder route, F3, calls at Surabaya, Banjarmasin and Makassar after which it returns to Surabaya. The circular route in Figure 1.2b has as characteristic that each port is called at exactly once during the round tour. Figure 1.2c shows the butterfly route Belawan - Surabaya - Banjarmasin - Makassar - Sorong - Surabaya - Jakarta - Belawan on which Surabaya is visited twice. Finally, in the pendulum route of Figure 1.2d all ports are visited twice, only the second time in reversed order.

	Belawan	Jakarta	Surabaya	Banjarmasin	Makassar	Sorong
Belawan	-	1,064	1,488	1,430	1,708	2,807
Jakarta	1,064	-	438	614	806	2,102
Surabaya	1,488	438	-	328	520	1,816
Banjarmasin	1,430	614	328	-	353	1,577
Makassar	1,708	806	520	353	-	1,375
Sorong	2,807	2,102	1,816	1,577	1,375	-

Table 1.2: Distances between the Indonesian ports in nmi (Source: www.ports.com/sea-route/)

Table 1.2 shows the distances in nautical miles (nmi) between the six Indonesian ports and Table 1.3 provides some characteristics of five ship types. Types 1, 2, 3 and 5 are obtained from Brouer et al. (2014a), while Type 4 is suggested by the Indonesian government and costs are obtained using interpolation. Note that the fuel usage in ton/day of Type 4 is larger than the usage of Type 5, because Type 4 has a higher design speed than Type 5. These data can be used to get some insight in the route cost using different ship types and network structures. In the calculations we use a simplified version of the fuel cost function as provided in Brouer et al. (2014a):

$$F_s(v) = 600 \cdot \left(\frac{v}{v_s^*}\right)^3 \cdot f_s \tag{1.1}$$

Here, $F_s(v)$ denotes the fuel cost in USD per day for a ship of type s sailing at a speed of v knots (nmi/hour). v_s^* and f_s are the design speed and fuel consumption in ton per day of a ship of type s sailing at design speed and can be found in Table 1.3. Remark that the bunker cost varies over time, but is assumed to be constant and equal to 600 USD per ton in this study (Brouer et al. 2014a). Table 1.4 shows the route distance in nautical miles, the duration in weeks, the frequency, the number of ships required to obtain the frequency and the sailing speed in knots for each route. Distances can be found by adding the distances of the individual sea legs, while the duration and frequency are manually fixed in this example.

Ship type	Capacity (TEU)	Charter cost (USD/day)	Draft (m)	Min speed (knots)	Design speed (knots)	Max speed (knots)	Fuel usage (ton/day)
1	900	5,000	8	10	12	14	18.8
2	1,600	8,000	9.5	10	14	17	23.7
3	2,400	11,000	12	12	18	19	52.5
4	3,500	16,000	12	12	18	20	60.0
5	4,800	21,000	11	12	16	22	57.4

Table 1.3: Data of the ship characteristics (Source: Brouer et al. 2014a)

Route	Distance (nmi)	Duration (weeks)	Frequency (per week)	Required ships	Speed (knots)
F1	2,990	2	1	2	11.33*
F2	3,632	2	1	2	12.61
F3	1,201	1	1	1	12.51
Circular	6,476	4	1	4	12.27
Butterfly	6,862	4	1	4	13.62
Pendulum	7,802	5	1	5	13.55

Table 1.4: Route characteristics for the different ships (an * indicated that the speed is outside the feasible range for some ship types)

In liner shipping it is common to use weekly port calls at a route. Route durations are typically an integer number of weeks such that an integer number of ships is needed to sail this route. For example, a route with duration three weeks and which is sailed by three ships, ensures a weekly frequency. Given the duration and frequency, the number of required ships can be found by taking the product of these two values.

Figure 1.3 shows the fuel cost in USD per nautical mile at different speeds for the five ship types. The fuel cost is a convex function, meaning that when the speed is doubled, the fuel cost per nautical mile is more than doubled. Hence, a constant sailing speed during the route will minimize the fuel cost. The speed is calculated under the assumption that every port call takes 24 hours and the durations as given in Table 1.4. The following

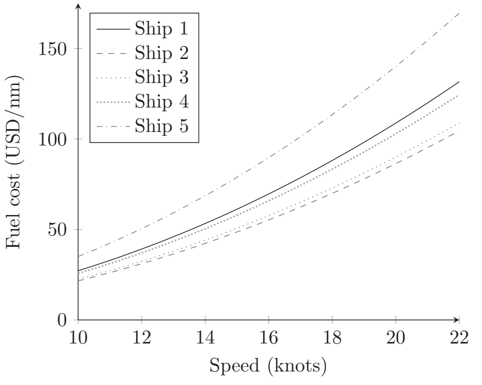


Figure 1.3: Fuel cost in USD per nautical mile

formula can then be used to determine the speed on each route:

$$v = \frac{\delta}{168 \cdot t - 24 \cdot n}, \tag{1.2}$$

where δ is the route distance in nautical miles, t the route duration in weeks and n the number of port calls on the route. An * in the column denoting the speed of Table 1.4 indicates that the speed is outside the feasible speed range for some ship types. The frequency is chosen in such a way that it is feasible for each ship type when sailing at maximum speed. Hence, the necessary speed can only be lower than the minimum speed of the ship type, in which case the ship will sail at minimum speed and will wait in one of the ports to obtain a weekly frequency.

Table 1.5 shows the weekly cost in USD for each of the routes given the frequency and duration as given in Table 1.4. The route costs consist of three components: the fixed ship costs, the port call costs and the fuel costs. When a liner company needs three ships to satisfy the required route duration and frequency, it bears weekly the fixed ship costs of all these three ships. Hence, the fixed ship cost is given by $7 \cdot S \cdot c_s^f$, with S the number of required ships and c_s^f the daily fixed ship cost of type s, which can be found in Table 1.3. The port call cost is the sum of the port fees of the ports visited on the route. If we assume that all port fees are the same, the port call cost is given by $F_p \cdot n \cdot q$, where F_p is the port fee per port visit and q the route frequency. In this example, we assume that $F_p = 650$ USD. The fuel cost is given by the product of the frequency, the number of days that a ship needs to sail one round tour and the fuel cost per day: $q \cdot \frac{\delta}{24 \cdot v} \cdot F_s(v)$, where $F_s(v)$ is the fuel cost in USD per day when sailing at speed v as given by (1.1). Consider a liner route with a duration of two weeks to which four ships are allocated. Each port on the route will then be called twice a week, resulting in a frequency of twice a week. Each ship needs two full weeks to sail a round tour, so in one week it will sail half of the route. Since there are four ships allocated to the route, in total two full round tours are made during a week (since the frequency is two). This explains the multiplication with the frequency in the fuel and port call cost. The total route cost in USD per week is now given by:

$$c_s^r = 7 \cdot S \cdot c_s^f + q \cdot \frac{\delta}{24 \cdot v} \cdot F_s(v) + F_p \cdot n \cdot q, \tag{1.3}$$

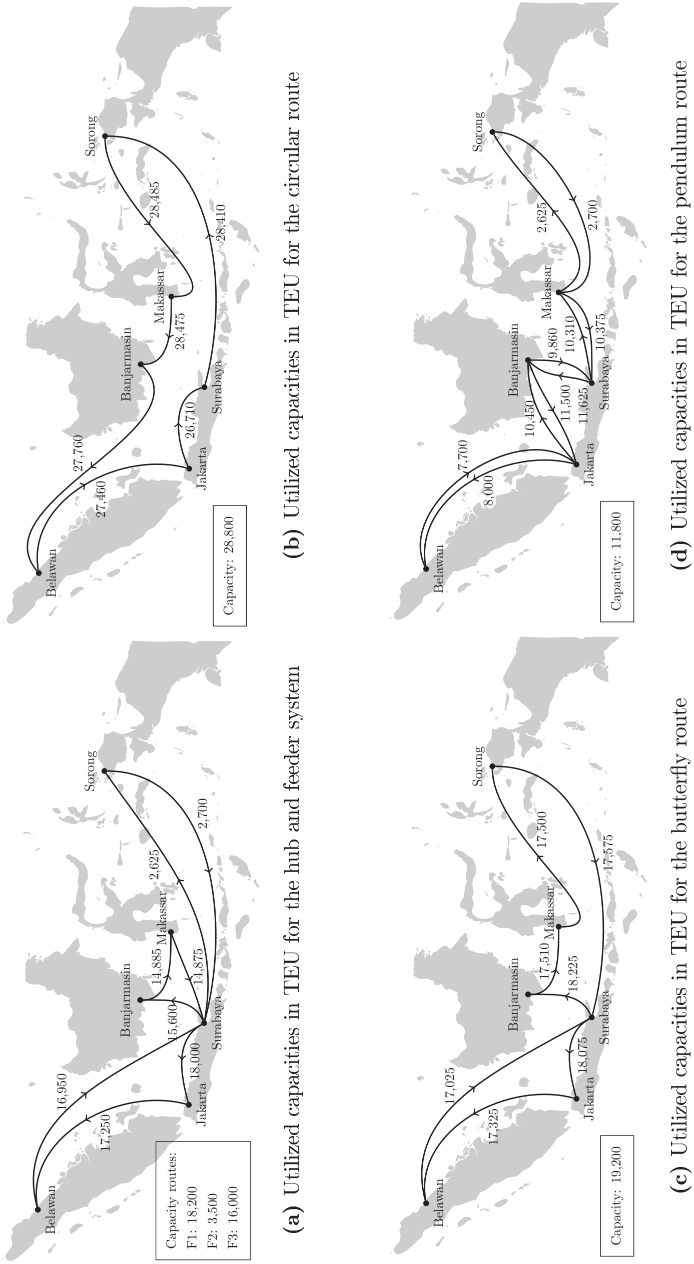
where c_s^r is the route cost in USD per week for a ship of type s. Doubling the capacity of a ship will not result in a doubling of the weekly route cost. This illustrates the effect of economies of scale: larger ships will in general have higher total costs, but lower costs per TEU, which is also exemplified in Table 1.6 by showing the weekly route cost per TEU under the assumption that the ship is fully utilized. The table also shows that the effect of economies of scale can differ quite a lot between ship types.

	Cost in USD/week				
Route/Ship	Type 1	Type 2	Type 3	Type 4	Type 5
F1	176,268	196,765	252,848	336,691	446,793
F2	228,411	238,026	285,297	373,868	497,669
F3	88,076	98,537	121,253	162,296	214,803
Circular	408,876	438,257	531,147	702,468	933,207
Butterfly	490,525	503,210	598,818	779,713	1,038,189
Pendulum	571,488	596,234	714,297	935,318	1,243,643

Table 1.5: Route cost per week for the duration and frequency as given in Table 1.4

	Cost per TEU in USD/week				
Route/Ship	Type 1	Type 2	Type 3	Type 4	Type 5
F1	196	123	105	96	93
F2	254	149	119	107	104
F3	98	62	51	46	45
Circular	454	274	221	201	194
Butterfly	545	315	250	223	216
Pendulum	635	373	298	267	259

Table 1.6: Economies of scale in ship size at full utilization



(a) Utilized capacities in TEU for the hub and feeder system

(b) Utilized capacities in TEU for the circular route

(c) Utilized capacities in TEU for the butterfly route

(d) Utilized capacities in TEU for the pendulum route

Figure 1.4: Route examples with utilized capacities

The disadvantage of the circular route is that the capacity can not be utilized efficiently. When containers from for example Surabaya to Jakarta are transported using the circular route, they will be on board of the ship on all sea legs except the leg from Jakarta to Surabaya. Butterfly routes are better able to utilize the available capacity, since some ports are visited twice on a round tour. In the butterfly route, the ports of Surabaya and Jakarta are visited directly after each other, such that the containers are only on board during one sea leg of the route. The pendulum route visits all ports twice, hence it needs the lowest capacity. In a hub and feeder network, usually many ports are connected by only one or a few sea legs. This ensures that hub and feeder networks are able to utilize the available capacity very efficiently. Figure 1.4 shows the utilized capacity at each sea leg in the four different route networks under the assumption that all demand has to be satisfied using only the given network. For the butterfly route, we assumed that the containers that have to be transported from Makassar to Banjarmasin will stay on board of the ship during the route segment Surabaya - Jakarta - Belawan - Jakarta - Surabaya. Alternatively, these containers can be unloaded during the first call at Surabaya and loaded again during the second port call at Surabaya in which case transshipment costs at Surabaya are incurred. The utilized capacities are found by adding all container flows that need to traverse the given sea leg in order to arrive at their destination. Table 1.7 shows the required capacity in TEU for each route, the number of port calls per week for each ship type in order to have enough capacity to satisfy all demand, the available capacity in TEU using these ship types and the total route costs in USD per week. The required capacity is found by taking the maximum utilized capacity of the route. Next, we make a combination of ship types such that enough capacity is available at each route. Given these ship allocations, the total route cost can be found by multiplying the weekly route cost for a ship type by the number of port calls per week divided by the route frequency. Note that the type and number of ships needed vary a lot between the three different route structures. For the hub and feeder system, $2 \cdot 1 + 1 \cdot 1 = 3$ ships of type 2 (since feeder route 1 has a duration of 2 weeks and feeder route 3 has a duration of one week), $2 \cdot 2 + 2 \cdot 1 = 6$ ships of type 4 and $2 \cdot 2 + 1 \cdot 3 = 7$ ships of type 5 are needed. The circular route uses $4 \cdot 6 = 24$ ships of type 5, while the butterfly route uses $4 \cdot 4 = 16$ ships of type 5. Finally, the pendulum route uses $5 \cdot 2 = 10$ ships of type 4 and $5 \cdot 1 = 5$ ships of type 5. Hence, the optimal solution to the fleet size and mix problem is highly dependent on the network structure.

Table 1.7 also shows the total network cost for the hub and feeder system, the circular route, the butterfly route and the pendulum route. The table indicates that the hub and feeder system and the pendulum route are by far the cheapest choices of networks in this

Route	Req. cap. (TEU)	Port calls per week					Av. cap. (TEU)	Cost (USD/week)
		Type 1	Type 2	Type 3	Type 4	Type 5		
F1	18,000	0	1	0	2	2	18,200	1,763,733
F2	2,700	0	0	0	1	0	3,500	373,868
F3	15,600	0	1	0	0	3	16,000	742,948
HF-Total	36,300	0	2	0	3	5	37,700	**2,880,549**
Circular	28,485	0	0	0	0	6	28,800	**5,599,239**
Butterfly	18,225	0	0	0	0	4	19,200	**4,152,755**
Pendulum	11,625	0	0	0	2	1	11,800	**3,114,280**

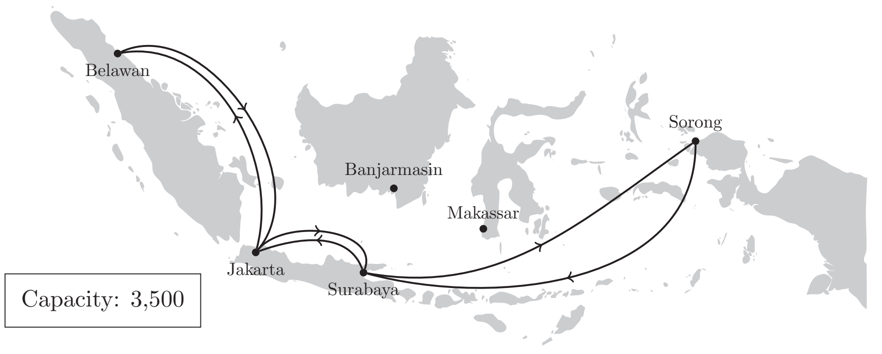
Table 1.7: Network cost per week when shipping all demand

example. They both cost approximately 3 million USD per week, while the circular and butterfly routes cost respectively about 5.5 and 4 million USD per week. One remark has to be made: in the hub and feeder system, a lot of containers need to be transshipped, adding additional costs that are not included in this example. In total 15,450 containers have to be transshipped per week in the hub and feeder system. If a transshipment costs for example 100 USD per container, the total cost of the hub and feeder system will rise to almost 4.5 million USD per week. Hence, the hub and feeder system will then have higher costs than the butterfly and pendulum routes. Of course, one could also make route networks with combinations of these routes, which might be more cost efficient.

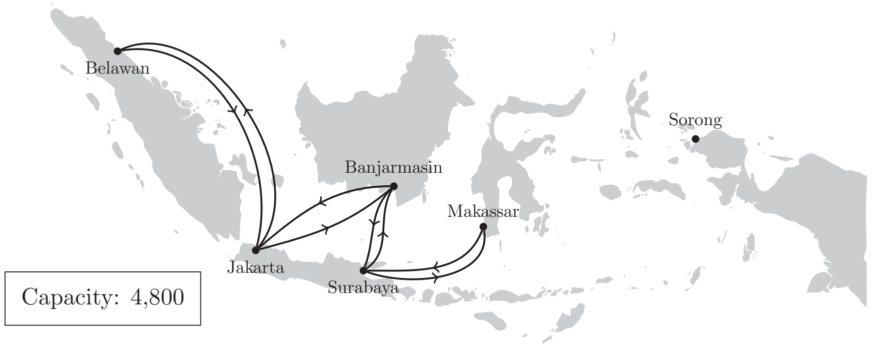
The good performance of the hub and feeder system and pendulum route is (partly) caused because of the better utilization of capacity in the hub and feeder system. Another advantage of hub and feeder systems is that liners can allocate different ship types to the different types of routes. Feeder ports usually have less demand than hub ports, hence it makes sense to allocate smaller ships to the feeder routes than to the main routes. If all ports are visited on similar routes, like circular, butterfly and pendulum routes, all these ports are visited by the same ship type. Hence, large ships might visit very low demand ports if these ships are able to berth in the smaller ports (smaller ports might have stricter draft restrictions than hub ports). Otherwise many small ships are needed in order to satisfy the demand of the large ports. However, a disadvantage of hub and feeder networks is that usually many transshipments are needed in order to satisfy the demand, which increases both the transportation price and transit time. In airline passenger transport, hub and feeder systems are very popular; an important reason for this is that transshipments are made by passengers at no apparent cost.

Finally, we determine the profit and efficiency of the networks when we assume that each container will generate a revenue of 200 USD if it is transported, (un)loading and

(a) Route 1



(b) Route 2



(c) Route 3

Figure 1.5: Optimal route network

transshipment costs are all 40 USD per container. Recently, the problem is also studied on request of the Indonesian government, resulting in a single pendulum route to be sailed. This pendulum route is also known under the name Pendulum Nusantara. We use the mixed integer programming model proposed in Chapter 3 to determine the optimal route network given an initial set of routes. Routes are constructed in the following way using the ordering of ports used for the pendulum route. Ports may be visited at most twice during a route: once on the eastbound trip and once on the westbound trip. All feasible routes are generated and given as input to the mixed integer programming problem, which makes use of a path formulation to solve the cargo allocation. Table 1.8 shows the profits of these networks and Figure 1.5 shows optimal route network. We see that the pendulum route performs indeed better than the hub-and-feeder network and the butterfly and circular routes. The efficiency of the networks is measured by the shipped distance in nmi per TEU. The shipped distance for direct shipping is equal to 836.57 nmi/TEU. Table 1.8 shows that the pendulum and optimal networks are both efficient networks with respect to shipped distance. Furthermore, the hub-and-feeder network is much more efficient than the circular and butterfly routes as expected.

The optimal route network as shown in Figure 1.5 consists of two pendulum routes (Routes 2 and 3) and a non-stop service (Route 1), which is a special type of pendulum route with only two port calls. The pendulum route structure ensures efficient transportation between all demand pairs. All routes have a frequency of once a week, reducing the number of required ships.

Network	Shipped distance (nmi/TEU)	Profit (USD)
Hub-and-feeder	1,428.06	3,159,651
Circular	3,269.79	1,058,961
Butterfly	2,227.57	2,284,445
Pendulum	996.80	3,642,916
Optimal	852.66	4,897,109

Table 1.8: Efficiency and profit of the different networks

1.6 Outline of the dissertation

In the Indonesian case study in the previous section, the goal is to design a route network for a liner company that maximizes the profit. For this problem, the distances, demand and revenue between ports are given together with a set of available ship types and their characteristics and port, (un)loading and transshipment costs. Using the data, a set of routes needs to be constructed, a ship and frequency have to be allocated to each route and the demand has to be transported through the network taking into account the ship capacities. In this problem, the network design and cargo routing problems are combined. Furthermore, speed optimization at the tactical level can be used to obtain weekly frequencies. Furthermore, the Indonesian case study also incorporates the fleet size and mix problem, because there is no limit on the number of available ships of each type. Chapter 2 considers the same problem for the Asia-Europe trade lane with 58 ports. The Indonesian case study shows that the problem of designing a route network is already difficult to solve when only six ports are considered. When the number of ports increases, the complexity of the problem will also increase. Therefore, we first cluster the ports into a smaller number of port clusters. We develop heuristic methods to design networks calling at the port clusters on main routes and introduce regional or feeder routes to transport containers from their origin to a port cluster and from port clusters to their destination. In this chapter, initial networks are generated at random and high performing networks are changed and combined in order to improve the current solution. Improvement heuristics in which feeder ports are inserted in a main route are used to obtain better networks. Chapter 3 considers a similar problem, but now an initial fleet is given, hence the fleet size and mix problem is not considered any more. In this chapter, we focus on the influence of large container ships on the number of port calls per route. Since large container ships are mainly benefiting from economies of scale at sea, routes with fewer port calls might be preferred over the traditional routes with many port calls. Again, we reduce the problem size by clustering ports, but this time an iterative heuristic is used to create networks. In this heuristic, the regional and hub networks are improved alternatively, until the resulting network does not change any more. The obtained network is compared to the current best networks in literature to assess the influence of reducing the number of port calls on the network performance. Note that the clustering phase has different implications in Chapter 2 than in this chapter. In both chapters, hub ports are selected before designing the networks and these hubs will not change anymore during the design phase. However, in Chapter 2 feeder ports can be reallocated to a main route, while in Chapter 3 only hubs can be visited on the hub routes of the final netrwork.

Consider one route from the network found in the Indonesian case study. The costs of this route are evaluated under the assumption of a fixed sailing speed leading to a weekly frequency. However, these costs might not be accurate when sailing the route. During the execution of the route, delays will be incurred. If liners adhere to the fixed sailing speed as determined in the case study, each incurred delay will immediately result in a delayed arrival or departure in a port. Furthermore, delays will be accumulated, resulting in very high arrival and departure delays. In general, liner companies have two possibilities to manage these delays. First, a liner can for example publish a timetable with 75 hours to sail from Belawan to Jakarta. However, using the preferred speed of 15 knots, only 71 hours are needed to sail this distance. The remaining 4 hours are referred to as buffer time and are used to absorb incurred delays. The total amount of buffer time included in a route is limited, because more ships are needed when too much buffer time is included. Adding an additional ship to a fleet is expensive, as can be seen from the data in the previous section. The second way to manage delays is to undertake an action to recover from incurred delays. A good example of such a recovery action is to adapt the sailing speed at certain sea legs. Ships can speed up to ensure a timely arrival, but then it will incur higher fuel costs. The best results are obtained when liners use both methods to manage delays: including additional buffer time in the schedule and use recovery actions during the execution of the route when delays are incurred. Hence, the liner shipping company has to make a trade-off between the inclusion of buffer times at a tactical planning level and the cost of recovery actions at the operational planning level. Chapters 4 and 5 both consider this robust schedule design problem. In these chapters, the goal is to jointly determine the arrival and departure times for the ports in a given route and the recovery policy that has to be performed when ships are delayed such that the total costs of operating the route are minimized. Chapter 4 introduces a mixed integer programming formulation together with heuristics to solve this problem. In Chapter 5 we are able to prove some theoretical results under the assumption that cost functions of delays and recovery actions are convex. These results ensure that we can determine subgradients of the cost function. We propose an exact algorithm using these subgradients to solve the problem.

Finally, Chapter 6 summarizes the main conclusions of each of the individual chapters.

1.7 Clarification of contribution

The chapters in this dissertation are based on papers that are published or (going to be) submitted to scientific journals. Each chapter introduces its own notation and contains its own introduction, literature review and conclusion and can therefore be read separately.

The data used in this dissertation varies between the chapters. In Chapter 2, we gathered data from recent publications and master theses. Since some parts of the data are highly confidential, missing data were estimated in the best possible way. The data with respect to demand, revenue, handling and transshipment costs, port call costs and shipping costs used in the other chapters of this dissertation are all obtained from the recent benchmark suite from Brouer et al. (2014a).

This dissertation is the result of a collaboration between myself, my promotor and my copromotor. For each chapter, the contributions were as followed:

- Chapter 1: The majority of the work in this chapter has been done independently by myself under close supervision of the promotor. The case study in this chapter is based on the theses of Hakan Kalem, Jeroen Meijer and Lisanne van Rijn. Section 1.2 has been written by the promotor and adjusted by myself.

- Chapter 2: This chapter is based on my master thesis research conducted under supervision of my promotor. Most data in this chapter is gathered by Stephan Lachner and Victor Boskamp during their master thesis. My promotor provided feedback which is incorporated in the chapter and in the paper on which the chapter is based. This chapter has been published in the European Journal of Operational Research.

- Chapter 3: The majority of the work in this chapter has been done independently by myself under supervision of the promotor.

- Chapter 4: The majority of the work in this chapter has been done independently by myself under close supervision of the promotor. This chapter is the result of many brainstorming sessions with my promotor. Norodin Ty has implemented some of the methods in this chapter during an internship at Maersk Line. In a later stage, Mehdi Sharifyazdi and my copromotor also contributed to this chapter by providing feedback, which has been incorporated.

- Chapter 5: This chapter is the result of a tight cooperation between myself and my copromotor under close supervision of my promotor. My copromotor and I worked closely together on the numerous proofs and the methodology described in this chapter.

Chapter 2

Methods for strategic liner shipping network design[1]

2.1 Introduction

Seaborne shipping is the most important mode of transport in international trade. More than 80% of the international trade in 2010 is transported over sea (UNCTAD 2010). In comparison to other modes of freight transport, like truck, aircraft, train and pipeline, ships are preferred for moving large amounts of cargo over long distances.

In the shipping market, three types of operations are distinguished: tramp shipping, industrial shipping and liner shipping (Lawrence 1972). Tramp ships do not have a fixed schedule and are used for immediate deliveries where the most profitable freight is available. Therefore, the activities in tramp shipping are very irregular. In industrial shipping the cargo owner controls the ship and the objective becomes to minimize the cost of shipping. In liner shipping, ships follow a fixed route within a fixed time schedule; this is most common in the container trade.

The decision making in liner shipping can be distinguished on three different levels: the strategic, tactical and operational planning levels (Agarwal and Ergun 2008). In the strategic planning level the optimal fleet-design is determined. This means that both the optimal number of ships in a fleet and the optimal ship sizes are determined in this level. This stage is very important, because the capital and operating cost in the (liner) shipping industry are very high. The ship-scheduling problem is solved in the tactical planning level. In this level, the service network is designed by creating ship routes and

[1]This chapter is based on the following article with small modifications: Mulder, J. and Dekker, R. Methods for strategic liner shipping network design (2014) published in volume 235 of the European Journal of Operational Research on pages 126-141.

allocating the available ships to these routes. Finally, in the operational planning stage, it is determined which cargo is transported and which route(s) are used to ship the cargo. This problem is also referred to as the cargo-routing problem. The decisions made in a planning level influence the decision making in the other levels. Therefore, it could be profitable to solve the problems on the different levels simultaneously.

2.1.1 Literature review

Over the last decades, maritime transport has become a more popular field of research. In 1983 the first survey on ship routing and scheduling was published (Ronen 1983). This survey gives a detailed overview on the research performed on ship routing and scheduling in the period before 1983. In a sequel, Ronen (1993) provides a detailed summary of published research on ship scheduling and related problem in the period from 1983-1993. Next, the survey of Christiansen et al. (2004) describes the major developments in the ship routing and scheduling problems in the period from 1994-2004. Finally, Wang and Meng (2011) give an overview of the most important existing literature on liner shipping studies and propose directions for further research.

Little research is performed on the determination of the optimal fleet design. Fagerholt (1999) develops a 3-phase solution approach to optimize the fleet size in liner shipping networks in which all feasible routes are generated and combined. Thereafter, the optimal fleet size is determined using a set partitioning problem, which is also solved in this phase. Powell and Perkins (1997) use an integer programming model to optimize the fleet deployment for a liner shipping company. They compare the results to the results obtained with a linear programming model. When using a linear programming model, manipulation of the results is needed to guarantee integer solutions. Both solution approaches become time consuming when the problem becomes larger.

Song et al. (2007) discuss a cargo allocation model with two objectives. The first objective is to minimize the unassigned cargo volume. The second objective is to minimize total costs corresponding to a given minimal unassigned cargo volume. Because the model is very difficult to solve by analytical methods, the solution space is first truncated. Thereafter, the authors select priority rules and make use of heuristics to find solutions of the model.

Most research is related to the combined ship-scheduling and cargo-routing problem. First, Rana and Vickson (1991) and Fagerholt (2004) present integer programming problems to solve the combined ship-scheduling and cargo-routing problem. They are able to solve small instances in a reasonable amount of time, but for larger instances their methods become too time consuming.

Next, some research exists in which mathematical programming methods, like Benders decomposition, are used to solve the combined ship-scheduling and cargo-routing problem (see for example Agarwal and Ergun 2008, Álvarez 2009, Gelareh and Pisinger 2011). These methods can be used to solve some very small instances to optimality, but for larger instances heuristic methods are still needed.

2.1.2 Contribution and outline

The objective of this study is to develop a service network in liner shipping. The service network should consist of a set of routes, the allocation of ships to the routes, the sailing speed of the ships on each route and the allocation of cargo over the routes. We assume that the liner shipping company is free to select the ships it needs to cover the routes, but we impose a maximum number of routes to which certain ship sizes can be allocated to prevent that only very large ships will be allocated in the network. In other words, our goal is to solve the fleet-design, ship-scheduling and cargo-routing problems simultaneously, where we consider the case with limited availability of ships.

In comparison to the current literature, our main contribution is to provide a solution approach for the integrated network design problem, in which we incorporate methods for fleet-design, ship-scheduling, speed-selection and cargo-routing. Since large liner shipping companies have to change their networks quite often, this integrated problem is essential in the current economy. In our test set, substantial savings can be obtained by solving the network design problem in an integrated way.

This paper is organized as follows. In Section 2.2 the problem is defined in more detail. Next, in Section 2.3 the methods used to solve the combined problem are proposed. Section 2.4 describes results from a case study and in Section 2.5 the main conclusions from this research are drawn.

2.2 Problem formulation

We consider the combined fleet-design, ship-scheduling and cargo-routing problem with limited availability of ships. In this section, first the three individual problems are described. Thereafter, a formulation of the combined problem is given.

2.2.1 Fleet-design problem

The goal of the fleet-design problem is to determine the optimal composition of the fleet. In this problem, both the number and the size of ships in the fleet have to be determined. For the shipping company it is important to determine the optimal fleet design, because the costs related to the fleet are very high. Costs related to the fleet composition can be distinguished in two types: fixed cost (e.g. Capital Expenditures (CAPEX)) and variable cost (e.g. Operating Expenditures (OPEX)).

In this paper, we investigate the optimal fleet design of a liner shipping company, so we do not consider an initial fleet. However, since liner shipping companies already have an existing fleet, we limit the freedom of the fleet design in such a way that companies can change their fleet towards the obtained fleet without having to replace many ships before they are depreciated, which is very costly. Hence, the number of routes that can be performed with ships is limited for each ship size. Finally, we assume that all ships are available at the beginning of the planning period.

The underlying route network and demand have to be considered when determining the fleet composition of a liner shipping company. However, the fleet design is determined for 10-20 years, because of the high cost incurred by replacing a ship. In such a period, the demand structure can change, which can cause changes in the route network. Therefore, when determining the optimal fleet design, both present and future demand have to be considered.

Economies of scale are another important factor in purchasing new ships. Larger ships usually have lower transportation cost per TEU than smaller ships. However, the fixed cost of larger ships are higher than that of smaller ships. The demand on the route that the ship will serve also influence the decision of the ship size.

2.2.2 Ship-scheduling problem

In the ship-scheduling problem, the service network has to be designed. A service network consists of a set of ship routes and the allocation of ships to the routes. Furthermore, the optimal sailing speed has to be determined for each ship route. A ship route is a sequence of ports that are visited by a ship. The ship routes are cyclic and consist of a westbound and an eastbound trip. Each port can be visited at most once on the westbound trip and at most once on the eastbound trip. So, ports can be visited twice on a ship route.

The allocation of ships to routes can be restricted, because for example a port on the route cannot handle a certain type of ship. Once a ship is allocated to a certain route, it will serve this route during the whole planning horizon. Most shipping companies operate schedules in which each route is served once a week to maintain a customer base and to provide customers with a regular schedule (Agarwal and Ergun 2008). Therefore, in general the number of ships needed for one ship route has to be at least equal to the number of weeks needed to complete an entire round tour (rounded above). In this paper we will also require weekly route schedules.

2.2.3 Cargo-routing problem

In the cargo-routing problem, the shipping company makes two decisions. They decide which demands they accept and which routes are used to transport this cargo from the origin to the destination port. When the cargo-routing problem is solved as an individual problem, the service network is assumed to be known beforehand. Our goal is to maximize the profit for one shipping company, so competition is not investigated. Revenues are obtained by transporting cargo between their origin and destination port. However, costs are also incurred by the transportation of the cargo. For some demand pairs the revenue that can be obtained will not exceed the cost incurred by transporting the cargo. This demand will then be rejected by the shipping company. Furthermore, it is possible that some profitable demands are rejected because other demands are more profitable.

When the demand of a demand pair is (partly) satisfied, the cargo will be picked up in the origin port and delivered at the destination port. When the origin port is visited on several ship routes, it has to be determined to which route the cargo is allocated. The same holds for the destination port. Some origin and destination ports will be visited on the same ship route, while other cargoes have to be transshipped to other routes for which costs are incurred. All these decisions are made in the cargo-routing problem.

2.2.4 Combined fleet-design, ship-scheduling and cargo-routing problem

The decisions made in the three individual problems affect the decision making in the other problems as well. For example, when the service network is determined in the ship-scheduling problem, the network structure and capacity limits for the cargo-routing problem are set. This implies that a bad choice of service network in the ship-scheduling phase can result into lower profits in the cargo-routing phase. Therefore, it may be profitable to consider the individual problems at the same time.

In the combined fleet-design, ship-scheduling and cargo-routing problem, all decisions explained above in the three individual problems have to be taken at the same time. The problem becomes to construct a service network and determine the routes used to transport cargo such that the profit is maximized given a certain demand matrix and cost/revenue data. In this research, we limit the available ships that we can use in the service network. Therefore, we only try to select the best ships out of the available ships instead of solving the fleet-design problem from scratch. Here, we assume that the possible ship types are given with known characteristics.

In this paper we only include ship-related cost into the combined network design problem, which is essential to answer strategic questions for liner shipping companies. In practice, liner shipping companies also incur several other costs, such as administrative costs, costs related to the sales organization or headquarters. However, since we have no data on these costs, we leave them out of consideration.

2.3 Solution algorithm

In this section, we discuss the main ideas of the methods used to solve the combined fleet-design, ship-scheduling and cargo-routing problem. For a more detailed description of the methods, we refer to Appendix 2.E. Since the combined fleet-design, ship-scheduling and cargo-routing problem is too large to solve efficiently and there is a lot of interaction between the different subproblems, we choose to aggregate the ports in the model and use a composite approach to solve the combined problem. Thereto, in Section 2.3.1, the method to aggregate ports is described. Next, in Section 2.3.2 we propose a method that generates initial feasible ship schedules and in Section 2.3.3 we construct a model that is able to optimally allocate the cargo to the available ships given these ship schedules. After solving the cargo routing model, the aggregated ports have to be disaggregated, which is explained in Section 2.3.4. To do this, we introduce feeder services to ship the cargo between the central port in a port cluster and all other ports of the same cluster. Finally, the genetic algorithm, described in Section 2.3.5, will generate new feasible ship schedules and this process will be repeated until a certain stopping criterion is met. Figure 2.1 shows a schematic overview of the solution algorithm.

2.3.1 Aggregation of ports

In Section 2.3.3, we will describe a linear programming model that can be used to solve the cargo routing over the ship routes. When the problem instance becomes larger, the computational time of the cargo routing model increases rapidly. Thus, for large problem instances, it is very time consuming to solve the cargo routing model repeatedly. The size of the problem instance decreases when the number of ports is reduced. Therefore, the

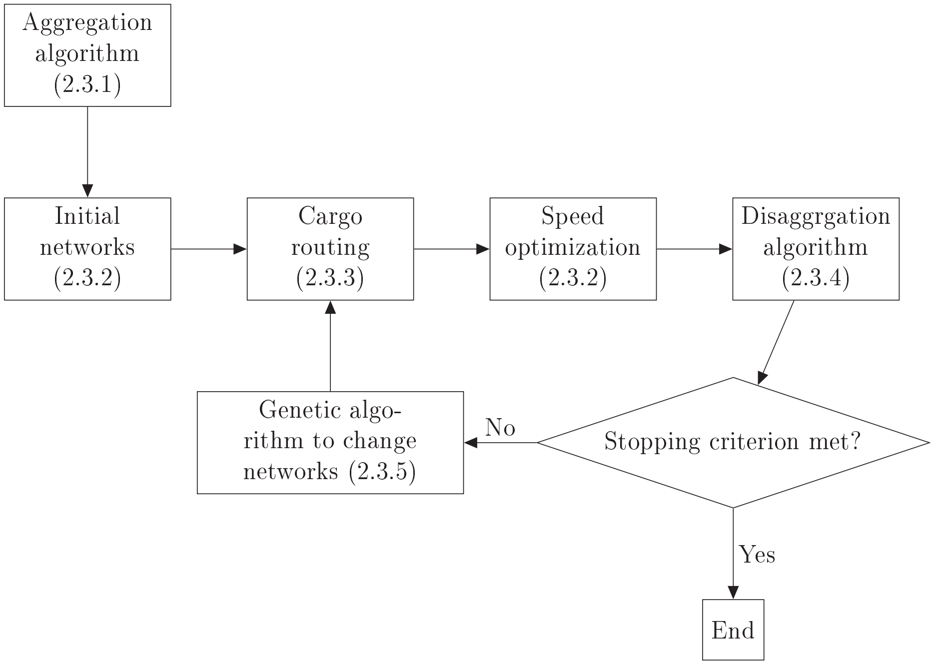


Figure 2.1: Schematic overview of the solution algorithm

computational time of the cargo routing model can be decreased by reducing the number of ports. It depends on the computational time of the cargo routing model to which number of ports the problem has to be reduced.

In a model with aggregated ports, ships stop only once per cluster. For each port cluster, the stop should always be at the same place. Therefore, three major decisions have to be made during the aggregation process. First, the ports that are aggregated into a port cluster have to be determined. Next, one of the ports in a cluster has to be chosen as the central port (the port of the cluster where ships will make their port visit). Finally, the data of individual ports have to be aggregated to port cluster data.

Ports are aggregated based on their mutual distance. Ports that are relatively near to each other are clustered. An upper bound on the distance between two ports that belong to the same cluster can be imposed. To avoid problems, we consider the distance of a port that we will add to the cluster to the central port in the comparison. Thus, first the central port of the cluster has to be determined. Thereto, we make three lists of ports based on their expected yearly throughput. If a port has an expected yearly throughput that is at least twice as large as the average expected yearly throughput over all ports, the port is added to the list containing central ports. Furthermore, a list of ports that are not allowed to be central ports (noncentral ports) contains all ports that have a throughput that is at most one fifth of the average throughput. The last list consists of all remaining (intermediary) ports. At the end of the aggregation phase, when it is known which ports will be clustered into one port cluster, the decision of the central port in the cluster is reconsidered. The final central port in the cluster will be the port for which the expected cluster transshipment costs are minimized. The expected cluster transshipment cost will be determined by the product of the transshipment cost per TEU of the central port and the total expected yearly throughput of all noncentral ports in the cluster (as long as the maximum distance restriction is still satisfied). These lists can now be used to design the clusters as will be explained in the remainder of this section.

First, we create a cluster for each port on the central port list. The ports of the other two lists that are within the maximum distance to their closest central port are added to the corresponding cluster. If there are intermediary ports remaining that are not yet added to one of the clusters, a new cluster is created for the largest intermediary port, which becomes a central port. All intermediary and noncentral ports that are closest to this port (compared to the other central ports) and within the maximum distance are added to this new cluster (if a port was already allocated to one of the other clusters, it is removed from this old cluster). This is repeated until all intermediary ports are allocated to a cluster. Finally, if there are still some noncentral ports remaining, they are allocated to the cluster they are closest to. Note that in this last step the maximum distance between ports in the same cluster is exceeded. However, since these ports have very little demand, it is probably quite costly to visit them on the main routes. Ships on the main network are generally larger (and thus more expensive) than ships on feeder services. Furthermore, the additional handling costs on the feeder network cannot be high, since the demand is low. Therefore, we decide to add these ports to an existing cluster instead of introducing a new cluster for them, forcing them to be visited on the main network.

Thus, the central port of a cluster is mainly chosen based on a combination of expected yearly throughput and transshipment cost. However, the distance to other ports also indirectly influences the central ports, because a maximum distance from a central port to all other ports in the cluster is imposed. Therefore, ports in regions with less throughput are still candidates to become a central port. Probably, some ports exist that are preferred to become central ports based on their location (such as Singapore), but are located in a region with less throughput. These ports will become central ports if they are the largest port in their region (unless the transshipment costs for these ports are too high).

However, it is possible that another port in the region is larger, but has a worse location. In this case, our algorithm will select the larger port to be the central port. If we want to avoid this, it is possible to include a distance measure, that gives an indication of the amount of additional nautical miles that has to be sailed to visit the port on a route, in the determination of central ports. Furthermore, noncentral ports that are added to a cluster will in first instance be visited on a feeder network. However, in our solution approach, they can be added to the main route network in a later stage of the algorithm. Thus, if a noncentral port is located along a main route, this noncentral port will probably be added to the main route, saving (a part of) the feeder costs to serve this port on the feeder network.

Finally, we have to aggregate the individual port data into port cluster data. Relevant port data in the model are: distance, demand, port cost, transshipment cost, (un)loading cost and port time. The distance, costs and port time only depend on the port at which the ship stops. Therefore, for these data the port cluster data is the same as the individual port data of the central port. The demand data also depends on the demand of the other ports in the cluster. Cluster demand equals the sum of the individual port demand in the cluster. Note that demand between ports in the same cluster disappears during the aggregation process. This demand can be reviewed after the disaggregation process.

2.3.2 Designing an initial liner shipping network

In our research we consider routes in which ports can be visited at most once on the eastbound and at most once on the westbound trip. Furthermore, we clustered ports into port clusters, so we reduced the maximum number of visits on a route. Since there are not that many port clusters, they are generally not very close to each other, such that a natural geographical order can be obtained. Using this order, we generate initial routes and determine the optimal speed on each route.

Generate initial routes

Initial routes are generated at random. First, we determine randomly the number of routes in the network between a minimum and maximum. Thereafter, for each route in the network, a cluster is called with a certain probability. Thus, we determine in geographical order at random whether a cluster will be visited or not. The routes obtained using this method are not always feasible routes. Therefore, we check during the generation of the routes whether they satisfy the following three conditions and change them when necessary.

The first condition is that the beginning cluster of each route should be equal to the end cluster, so that a round tour is made. If this is not satisfied, the beginning or end cluster is adjusted, so that this condition is met. The second condition is that the two middle clusters of the routes should be unequal, because otherwise the same cluster is visited twice in a row. we remove one of the middle clusters if this condition is violated. The last condition is that the route length is at least equal to the minimum route length. If this condition is violated for a certain route, this route is deleted.

Finally, a ship type is allocated to the ship route. Thereto, we select randomly a ship type from the set of available ship types. Note that this set has to be updated when the last available ship of a certain type is allocated to a route. In this case, the ship type is removed from the set of available ship types.

Using the above procedure a network is obtained with a random number of feasible routes. This network can be used to run the cargo allocation model and obtain the different flows. Later on, the networks will be changed using a genetic algorithm based method, so that better networks are constructed.

Determine optimal speed

When the routes and capacities of the liner ships are known, it is possible to determine the optimal speed of the liner ship serving a certain route. In this section, the method to determine this optimal speed will be explained.

Consider a route of the route network and the capacity of the liner ship used to serve this route. To maintain a weekly frequency, the number of ships needed will always equal the route duration, thus more ships are needed for lower sailing speeds. For each integer number of weeks between the minimum and maximum route duration (based on the maximum and minimum speed respectively, where the route duration may not exceed the maximum route duration imposed in the model), we calculate the cost of sailing the route in the given number of weeks. These costs include the capital, operating and fuel costs. After we calculated the costs for all feasible route durations, we select the route duration and corresponding speed that is associated to the lowest costs.

Thus, we determine the optimal sailing speed by a simple enumeration procedure. We will repeat this procedure for each route to obtain the optimal sailing speed for each individual route.

2.3.3 Cargo Routing Model

In the cargo-routing problem, we want to allocate the cargo flows over the route network given the origin-destination demand matrix and capacities on the routes. The cargo-routing problem can then be formulated as a multi-commodity flow problem, for which the linear programming formulation is given below. We will refer to this model as the cargo routing model (CRM). Introduce the following sets:

$h \in \mathcal{H}$	Set of ports.
$l \in \mathcal{L}$	Set containing pairs of consecutive ports h and h', where $l = (h, h')$.
$t \in \mathcal{T} \subseteq \mathcal{H}$	Set of transshipment ports.
$s \in \mathcal{S}$	Set of ship routes.
$j \in \mathcal{J}$	Indicator set denoting whether a ship passes both ports $h \in \mathcal{H}$ and $h' \in \mathcal{H}$ on ship route $s \in \mathcal{S}$, where $j = (h, h', s)$.
$k \in \mathcal{K}$	Indicator set denoting whether consecutive ports $l \in \mathcal{L}$ are visited on ship route $s \in \mathcal{S}$, where $k = (l, s)$.

The following parameters are used in the model:

$r_{hh'}$	Revenue of transporting one TEU from port $h \in \mathcal{H}$ to port $h' \in \mathcal{H}$.
c_t^t	Cost of transshipping one TEU in transshipment port $t \in \mathcal{T}$.
c_h^h	Cost of (un)loading one TEU in origin or destination port $h \in \mathcal{H}$.
$d_{hh'}$	Demand with origin port $h \in \mathcal{H}$ and destination port $h' \in \mathcal{H}$.
b_s	Capacity on ship route $s \in \mathcal{S}$.
$q_{hh'ls}$	0/1 parameter that takes the value 1 if a ship passes consecutive ports $l \in \mathcal{L}$ when sailing from port $h \in \mathcal{H}$ to port $h' \in \mathcal{H}$ on ship route $s \in \mathcal{S}$.

In the model formulation, we distinguish between direct flows and transshipment flows. Direct flows are cargo flows between the origin and destination port of a demand pair for which no transshipment movement has to be made. Cargo flows for which transshipment movements are necessary are called transshipment flows. Now, we introduce the following decision variables:

x_{ls}	Cargo flow on ship route $s \in \mathcal{S}$ between consecutive ports $l \in \mathcal{L}$.
$x^{od}_{hh's}$	Direct cargo flow on ship route $s \in \mathcal{S}$ between ports $h \in \mathcal{H}$ and $h' \in \mathcal{H}$.
$x^{ot}_{hth's}$	transshipment flow on ship route $s \in \mathcal{S}$ between port $h \in \mathcal{H}$ and transshipment port $t \in \mathcal{T}$ with destination port $h' \in \mathcal{H}$.
$x^{td}_{thss'}$	transshipment flow on ship route $s' \in \mathcal{S}$ between transshipment port $t \in \mathcal{T}$ and destination port $h \in \mathcal{H}$, where the flow to transshipment port $t \in \mathcal{T}$ was transported on ship route $s \in \mathcal{S}$.
$x^{tt}_{tt'hss'}$	transshipment flow on ship route $s' \in \mathcal{S}$ between transshipment port $t \in \mathcal{T}$ and transshipment port $t' \in \mathcal{T}$ with destination port $h \in \mathcal{H}$, where the flow to transshipment port $t \in \mathcal{T}$ was transported on ship route $s \in \mathcal{S}$.
$x^{tot}_{hh's}$	Total cargo flow on ship route $s \in \mathcal{S}$ between ports $h \in \mathcal{H}$ and $h' \in \mathcal{H}$.

Now, we can give the linear programming formulation.

$$\begin{aligned} \max & \sum_{h \in \mathcal{H}} \sum_{h' \in \mathcal{H}} r_{hh'} \sum_{s \in \mathcal{S}} \left(x^{od}_{hh's} + \sum_{t \in \mathcal{T}} x^{ot}_{hth's} \right) \\ & - \sum_{h \in \mathcal{H}} c^h_h \left(\sum_{t \in \mathcal{T}} \sum_{h' \in \mathcal{H}} \sum_{s \in \mathcal{S}} \left[x^{ot}_{hth's} + x^{ot}_{h'ths} \right] + \sum_{h' \in \mathcal{H}} \sum_{s \in \mathcal{S}} \left[x^{od}_{hh's} + x^{od}_{h'hs} \right] \right) \\ & - \sum_{t \in \mathcal{T}} c^t_t \left(\sum_{t' \in \mathcal{T}} \sum_{h' \in \mathcal{H}} \sum_{s \in \mathcal{S}} \sum_{s' \in \mathcal{S}} x^{tt}_{tt'h'ss'} + \sum_{h' \in \mathcal{H}} \sum_{s \in \mathcal{S}} \sum_{s' \in \mathcal{S}} x^{td}_{th'ss'} \right) \end{aligned} \tag{2.1}$$

subject to

$$\sum_{t\in\mathcal{T}}\sum_{s\in\mathcal{S}} x^{ot}_{hth's} + \sum_{s\in\mathcal{S}} x^{od}_{hh's} \leq d_{hh'} \qquad h \in \mathcal{H}, h' \in \mathcal{H} \tag{2.2}$$

$$x_{ls} \leq b_s \qquad (l,s) \in \mathcal{K} \tag{2.3}$$

$$\sum_{h\in\mathcal{H}} x^{ot}_{hth's} + \sum_{t'\in\mathcal{T}}\sum_{s'\in\mathcal{S}} x^{tt}_{t'th's's} - \sum_{s'\in\mathcal{S}} x^{td}_{th'ss'} - \sum_{t'\in\mathcal{T}}\sum_{s'\in\mathcal{S}} x^{tt}_{tt'h'ss'} = 0 \qquad (t,h',s) \in \mathcal{J} \tag{2.4}$$

$$x_{ls} - \sum_{h\in\mathcal{H}}\sum_{h'\in\mathcal{H}} x^{tot}_{hh's} q_{hh'ls} = 0 \qquad (l,s) \in \mathcal{K} \tag{2.5}$$

$$x^{tot}_{hh's} - x^{od}_{hh's} - \sum_{h''\in\mathcal{H}} x^{ot}_{hh'h''s} - \sum_{s'\in\mathcal{S}} x^{td}_{hh's's} - \sum_{h''\in\mathcal{H}}\sum_{s'\in\mathcal{S}} x^{tt}_{hh'h''s's} = 0 \qquad (h,h',s) \in \mathcal{J} \tag{2.6}$$

$$x_{ls} \geq 0 \qquad (l,s) \in \mathcal{K} \tag{2.7}$$

$$x^{od}_{hh's} \geq 0 \qquad (h,h',s) \in \mathcal{J} \tag{2.8}$$

$$x^{tt}_{tt'hss'} \geq 0 \qquad h \in \mathcal{H}, s \in \mathcal{S}, (t,t',s') \in \mathcal{J} \tag{2.9}$$

$$x^{td}_{thss'} \geq 0 \qquad s \in \mathcal{S}, (t,h,s') \in \mathcal{J} \tag{2.10}$$

$$x^{ot}_{hth's} \geq 0 \qquad h' \in \mathcal{H}, (h,t,s) \in \mathcal{J} \tag{2.11}$$

Objective (2.1) of the cargo routing problem is to maximize total profit. Profit is given by the revenue minus the costs. The costs only consist of (un)loading cost and transshipment cost, because the route network is given, so all other costs are fixed and can be subtracted afterwards.

Constraints (2.2) ensure that the total cargo shipped from one port to another does not exceed the demand of that port combination. Next, Constraints (2.3) make sure that the total load of a ship between each two consecutive ports does not exceed the capacity of the ship. Constraints (2.4) ensure that the flow to a transshipment port with destination port $h' \in \mathcal{H}$ has to equal the flow from that transshipment port to port h'. In other words, they make sure that all flow unloaded to be transshipped, will also be loaded on another route. Constraints (2.5) define the the amount of flow between two consecutive ports and Constraints (2.6) define the total flow between each two ports in the same cycle. Finally, Constraints (2.7)-(2.11) guarantee a nonnegative flow between each two ports

2.3.4 Disaggregation method

The cargo routing model can be executed with the clusters as determined in Section 2.3.1 as input. We will obtain cargo flows between the clusters as output from the cargo routing model. In practice, it is necessary to know the exact route of each load from the origin port to the destination port. Therefore, we first have to determine the real origin and destination ports of each cargo flow. Thereafter, we will use feeder services to ship the cargo from the origin to the destination port in the cluster, so a feeder network has to be constructed. The problem of constructing a feeder network is similar to the vehicle routing problem with pickups and deliveries (see for example Berbeglia et al. 2007, for an overview). Nagy and Salhi (2005) provide some routines that can be used to improve existing routes for this problem. When constructing the feeder service, in first instance, a feeder service is added from the central port in the cluster to each other port belonging to the cluster. Thereafter, the feeder routes will be improved, for which we will use a method to decrease the ship sizes on the feeder routes, a method comparable to the SWITCH routine discussed in Nagy and Salhi (2005) and a method in which feeder ports are added to the main route network. Figure 2.2 shows an schematic overview of the disaggregation method.

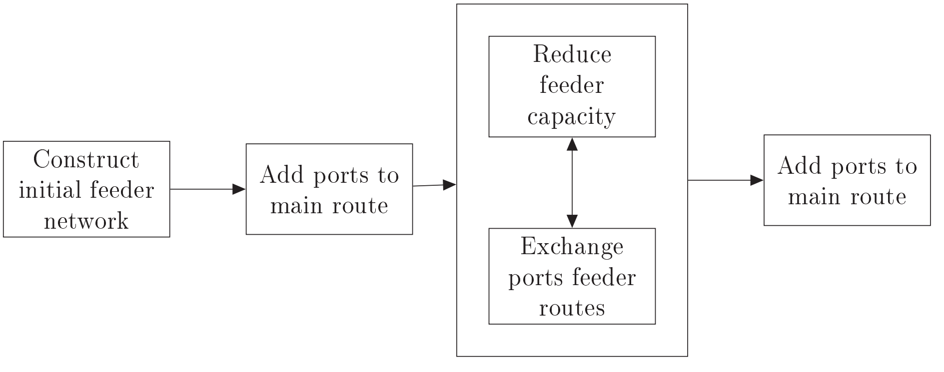


Figure 2.2: Schematic overview of the disaggregation method

Constructing initial feeder network

The cargo flow between each pair of clusters is obtained from the CRM model. For each cluster, we know which ports belong to this cluster. For each port pair with origin port in the origin cluster and destination port in the destination cluster, we can determine the revenue of transporting cargo between these ports. Furthermore, we know the demand between these two ports. Then, the disaggregation is done by repeatedly allocating as much cargo as possible to the port pair with the highest revenue until the total cargo flow is allocated. Because the cluster demand equals the sum of the individual port demand of the ports in the cluster and all possible combinations of port pairs with origin in the origin cluster and destination in the destination cluster are considered in the disaggregation method, the total transshipped cargo is always fully allocated using this method. Finally, this procedure is repeated for all combinations of two port clusters of the cargo routing model.

When all combinations are considered and the cargo routing between each two real ports is known, the size of the feeder services can be determined. In first instance, for each noncentral port in a cluster, a feeder service is made. This feeder ship will then sail from the central port of the cluster to the noncentral and back to the central port. The size of the ship can be obtained when considering the cargo transshipped from and to the noncentral port. These amounts will not be on the ship at the same time, so the maximum of the amount of cargo transshipped to and the amount transshipped from the noncentral port is the maximum load on the feeder ship. We will then use the ship that has minimal size but can still transport the maximum load. Note that only feeder services that sail with a frequency of once a week are considered. Exemptions are only made for ports that are placed in a cluster because their demand is too low, but cannot be served within one week with a direct feeder route from the central port of the cluster. For these ports, we can maintain a weekly frequency by assigning more than one ship to the route. However, this can be more costly, because more ships are needed. Therefore, in these cases we allow ports to be visited only once every two weeks if this is cheaper. Thus, we compare the costs of visiting the port every week (using two feeder ships for that route) with the costs of visiting the port every week (such that only one feeder ship is needed for the route).

Reducing the size of a feeder service

In this section, we describe the method used to reduce the size of the existing feeder services by reallocating the origin and/or destination port of a cluster cargo obtained from the cargo routing model. Each time, we select a port pair and consider the cargo flow from the origin to the destination port. When we start, all feeder services are direct services between the central port in a cluster and a noncentral port in the same cluster. In this case, only two cases have to be distinguished when reducing the size of a feeder service. The noncentral port can be the origin port of a cargo flow, in which case the cargo is on board when the ship sails from the noncentral port to the central port of the cluster or the noncentral port is the destination port of a cargo flow and the cargo is on board when the ship sails from the central to the noncentral port. In both cases, the cargo is only at one of the two legs on board, so only one leg has to be considered for each cargo flow. However, when ports are exchanged between feeder services, some feeder services are created that visit more than one noncentral port. In this case more legs have to be considered when a cargo flow is viewed. This makes the size reducing process more complicated.

Algorithm 1 describes the steps that have to be performed to determine the increase in profit when the size of a feeder service is reduced. The algorithm will be performed for each cluster separately. Note that no real changes are made in the algorithm. So, when the algorithm starts over in Step 1, the data is still the same as at the beginning. If a change is mentioned in the algorithm it is a temporarily change, which only holds during one iteration of the algorithm. In Appendix 2.E a more detailed explanation of the steps in the algorithm can be found.

Algorithm 1: Feeder service reduction algorithm

1. Consider a feeder service in the cluster. Determine the capacity of the feeder service when it is reduced by one size.
2. Determine the reduction needed on each leg of the feeder service.
3. Repeat the following as long as the sum of the reduction needed over all legs is larger than 0 and not all port combinations are considered.
    - Exchange as much cargo as possible between the port combinations that are not yet considered and have the lowest revenue decrease.
    - Update the reduction needed on each leg.
4. Check whether the sum of the reduction needed is 0.
    - If the sum is 0, determine the increase in profit when the exchanges are performed.
        - If the increase in profit is higher than the highest increase found earlier, save the new increase in profit and the reallocation of demand needed to decrease the feeder size.
    - Else, the reduction is not possible.
    - If all feeder services are considered, then stop. Else, return to Step 1.

Exchanging a port between two feeder services

In this step a port is exchanged between two feeder services. The cargo allocation does not change during this step, so the revenue and feeder handling costs will also not change. Because the costs on the main route will also stay the same, the only changes will occur in the feeder capital, operating and fuel costs. Finding a profitable exchange corresponds now to finding an exchange for which these feeder costs are reduced.

We will explain the method used to exchange ports using Algorithm 2. A more detailed description of the steps of the algorithm can again be found in Appendix 2.E. Again, the algorithm will be executed for each cluster separately and the changes made in the algorithm are only temporarily changes. In this case, each time the algorithm returns to Step 4 or Step 1, the data is the same as at the beginning. The real changes are only made in the third step of the method (the comparison).

Algorithm 2: Port exchange algorithm

1. Consider two feeder services, F_1 and F_2 in the cluster.
2. Determine all noncentral ports that are served by the feeder service F_1.
3. Determine all consecutive port combinations on feeder service F_2.
4. Repeat the following steps as long as not all combinations of a noncentral port and a consecutive port combination are considered.
    - Select a combination of a noncentral port N and a consecutive port combination (P_1, P_2).
    - Remove port combination (P_1, P_2) from feeder service F_2 and add the combinations (P_1, N) and (N, P_2) to F_2. Furthermore, remove port N from feeder service F_1.
    - Determine the new loads on and capacities of feeder services F_1 and F_2.
    - Determine the increase in profit obtained by adding port N between ports P_1 and P_2 on feeder service F_2.
        - If the increase in profit is higher than the highest increase found earlier, save F_1, F_2, N, P_1, P_2 and the new highest increase in profit.
5. If all combinations of two feeder services in the same cluster are considered, then stop. Else, return to Step 1.

Comparison

After the first two steps, both the most profitable reduction in the size of a feeder service and the most profitable exchange of a port between feeder services are known. Furthermore, the amount of increase in profit is known for both changes. Note, that the increase in profit can also be negative, which corresponds to a decrease in profit (loss), because it is not checked in the first two steps whether the increase in profit is positive or not. First check which increase in profit is the highest, that of the reduction in size or that of the port exchange. If the increase is positive, make the changes that corresponds to the increase. So, if the highest increase in profit is obtained by a reduction in the size of a feeder ship, reduce the saved feeder ship by one size and reallocate the demand necessarily to make this reduction possible. However, if the highest increase is obtained by a port exchange, remove the saved port from the first saved feeder service and add this port between the saved combination on the second saved feeder service. Finally, if a profitable change is made, go back to the first step, otherwise the feeder network cannot be improved further using this method, so stop.

Note that reducing the capacity on a feeder service or exchanging a port between feeder services influences the effect of future changes in the feeder network. For example, when the capacity on a feeder service is reduced, it might not be possible to add an additional port on that feeder service any more, because the increase in flow over the feeder service would exceed the new lower capacity. Thus, if we do not make the right change in first instance, it is possible that we block some very profitable changes in the feeder network. Therefore, we consider first all possible changes, before we make the most profitable one and start over again. However, changes on different clusters can be performed at the same time, because they are independent of each other. Thus, the algorithm is performed for each cluster separately.

Add ports to main route

Next, we investigate whether it is profitable to add some ports to the main route and thereby reducing the size of the feeder routes serving those ports. At the moment, only the central port of a cluster is visited on the main routes of the route network. All other ports are served by a feeder service. However, some noncentral ports exist, which also have a large amount of cargo handling movements. Now the flows are known, it can be calculated whether it is profitable to visit these noncentral ports on one of the main routes. A part of the cargo flows from and to these ports can then be transported over the main routes. This diminishes the flow on the feeder service networks, which can reduce the costs of the feeder network. Ports can be visited both on one or more main routes and on a feeder route.

The method is performed once before and once after the method to decrease the feeder network. When it is performed before reducing the feeder network, the exact feeder costs are not yet known. Therefore, in this case only the decrease in feeder (un)loading cost are seen as cost reduction, where also the capital, operating and fuel costs are considered when the feeder network is already decreased.

Algorithm 3 gives a brief description of the method used to investigate whether ports should be added to main routes or not. In this algorithm, changes are only made in Step 6. So, changes in other steps of the algorithm are only temporarily. When the algorithm returns to a previous step, the changes are undone. In Appendix 2.E the method is explained in more detail.

2.3.5 Designing a new route network

Even if only a few ports are included in the problem, a lot of different route networks can be constructed. Since it is impossible to generate and evaluate all possible networks and the performance of comparable networks will probably also be comparable, we use a genetic algorithm based method to change the route networks. The representation of a network in this study is a binary string consisting $R(2n-1)$ 0/1 elements, where R is the number of routes in a network and n the number of port clusters found after the aggregation phase. So, $2n-1$ are the possible stops of a ship on a route. When the element corresponding to port cluster $c \leq 2n-1$ and route $r \leq R$ has value 1, port cluster C is called on route r. Furthermore, the capacity of the ship that will serve the route is stored in the representation. A route network then corresponds with a set of routes.

The following operators are used to change the route network: elitism, crossover and mutation. In the elitism step the best route network(s) of the current iteration are selected and are placed unchanged in the network set of the next iteration. Elitism ensures that the performance of the best network in the next iteration cannot be decreased in comparison to the best network in the current iteration.

For the crossover and mutation operators, we will use roulette wheel selection to select networks from the current set. Two different crossover methods are used: uniform and route crossover. In both crossover method two existing networks are selected and changed into two new networks. In the uniform crossover methods, each 0/1 value of the strings in each route of the network is with equal probability exchanged between

Algorithm 3: Add port to main route algorithm

1. Consider a main route and determine the clusters that are visited on that main route.

2. Consider one of those clusters.

3. Determine which (noncentral) ports that belong to the cluster are not yet visited on the considered main route.

4. Determine the consecutive port combinations on the main route for which at least one of the ports belongs to the cluster.

5. Repeat the following steps as long as not all combinations of noncentral ports and consecutive port combinations are considered.

   (a) Select a combination of a noncentral port N and a port combination (P_1, P_2).

   (b) Remove port combination (P_1, P_2) from the main route and add the combinations (P_1, N) and (N, P_2) to the route.

   (c) Determine the new loads on the main route and on the feeder service serving port N after reallocating as much cargo from and to port N as possible to the main route.

   (d) Determine the increase in profit obtained by adding port N between ports P_1 and P_2 at the main route. If the increase in profit is higher than the highest increase in profit obtained earlier, save the new highest increase, the considered cluster and main route and ports N, P_1 and P_2.

6. In this step a port is finally added to a route.

   - Consider first all clusters and all routes and add thereafter the most profitable port to the main route.
     - Return to Step 2 as long as not all clusters are considered.
     - Return to Step 1 as long as not all routes are considered.
     - Add the most profitable port at the most profitable place to the main route if the increase in profit is larger than 0.
     - If a profitable change is made in Step 6, return to Step 1, else stop.

the two selected networks or not exchanged. Similarly, the capacities of the routes are with equal probability exchanged or not. Finally, the routes are made feasible using the method earlier described and it is checked whether the ship types allocated to the routes in the networks still satisfy the available fleet constraint. In the route crossover method, existing routes are not changed, but only exchanged between the selected route networks. We randomly select a cut point, after which the routes including capacities of the two selected networks are exchanged. All routes that occur in the new route networks that are created using the route crossover method are always feasible, because they are unchanged according to the routes in the current iteration. However, the fleet composition of the networks can violate the available fleet constraint. Thus, we check whether this happens and exchange routes with allocated ships where necessarily.

Finally, the mutation method changes the value of some elements. When a route network is selected, some elements corresponding to a route and a port cluster are selected at random. The selected ports are added to the route when they are not visited on the current route. On the other hand, when the ports are visited on the current route, they are deleted from the route. Furthermore, the feasibility of the route has to be checked, because the mutation operator can make routes infeasible. The capacity of a certain route will also be changed with a certain probability. The new capacity will then be randomly chosen from the existing, still available, capacities. When the capacity of the routes can also be changed, more feasible route networks will be considered.

2.4 Case study

We apply the proposed algorithm to design a service network that consists of 58 ports on the Asia-Europe trade lane. Most data is based on the service network of Maersk on the Asia-Europe trade lane during spring 2010. It is difficult to obtain real data, so demand is estimated using port throughput data and data obtained from the annual reports of Maersk. Furthermore, costs for different types of ships and sailing speeds are estimated using data obtained from Lachner and Boskamp (2011), Francesetti and Foschi (2002) and Notteboom (2006). The data can be found in Appendix 2.A.

2.4.1 Reference network

We define a reference network to which the best obtained networks (i.e. the networks with highest profits) can be compared. In the reference network, no feeder services are included. Comparing our best networks to a reference network will give some information on the performance of the obtained networks. The routes of the reference network can be found in Appendix 2.B.

The original Maersk route network, on which the data is based, is used as reference network. The network consist of nine routes. On each route a few ships with different capacities are sailing to serve the demand. In our study, a route is served by ships of the same size and each ship size is a multiple of 1000 TEU. Therefore, we round the average capacities of the ships used on a route of the Maersk service network to the nearest multiple of 1000 TEU that is larger than the average to obtain the capacities of the reference network. The profit of the reference network can be determined after using the speed-optimization method and the cargo-routing model and is 1.707 billion USD. However, the capacities of the ships sailing on the reference network are fixed as explained above, while the liner shipping company is allowed to choose ships from an available set in our method. Thus, a comparison between our method and the reference network is not completely fair. Therefore, we used a simple improving heuristic to improve the allocated ships on the reference networks, such that the fleet satisfies the same restrictions as imposed in our method. In this heuristic, we first determine the optimal ship capacity on each route of the reference network, without taking the fleet restrictions imposed in this research into account. Thereafter, we make the allocation of ships to routes feasible by reducing the size of the ships on the routes for which the profit decrease is smallest. Finally, we repeatedly try to reduce the ship capacity on one of the routes until this is not possible anymore without causing a decrease in profit. The improved network has a profit of 1.772 billion USD, which is already an increase of 3.8% compared to the reference network with fixed capacities.

2.4.2 Characteristics of the best networks

Table 2.1 first shows the overall characteristics of the reference network and the best networks obtained using the described method with different numbers of clusters. We performed our methods with different number of clusters obtained from the aggregation step in order to investigate the importance of choosing the right number of clusters. Less clusters will result in lower computational time of the cargo-routing model, but a larger feeder network. Thus, by comparing different numbers of clusters, a trade-off can be made between performance and computational time. The best network is found when including only ten clusters and results in a profit of 1.953 billion USD, which is an increase of 14.4% compared to the reference network and an increase of 10.2% compared to the improved reference network. The increase in profit is the result of a cost reduction. The total demand delivered and the revenue in the best networks are lower than in the reference network, but the routes are more cost efficient. Note that we only consider ship-related costs and revenues in this model. In reality, liner shipping companies will also face other types of costs. However, it is difficult to estimate these non-ship-related costs, so we leave them out of consideration here. Furthermore, we only use estimates on demand, revenue and costs in this research. Since these data are highly sensitive to fluctuations, the increase in profit can change over time.

	Reference network	8 clusters	10 clusters	12 clusters	15 clusters
Profit in billion USD	1.707	1.910	1.953	1.927	1.837
Revenue in billion USD	7.015	5.797	6.516	6.750	6.620
Cost in billion USD	5.307	3.895	4.563	4.823	4.783
Fraction demand delivered	0.832	0.669	0.739	0.793	0.777
Computational time in seconds	215.7	88.6	103.9	135.3	210.7
Main network					
Capital and operating cost in billion USD	1.150	0.840	0.952	1.077	1.039
Fuel cost in billion USD	1.453	1.064	1.174	1.338	1.316
Port cost in billion USD	0.189	0.140	0.160	0.211	0.194
Handling cost in billion USD	1.972	1.585	1.753	1.880	1.843
Transhipment cost in billion USD	0.544	0	0.100	0	0.035
Total liner cost in billion USD	5.307	3.630	4.139	4.506	4.426
Fleet size	91	64	71	80	79
Number of routes	9	6	7	7	7
Average number of ports per route	16.4	18.0	17.7	19.3	21.3
Distance travelled in nmi	191,754	136,041	145,719	164,680	163,266
Feeder network					
Capital and operating cost in million USD		40.200	59.925	48.300	51.750
Fuel cost in million USD		35.153	61.520	43.399	41.648
Port cost in million USD		21.840	27.300	26.520	25.740
Handling cost in million USD		167.849	275.950	199.383	237.297
transshipment cost in million USD		0	0	0	0
Total feeder cost in million USD		265.042	424.694	317.602	356.434
Fleet size		12	16	14	15
Number of routes		12	16	14	15
Average number of ports per route		3.3	3.2	3.4	3.2
Distance travelled in nmi		21,469	27,555	25,729	22,259
Fraction of cargo on feeder network		0.148	0.220	0.149	0.180

Table 2.1: Characteristics of the best networks

The lower part of Table 2.1 shows some characteristics of the main and feeder routes of the best networks and the main routes of reference network. First note that compared to the reference network, the number of ships decreases in the best network with ten clusters, because the use of a feeder network leads to a more efficient use of ships. In the best network, only seven routes are included compared with nine in the reference network. Since, the number of ships needed for a route equals the round tour time, this will already lead to a significant reduction in the fleet size. The feeder services are all designed in such a way that they will have a round tour time of one week, so only one ship per feeder service is needed. Furthermore, the ships on the feeder network are in general smaller than the ships on the main network. Therefore, the capital, operating and fuel costs on the feeder network are much lower than on the main network. Thus, by using smaller ships for the feeder network, a cost reduction in the ship costs will be obtained. The best networks with twelve and fifteen clusters have a larger fleet than the reference network, but because the ship costs on the feeder network are lower, the total ship costs of these networks are still lower than for the reference network.

Note that the total feeder costs of the best network with twelve clusters are lower than those corresponding to the best network obtained when using fifteen clusters. One would expect the total feeder costs to decrease when the number of included clusters increases as less ports have to be served on the feeder network. However, from Table 2.1 we can conclude that in both cases 48 port visits are made on the feeder network ($15 \cdot 3.2 = 48$ and $14 \cdot 3.4 \approx 48$). For instances with more clusters, the initial main routes will probably contain more ports and thus more port combinations. Then, the average load on a route is also probably higher, which makes it more difficult to add noncentral ports to the main routes. Furthermore, less cargo can be allocated to the main route when a noncentral port is added to that route. Therefore, the total amount of cargo transshipped over the main routes is larger, which increases the feeder handling costs. We can indeed observe from the table that more handling costs are incurred in the instance with fifteen clusters than in that with twelve clusters.

Similarly, the total feeder costs of the best network with eight clusters is lower than the feeder costs in all other best networks. This time, however, the fraction of demand delivered is also much lower than in the other networks. With less clusters, the initial routes are smaller and some ports are not visited at all on the best networks. Therefore, less cargo has to be transported over the feeder network and consequently less feeder costs are incurred.

It follows from Table 2.1 that a large cost reduction is obtained in the transshipment cost compared to the reference network. By clustering ports, less ports are included in the main route network and thus less transshipment movements are needed. However, additional handling costs on the feeder network are incurred, but in total the handling costs decrease, also since less demand is satisfied on the best networks than on the reference network.

In the table, also the fraction of cargo that is transported over the feeder network is given. Since each container can be transported twice over the feeder network (once from the origin port to the central port of the origin cluster and once from the central port of the destination cluster to the destination port), this fraction can vary between 0 and 2. The fraction of cargo transported over the feeder network gives some indication on the average transit times of containers in the network. Cargo that is only transported over the main network will in general have lower transit times than cargo that has to be transshipped to the feeder network. Furthermore, the total distance travelled on the main network combined with the fleet size also gives an indication of the average transit times in the network. If routes are shorter, less ships are needed and the travel times between two ports on the same route will on average be lower than when larger routes are used. Finally, the transshipment costs also give an indication of the average transit time. If more transshipment moves are needed, the transit times will generally be higher. Combining these three indicators, we can conclude from the table that the best networks obtained with eight and twelve clusters probably have lower average transit times than the best networks obtained with ten and fifteen clusters. In the networks with eight and twelve clusters, no transshipment moves take place on the main network and the fraction of cargo that is transported over the feeder network is also significantly lower than for the other networks. Furthermore, the network with eight clusters also has the smallest fleet size and distance travelled, so this network will probably perform best with respect to average transit time.

The computational times are given in seconds for the reference network and in seconds per iteration for the best network. In one iteration, the profit of all twenty route networks in the set is determined and a new set is made. The composite algorithm is stopped after eight hours of computational time in which an average improvement of 12% is found compared to the best initial network. The best network with ten clusters has both the highest profit and one of the lowest average computational times per iteration. Therefore, performing the methods with ten initial port clusters is preferred in this study. Furthermore, it can be seen that including too many clusters can have a negative effect on the performance. From Table 2.1, it follows that the computational time per iteration of the method increases when more port clusters are considered. Thus, less iterations can be performed in the maximum computational time and the profit of the best network obtained with fifteen clusters is 5.9% lower than the highest profit found with only ten clusters.

2.4.3 Best network

The routes of both the main and the feeder network of the best obtained route network can be found in Appendix 2.D. The reference network only consists of a main route network, so all ports are at least visited once on a main route. However, not all ports are visited on a main route in the best network. First, the ports in Japan are not visited any more on the main route. Many ports in China are visited on a main route, but ships have to cross the ocean to visit ports in Japan. Because China is the turning point of the route, the crossing distance should be covered twice when ports in Japan are included in the main routes. Therefore, it is quite logical that the ports in Japan are only visited on the feeder services.

Many ports in Southern Europe are located in a cove, so that a lot of additional distance has to be covered to visit these ports on the main routes. Again, it is then logical that these ports are not visited on the main routes, but are fully served by the feeder network, since smaller and cheaper ships are used on the feeder routes.

Finally, some ports, for example Taipei, are not visited on the main route because they are relatively small. The additional distance that has to be covered to visit these ports is not very large (Taipei is located near Kaohsiung, which is visited on some main routes). However, since they are small ports, the additional costs incurred by adding them to a main route, will probably not be covered by the decrease in feeder costs.

Ports are mainly visited in geographical order on the routes in the best network found. However, sometimes small deviations from the geographical order are observed. These deviations are caused by the cargo flows. For example, on one of the routes first Ningbo is visited and thereafter respectively Qingdao, Busan, Xingang, Dalian and Shanghai. All these ports belong to the cluster with central port Shanghai. The geographical order would be Busan-Dalian-Xingang-Qingdao-Shanghai-Ningbo. Thus, for example, the location of Ningbo in the obtained route deviates from the geographical order. This can be explained in the following way. The method that adds ports to the main routes determines the best location to place a port on the existing route. The best location is defined as the location where the highest increase in profit can be obtained. The additional distance that has to be sailed is part of the decision, because additional costs are incurred when more distance has to be covered. However, the optimal place to add a port to the main route depends also on the reduction of the costs that can be obtained. A cost reduction can be obtained by reallocating cargo to the main routes, such that transshipment costs to the feeder services are saved. We have seen that the amount of cargo that can be loaded in the added port on the main route is bounded when a port is placed before the central port of the cluster, while the amount of cargo unloaded from the main route is bounded when the port is visited after the central port of the cluster. Thus, although the optimal geographical location of Ningbo would be after Shanghai, it can be more profitable to visit Ningbo before Shanghai when much cargo with destination Ningbo is on the ship.

2.4.4 Uncertain demand

The best networks are found using a given origin-destination demand matrix, while the reference network is probably constructed with other (uncertain) demand data. Thus, the comparison between the reference network and the best networks found in this case study, will not be completely fair. Therefore, we created ten instances with randomly generated origin-destination demand data, where each origin-destination pair has demand between 80% and 120% of the given demand matrix. For each of these ten instances, we solved the cargo-routing model for the reference network and for the best network (with ten clusters). The average demand of the ten test cases is 0.2% higher than in the original scenario. The reference network has on average an increase in profit of 0.9% with respect to the reference network with given demand over these ten instances, while the profit increase of the best network is on average 0.8%. Also, the standard deviation of the profit does not differ that much: 1.0% for the reference network against 1.4% for the best network. These figures indicate that the best network is a bit more volatile for changes in the demand, but they do not suggest that the method is very sensitive to the used demand matrix.

2.5 Conclusion

In this study methods to solve the combined fleet-design, ship-scheduling and cargo-routing problem are developed. Thereto, an aggregation method is proposed, which aggregates ports into port clusters to reduce the size of the problem. Thereafter, initial route networks are constructed and a linear programming formulation (cargo routing model) is introduced that can be used to solve the cargo-routing problem to optimality. The number of clusters is chosen based on the computational time of the model. In our study, eight to fifteen clusters are appropriate to work with. The design of the clusters is based on the geographical location of the ports.

After the results are obtained in clustered ports, they have to be disaggregated again in individual port results. Some methods are developed and explained in this study. In these methods, a distinction is made between main services and feeder services. The feeder services are used to transport the cargo from the cluster centers to the other ports in the cluster. In first instance, only cluster centers can be part of the main service network. However, other ports are added to the main routes when this is profitable. Furthermore, we try to decrease the ship capacities on the feeder services in the proposed methods.

The above methods can be used to determine the profit of a certain route network. Then, methods are given to change existing networks into new networks, which can again be solved using above methods. We considered different numbers of port clusters obtained from the aggregation method, to investigate the importance of selecting the right number of clusters. When less clusters are considered, the average computational time per iteration reduces and the performance increases. The best results are obtained when ten port clusters are included, while including fifteen clusters results in more than 5% less profit in our case study.

The best network found using the overall model in our case study gives an improvement of 14.4% compared to the reference network. However, in the reference network, the used fleet is fixed, while we construct a new fleet under some restrictions in our methods. Therefore, we also compare the best network to the reference network with improved ship capacities on the routes. The profit of the best network is 10.2% higher than the profit of this reference network with improved fleet. The increase in profit is the result of a cost reduction obtained when integrating liner shipping network design. However, since we only consider ship-related costs and revenues and the data is highly sensitive to fluctuations, the profits of the liner shipping company for both the reference network and the best network will be lower in reality. The percentage of improvement will then also be lower.

The reference network only consists of a main service network. Therefore, all ports are visited on at least one of the main routes. However, in the best obtained network in our case study not all ports are visited on a main route any more. There are basically three possible reasons for this. When the last ports before a turning port are noncentral ports in a network, the distance from the central port to these ports has to be covered twice when these ports are added to a main route. Thus, these ports can probably better be visited on a (cheaper) feeder route. Furthermore, some ports are located in a cove. The additional distance that has to be travelled to visit these ports can therefore become large. In this case, it is probably more profitable to serve these ports on feeder routes instead of a main route. Finally, some ports have very little demand. For these ports, the additional costs of visiting these ports on a main route are higher than the maximum reduction in costs that can be obtained. Therefore, these ports can also better be visited on a feeder route.

In our case study, the order in which the ports are visited on the main routes in the best obtained network corresponds most of times to the geographical order. Some deviations can be found, because ports are afterwards added to the main routes. The amount of cargo that can maximally be (un)loaded in a port that is added to a main route depends on whether the port is added before or after the central port of the cluster. Therefore, it can be more profitable to add a port after the central port of the cluster, even when the geographical order implies that the port should be added before the central port and vice versa.

Finally, the reference network is probably constructed with other (uncertain) demand, so we constructed ten scenarios with uncertain demand and compared the difference in performance of the reference network and the best obtained profit. In our study, the reference network and best network react comparable to changes in demand for the scenario, so this does not seem to be a large issue.

Only intra-regional demand is considered in this study, but it is possible to add also regional demand in the model. The idea behind the methods will stay the same when regional demand is included. The regional demand will not be considered in the methods discussed in the improvement steps. Because the revenue of the regional demand will be relatively low compared to intra-regional demand (because the distance between origin and destination is much smaller), this will hardly influence the performance of the methods.

Appendix

2.A Data

Ports

The ports considered in this study are obtained by merging all routes in the Asia-Europe trade lane of Maersk during spring 2010. Port Los Angeles is removed from the list, because it is not on the Asia-Europe trade lane. The 58 remaining ports, countries and regions can be found in natural order in Table 2.2.

Distance

The distances between ports can be computed using distance calculators on the internet. The distances between the port combinations can be found in Table 2.3.

Demand

In the cargo routing model it is important to know the demand between two ports. However, it is hard to achieve realistic data on the demand. The demand data is obtained from Lachner and Boskamp (2011). First, they determine total demand to be allocated on the Asia-Europe trade lane. This is done using annual reports of Maersk. Furthermore, a growth percentage is included in the calculation and corrections are made for joint services. Thereafter, the total demand is divided over port combinations using port throughput. The port throughput of both the origin and the destination port is used to determine the demand of a port combination. The demand that is generated in this way can be found in Table 2.5.

Revenue

The revenue data is also obtained from Lachner and Boskamp (2011). It is assumed that the revenue per unit only depends on the distance between the origin and destination port of the demand and on the direction in which the demand has to be transported. Thereto, two revenue factors are introduced. The first factor gives the revenue of transporting one unit of cargo over one nautical mile in the westbound direction. The other revenue factor gives the revenue of transporting one unit of cargo over one nautical mile in the eastbound direction. Then, for each port combination, it is checked whether cargo has to be transported in westbound or eastbound direction. Finally, the corresponding revenue factor is multiplied with the direct distance between origin and destination port, which gives the revenue per unit of the considered port combination.

Lachner and Boskamp (2011) obtained the revenue by taking the 10-year average of historical data. This calculation gives the revenue in USD/TEU for both the eastbound and the westbound direction. Thereafter, they divided these revenues by the average distance between Asian and European ports. This results in the two revenue factors. The revenue factor is 0.0838 USD/nmi in eastbound direction and 0.1677 USD/nmi in westbound direction.

Port name	Country	Region	Port name	Country	Region
Yokohama	Japan	Asia	Port Said	Egypt	Middle East
Shimizu	Japan	Asia	Damietta	Egypt	Middle East
Nagoya	Japan	Asia	Izmit	Turkey	Europe
Kobe	Japan	Asia	Istanbul Ambarli	Turkey	Europe
Busan	South Korea	Asia	Odessa	Ukraine	Europe
Kwangyang	South Korea	Asia	Ilyichevsk	Ukraine	Europe
Dalian	China	Asia	Constantza	Romania	Europe
Xingang	China	Asia	Piraeus	Greece	Europe
Qingdao	China	Asia	Rijeka	Croatia	Europe
Liangyungang	China	Asia	Koper	Slovenia	Europe
Shanghai	China	Asia	Trieste	Italy	Europe
Ningbo	China	Asia	Gioia Tauro	Italy	Europe
Fuzhou	China	Asia	Genoa	Italy	Europe
Taipei	Taiwan	Asia	Fos	France	Europe
Xiamen	China	Asia	Barcelona	Spain	Europe
Kaohsiung	Taiwan	Asia	Valencia	Spain	Europe
Shenzhen Yantian	China	Asia	Malaga	Spain	Europe
Hong Kong	China	Asia	Algeciras	Spain	Europe
Shenzhen Chiwan	China	Asia	Tangiers	Marocco	Europe
Shenzhen Da Chan Bay	China	Asia	Le Havre	France	Europe
Vung Tau	Vietnam	Asia	Felixstowe	United Kingdom	Europe
Laem Chabang	Thailand	Asia	Zeebrugge	Belgium	Europe
Singapore	Singapore	Asia	Antwerp	Belgium	Europe
Tanjung Pelepas	Malaysia	Asia	Rotterdam	Netherlands	Europe
Port Klang	Malaysia	Asia	Bremerhaven	Germany	Europe
Colombo	Sri Lanka	Asia	Hamburg	Germany	Europe
Jebel Ali	Dubai	Middle East	Gothenburg	Sweden	Europe
Salalah	Oman	Middle East	Aarhus	Denmark	Europe
Jeddah	Saudi Arabia	Middle East	Gdansk	Poland	Europe

Table 2.2: List of ports

Origin \ Destination	Yokohama	Shimizu	Nagoya	Kobe	Busan	Kwangyang	Dalian	Xingang	Qingdao	Liangyungang	Shanghai	Ningbo	Fuzhou	Taipei	Xiamen	Kaohsiung	Shenzhen Yantian	Hong Kong	Shenzhen Chiwan	Shenzhen Da Chan Bay	Vung Tau	Laem Chabang	Singapore	Tanjung Pelepas	Port Klang	Colombo	Jebel Ali	Salalah	Jeddah
Yokohama	0	96	188	333	626	679	1095	1246	1060	1086	1008	1020	1211	1134	1316	1319	1565	1582	1592	1597	2418	2972	2879	2899	3072	4431	6303	6045	7190
Shimizu	96	0	119	268	560	614	1030	1181	994	1020	942	954	1146	1064	1250	1254	1500	1516	1526	1531	2355	2907	2818	2838	3011	4370	6242	5984	7129
Nagoya	188	119	0	219	511	565	980	1132	945	971	893	905	1097	1028	1201	1205	1451	1467	1477	1482	2322	2858	2771	2791	2964	4323	6195	5937	7082
Kobe	333	268	219	0	341	394	810	961	775	801	729	761	999	917	1103	1107	1353	1369	1379	1384	2213	2760	2679	2699	2872	4231	6103	5845	6990
Busan	626	560	511	341	0	72	503	654	467	493	445	495	752	688	869	898	1120	1137	1147	1152	1977	2551	2483	2503	2676	4035	5907	5649	6794
Kwangyang	679	614	565	394	72	0	437	588	401	427	382	436	699	616	817	840	1062	1084	1094	1099	1905	2493	2428	2448	2621	3980	5852	5594	6739
Dalian	1095	1030	980	810	503	437	0	188	251	324	477	556	843	851	961	1006	1212	1229	1249	1244	2111	2659	2576	2596	2769	4128	6000	5742	6887
Xingang	1246	1181	1132	961	654	588	188	0	394	468	621	699	987	996	1105	1150	1356	1372	1382	1387	2242	2773	2720	2740	2913	4272	6144	5886	7031
Qingdao	1060	994	945	775	467	401	251	394	0	89	307	386	675	697	793	838	1044	1061	1071	1076	1959	2526	2408	2428	2601	3960	5832	5574	6719
Liangyungang	1086	1020	971	801	493	427	324	468	89	0	258	337	626	641	745	790	996	1012	1022	1027	1876	2407	2360	2380	2553	3912	5784	5526	6671
Shanghai	1008	942	893	729	445	382	477	621	307	258	0	101	390	421	509	554	760	776	786	791	1675	2242	2124	2144	2317	3676	5548	5290	6435
Ningbo	1020	954	905	761	495	436	556	699	386	337	101	0	318	355	436	481	687	704	714	719	1572	2103	2051	2071	2244	3603	5475	5217	6362
Fuzhou	1211	1146	1097	999	752	699	843	987	675	626	390	318	0	150	168	235	420	436	456	451	1318	1888	1784	1804	1977	3336	5208	4950	6095
Taipei	1134	1064	1028	917	688	616	851	996	697	641	421	355	150	0	222	224	446	471	481	486	1312	1877	1812	1832	2005	3364	5236	4970	6124
Xiamen	1316	1250	1201	1103	869	817	961	1105	793	745	509	436	168	222	0	166	270	287	297	302	1150	1714	1638	1658	1831	3190	5062	4804	5949
Kaohsiung	1319	1254	1205	1107	898	840	1006	1150	838	790	554	481	235	224	166	0	337	354	364	369	1121	1653	1611	1631	1804	3163	5035	4777	5922
Shenzhen Yantian	1565	1500	1451	1353	1120	1062	1212	1356	1044	996	760	687	420	446	270	337	0	46	56	61	947	1482	1436	1456	1629	2988	4860	4602	5747
Hong Kong	1582	1516	1467	1369	1137	1084	1229	1372	1061	1012	776	704	436	471	287	354	46	0	10	15	927	1489	1410	1430	1603	2962	4834	4576	5721
Shenzhen Chiwan	1592	1526	1477	1379	1147	1094	1249	1382	1071	1022	786	714	456	481	297	364	56	10	0	5	937	1499	1420	1440	1613	2983	4844	4586	5731
Shenzhen Da Chan Bay	1597	1531	1482	1384	1152	1099	1244	1387	1076	1027	791	719	451	486	302	369	61	15	5	0	942	1504	1425	1445	1618	2988	4849	4591	5736
Vung Tau	2418	2355	2322	2213	1977	1905	2111	2242	1959	1876	1675	1572	1318	1312	1150	1121	947	927	937	942	0	681	646	666	839	2198	4070	3796	4958
Laem Chabang	2972	2907	2858	2760	2551	2493	2659	2773	2526	2407	2242	2103	1888	1877	1714	1653	1482	1489	1499	1504	681	0	831	836	1009	2368	4240	3982	5127
Singapore	2879	2818	2771	2679	2483	2428	2576	2720	2408	2360	2124	2051	1784	1812	1638	1611	1436	1410	1420	1425	646	831	0	20	193	1552	3424	3166	4311
Tanjung Pelepas	2899	2838	2791	2699	2503	2448	2596	2740	2428	2380	2144	2071	1804	1832	1658	1631	1456	1430	1440	1445	666	836	20	0	173	1532	3404	3146	4291
Port Klang	3072	3011	2964	2872	2676	2621	2769	2913	2601	2553	2317	2244	1977	2005	1831	1804	1629	1603	1613	1618	839	1009	193	173	0	1370	3243	2985	4130
Colombo	4431	4370	4323	4231	4035	3980	4128	4272	3960	3912	3676	3603	3336	3364	3190	3163	2988	2962	2983	2988	2198	2368	1552	1532	1370	0	1887	1633	2788
Jebel Ali	6303	6242	6195	6103	5907	5852	6000	6144	5832	5784	5548	5475	5208	5236	5062	5035	4860	4834	4844	4849	4070	4240	3424	3404	3243	1887	0	894	2152
Salalah	6045	5984	5937	5845	5649	5594	5742	5886	5574	5526	5290	5217	4950	4970	4804	4777	4602	4576	4586	4591	3796	3982	3166	3146	2985	1633	894	0	1270
Jeddah	7190	7129	7082	6990	6794	6739	6887	7031	6719	6671	6435	6362	6095	6123	5949	5922	5747	5721	5731	5736	4957	5127	4311	4291	4130	2788	2152	1270	0
Port Said	7894	7833	7786	7694	7498	7443	7591	7735	7423	7375	7139	7066	6799	6822	6653	6626	6451	6425	6435	6440	5656	5837	5015	4995	4834	3492	2857	1975	715
Damietta	7925	7864	7817	7725	7519	7426	7622	7766	7454	7406	7170	7097	6830	6842	6674	6657	6482	6456	6466	6471	5668	5806	5036	5025	4854	3508	2888	2006	746
Izmit	8718	8650	8617	8518	8329	8278	8422	8566	8254	8206	7970	7898	7630	7644	7484	7450	7275	7256	7266	7271	6478	6654	5832	5825	5639	4303	3687	2805	1539
Istanbul Ambarli	8669	8608	8561	8469	8273	8219	8366	8510	8198	8150	7914	7841	7574	7591	7428	7401	7226	7200	7210	7215	6425	6599	5790	5770	5609	4267	3632	2750	1490
Odessa	9016	8955	8908	8816	8620	8539	8713	8857	8545	8497	8261	8188	7921	7938	7775	7748	7573	7547	7557	7562	6772	6956	6137	6117	5943	4607	3979	3097	1837
Ilyichevsk	9013	8941	8905	8809	8617	8526	8731	8870	8563	8494	8279	8185	7939	7935	7772	7741	7592	7565	7570	7575	6769	6943	6123	6114	5930	4594	3996	3094	1834
Constantza	8872	8811	8764	8672	8476	8422	8569	8713	8401	8353	8117	8044	7777	7794	7631	7604	7429	7403	7413	7418	6628	6802	5993	5973	5812	4470	3835	2953	1693
Piraeus	8481	8420	8373	8281	8085	8033	8178	8322	8010	7962	7726	7653	7386	7412	7240	7213	7038	7012	7022	7027	6246	6413	5602	5582	5421	4079	3444	2562	1302
Rijeka	9139	9078	9031	8939	8743	8691	8836	8980	8668	8620	8384	8311	8044	8070	7898	7871	7696	7670	7680	7685	6904	7071	6260	6240	6079	4737	4102	3220	1960
Koper	9178	9117	9070	8978	8782	8727	8875	9019	8707	8659	8423	8350	8083	8109	7937	7910	7735	7709	7719	7724	6943	7124	6299	6279	6117	4776	4141	3259	1999
Trieste	9186	9120	9078	8986	8790	8735	8883	9027	8715	8667	8431	8358	8091	8114	7945	7918	7738	7717	7727	7732	6948	7116	6302	6282	6109	4773	4149	3267	2007
Gioia Tauro	8822	8761	8714	8622	8426	8371	8519	8663	8351	8303	8067	7994	7727	7753	7581	7554	7379	7353	7363	7368	6587	6768	5943	5923	5762	4420	3785	2903	1643
Genoa	9305	9244	9197	9105	8909	8854	9002	9146	8834	8786	8550	8477	8210	8237	8064	8037	7862	7836	7847	7852	7071	7252	6426	6406	6245	4903	4268	3386	2126
Fos	9405	9344	9297	9205	9009	8954	9102	9246	8934	8886	8650	8577	8310	8332	8164	8137	7962	7936	7946	7951	7166	7340	6526	6506	6345	5003	4368	3486	2226
Barcelona	9476	9415	9368	9276	9080	9029	9173	9317	9005	8957	8721	8648	8381	8407	8235	8208	8033	8007	8034	8039	7241	7409	6597	6577	6416	5074	4439	3557	2297
Valencia	9556	9495	9448	9356	9160	9109	9253	9397	9085	9037	8801	8728	8461	8487	8315	8288	8113	8087	8097	8102	7321	7489	6677	6657	6496	5154	4519	3637	2377
Malaga	9758	9690	9650	9558	9362	9275	9455	9599	9324	9239	9040	8930	8663	8684	8517	8490	8315	8326	8336	8341	7518	7683	6872	6859	6679	5343	4758	3839	2579
Algeciras	9804	9743	9696	9604	9408	9340	9501	9645	9333	9285	9049	8976	8709	8735	8563	8536	8361	8335	8385	8390	7569	7736	6925	6905	6744	5402	4767	3885	2625
Tangiers	9835	9763	9728	9624	9440	9348	9533	9677	9365	9305	9081	9008	8741	8757	8595	8556	8393	8367	8410	8415	7591	7757	6945	6937	6752	5416	4799	3915	2657
Le Havre	10933	10872	10825	10736	10540	10485	10633	10774	10462	10414	10178	10105	9841	9864	9692	9668	9490	9467	9477	9482	8698	8879	8054	8034	7873	6534	5899	5014	3757
Felixstowe	11056	10995	10948	10856	10663	10605	10756	10897	10585	10537	10301	10228	9961	10013	9815	9788	9613	9587	9640	9645	8847	9014	8177	8157	7996	6657	6019	5137	3877
Zeebrugge	11083	11022	10975	10883	10687	10624	10780	10924	10612	10564	10328	10255	9988	10017	9842	9815	9640	9614	9624	9629	8851	9018	8204	8184	8023	6681	6046	5164	3904
Antwerp	11170	11109	11062	10970	10774	10711	10867	10986	10699	10626	10415	10317	10075	10078	9929	9902	9701	9701	9711	9716	8912	9077	8291	8246	8094	6758	6133	5234	3991
Rotterdam	11133	11072	11025	10933	10737	10688	10830	10974	10662	10614	10378	10305	10038	10081	9892	9865	9690	9664	9674	9679	8915	9082	8254	8234	8084	6742	6107	5214	3965
Bremerhaven	11328	11267	11220	11128	10932	10877	11025	11169	10857	10809	10573	10500	10233	10286	10087	10060	9885	9859	9889	9894	9120	9287	8449	8429	8268	6926	6291	5409	4149
Hamburg	11416	11355	11322	11216	11034	10971	11142	11242	10974	10882	10690	10573	10350	10334	10175	10148	9956	9976	9979	9984	9168	9332	8537	8502	8342	7006	6408	5490	4237
Gothenburg	11568	11507	11460	11368	11175	11117	11268	11409	11097	11049	10813	10740	10476	10524	10327	10300	10125	10099	10141	10146	9358	9525	8689	8669	8508	7169	6531	5649	4389
Aarhus	11662	11601	11554	11462	11269	11127	11419	11391	11248	11025	10964	10721	10615	10454	10421	10394	10276	10250	10258	10263	9453	9620	8783	8763	8602	7263	6682	5626	4483
Gdansk	11792	11731	11684	11592	11386	11293	11489	11633	11321	11273	11037	10964	10697	10833	10541	10524	10349	10323	10509	10514	9726	9893	8903	8892	8721	7375	6755	5873	4613

Table 2.3: Distances between ports

Origin \ Destination	Port Said	Damietta	Izmit	Istanbul Ambarli	Odessa	Ilyichevsk	Constantza	Piraeus	Rijeka	Koper	Trieste	Gioia Tauro	Genoa	Fos	Barcelona	Valencia	Malaga	Algeciras	Tangiers	Le Havre	Felixstowe	Zeebrugge	Antwerp	Rotterdam	Bremerhaven	Hamburg	Gothenburg	Aarhus	Gdansk
Yokohama	7894	7925	8718	8669	9015	9013	8872	8481	9139	9178	9186	8822	9305	9405	9476	9556	9758	9804	9835	10933	11056	11083	11170	11133	11328	11416	11568	11662	11792
Shimizu	7833	7864	8650	8608	8954	8941	8811	8420	9078	9117	9120	8761	9244	9344	9415	9495	9690	9743	9763	10872	10995	11022	11109	11072	11267	11355	11507	11601	11731
Nagoya	7786	7817	8617	8561	8907	8905	8764	8373	9031	9070	9078	8714	9197	9297	9368	9448	9650	9696	9728	10825	10948	10975	11062	11025	11220	11322	11460	11554	11684
Kobe	7694	7725	8518	8469	8815	8809	8672	8281	8939	8978	8986	8622	9105	9205	9276	9356	9558	9604	9624	10736	10856	10883	10970	10933	11128	11216	11368	11462	11592
Busan	7498	7519	8329	8273	8619	8617	8476	8085	8743	8782	8790	8426	8909	9009	9080	9160	9362	9408	9440	10540	10663	10687	10774	10737	10932	11034	11175	11269	11386
Kwangyang	7443	7426	8278	8219	8539	8526	8422	8033	8691	8727	8735	8371	8854	8954	9029	9109	9275	9340	9348	10485	10605	10624	10711	10690	10877	10971	11117	11127	11293
Dalian	7591	7622	8422	8366	8712	8731	8569	8178	8836	8875	8883	8519	9002	9102	9173	9253	9455	9501	9533	10633	10756	10780	10867	10830	11025	11142	11268	11419	11489
Xingang	7735	7766	8566	8510	8856	8869	8713	8322	8980	9019	9027	8663	9146	9246	9317	9397	9599	9645	9677	10774	10897	10924	10986	10974	11169	11242	11409	11391	11633
Qingdao	7423	7454	8254	8198	8544	8563	8401	8010	8668	8707	8715	8351	8834	8934	9005	9085	9324	9333	9365	10462	10585	10612	10699	10662	10857	10974	11097	11248	11321
Liangyungang	7375	7406	8206	8150	8496	8494	8353	7962	8620	8659	8667	8303	8786	8886	8957	9037	9239	9285	9305	10414	10537	10564	10626	10614	10809	10882	11049	11025	11273
Shanghai	7139	7170	7970	7914	8260	8279	8117	7726	8384	8423	8431	8067	8550	8650	8721	8801	9040	9049	9081	10178	10301	10328	10415	10378	10573	10690	10813	10964	11037
Ningbo	7066	7097	7898	7841	8187	8185	8044	7653	8311	8350	8358	7994	8477	8577	8648	8728	8930	8976	9008	10105	10228	10255	10317	10305	10500	10573	10740	10721	10964
Fuzhou	6799	6830	7630	7574	7920	7939	7777	7386	8044	8083	8091	7727	8210	8310	8381	8461	8663	8709	8741	9841	9961	9988	10075	10038	10233	10350	10476	10615	10697
Taipei	6822	6842	7644	7591	7938	7935	7794	7412	8070	8109	8114	7753	8237	8332	8407	8487	8684	8735	8757	9864	10013	10017	10078	10081	10286	10334	10524	10454	10833
Xiamen	6653	6674	7484	7428	7774	7772	7631	7240	7898	7937	7945	7581	8064	8164	8235	8315	8517	8563	8595	9692	9815	9842	9929	9892	10087	10175	10327	10421	10541
Kaohsiung	6626	6657	7450	7401	7747	7741	7604	7213	7871	7910	7918	7554	8037	8137	8208	8288	8490	8536	8556	9668	9788	9815	9902	9865	10060	10148	10300	10394	10524
Shenzhen Yantian	6451	6482	7275	7226	7572	7592	7429	7038	7696	7735	7738	7379	7862	7962	8033	8113	8315	8361	8393	9490	9613	9640	9701	9690	9885	9956	10125	10276	10349
Hong Kong	6425	6456	7256	7200	7546	7565	7403	7012	7670	7709	7717	7353	7836	7936	8007	8087	8326	8335	8367	9467	9587	9614	9701	9664	9859	9976	10099	10250	10323
Shenzhen Chiwan	6435	6466	7266	7210	7556	7573	7413	7022	7680	7719	7727	7363	7847	7946	8034	8097	8336	8385	8410	9477	9640	9624	9711	9674	9889	9979	10141	10258	10509
Shenzhen Da Chan Bay	6440	6471	7271	7215	7561	7578	7418	7027	7685	7724	7732	7368	7852	7951	8039	8102	8341	8390	8415	9482	9645	9629	9716	9679	9894	9984	10146	10263	10514
Vung Tau	5656	5668	6478	6425	6772	6769	6628	6246	6904	6943	6948	6587	7071	7166	7241	7321	7518	7569	7591	8698	8847	8851	8912	8915	9120	9168	9358	9369	9726
Laem Chabang	5837	5806	6654	6599	6964	6951	6818	6413	7071	7124	7116	6759	7243	7340	7409	7489	7683	7736	7757	8882	9014	9018	9077	9082	9287	9332	9525	9507	9893
Singapore	5015	5036	5832	5790	6136	6123	5993	5602	6260	6299	6302	5943	6426	6526	6597	6677	6872	6925	6945	8054	8177	8204	8291	8254	8449	8537	8689	8783	8903
Tanjung Pelepas	4995	5025	5825	5770	6116	6114	5973	5582	6240	6279	6282	5923	6406	6506	6577	6657	6859	6905	6937	8034	8157	8184	8246	8234	8429	8502	8669	8763	8892
Port Klang	4834	4854	5639	5609	5943	5930	5812	5421	6079	6117	6109	5762	6245	6345	6416	6496	6679	6744	6752	7873	7996	8023	8094	8084	8268	8342	8508	8602	8721
Colombo	3492	3508	4303	4267	4607	4594	4470	4079	4737	4776	4773	4420	4903	5003	5074	5154	5343	5402	5416	6534	6657	6681	6758	6742	6926	7006	7169	7263	7375
Jebel Ali	2857	2888	3687	3632	3979	3996	3835	3444	4102	4141	4149	3785	4268	4368	4439	4519	4758	4767	4799	5899	6019	6046	6133	6107	6291	6408	6531	6682	6755
Salalah	1975	2006	2805	2750	3097	3094	2953	2562	3220	3259	3267	2903	3386	3486	3557	3637	3839	3885	3915	5014	5137	5164	5234	5214	5409	5490	5649	5626	5873
Jeddah	715	746	1539	1490	1837	1834	1693	1302	1960	1999	2007	1643	2126	2226	2297	2377	2579	2625	2657	3757	3877	3904	3991	3965	4149	4237	4389	4483	4613
Port Said	0	31	824	779	1126	1115	982	590	1248	1287	1294	931	1413	1513	1585	1665	1864	1913	1941	3042	3165	3195	3279	3256	3437	3525	3677	3771	3898
Damietta	31	0	848	793	1150	1137	996	607	1265	1307	1310	962	1434	1536	1603	1683	1877	1930	1951	3073	3196	3212	3271	3276	3468	3526	3707	3701	3867
Izmit	824	848	0	55	397	384	251	390	1150	1186	1190	837	1321	1420	1504	1588	1762	1643	1675	2798	3121	3062	3147	3126	3378	3261	3432	3535	3800
Istanbul Ambarli	779	793	55	0	347	344	203	349	1097	1137	1145	790	1273	1373	1460	1542	1731	1698	1730	2853	2976	2992	3079	3053	3248	3336	3487	3582	3829
Odessa	1126	1150	397	347	0	13	170	694	1444	1484	1492	1137	1620	1720	1807	1889	2078	2040	2156	3195	3391	3334	3421	3395	3663	3751	3903	3997	4244
Ilyichevsk	1115	1137	384	344	13	0	160	681	1441	1477	1481	1128	1612	1707	1798	1879	2075	2027	2059	3280	3305	3418	3397	3482	3687	3645	3816	3911	4184
Constantza	982	996	251	203	170	160	0	552	1300	1340	1348	993	1476	1576	1663	1745	1934	1894	2012	3049	3247	3188	3359	3249	3519	3607	3759	3853	4100
Piraeus	590	607	390	349	694	681	552	0	790	829	837	483	966	1066	1153	1235	1432	1484	1505	2614	2737	2773	2847	2834	3009	3095	3249	3343	3590
Rijeka	1248	1265	1150	1097	1444	1441	1300	790	0	103	110	581	1064	1164	1251	1338	1554	1603	1628	2732	2855	2892	2973	2953	3127	3215	3367	3461	3708
Koper	1287	1307	1186	1137	1484	1477	1340	829	103	0	8	620	1103	1202	1290	1377	1593	1642	1667	2772	2894	2931	3009	2992	3166	3254	3406	3500	3747
Trieste	1294	1310	1190	1145	1492	1481	1348	837	110	8	0	625	1109	1208	1296	1383	1598	1647	1672	2777	2925	2936	3014	3000	3205	3260	3436	3531	3804
Gioia Tauro	931	962	837	790	1137	1128	993	483	581	620	625	0	484	583	671	758	973	1022	1047	2152	2275	2311	2389	2372	2547	2635	2787	2881	3128
Genoa	1413	1434	1321	1273	1620	1612	1476	966	1064	1103	1109	484	0	208	350	508	798	847	871	1997	2100	2136	2213	2197	2372	2460	2630	2706	2953
Fos	1513	1536	1420	1373	1720	1707	1576	1066	1164	1202	1208	583	208	0	172	331	641	680	712	1830	1932	1969	2029	2030	2204	2285	2462	2538	2804
Barcelona	1585	1603	1504	1460	1807	1798	1663	1153	1251	1290	1296	671	350	172	0	161	499	513	545	1663	1786	1802	1881	1863	2058	2129	2297	2392	2639
Valencia	1665	1683	1588	1542	1889	1879	1745	1235	1338	1377	1383	758	508	331	161	0	338	382	411	1512	1635	1672	1752	1712	1907	1995	2147	2241	2488
Malaga	1864	1877	1762	1731	2078	2075	1934	1432	1554	1593	1598	973	798	641	499	338	0	65	87	1219	1343	1358	1429	1422	1627	1677	1854	1949	2222
Algeciras	1913	1930	1643	1698	2064	2054	1894	1484	1603	1642	1647	1022	847	680	513	382	65	0	32	1155	1278	1294	1370	1355	1550	1618	1789	1884	2131
Tangiers	1941	1951	1675	1730	2077	2059	1933	1505	1628	1667	1672	1047	871	712	545	411	87	32	0	1137	1273	1276	1347	1340	1545	1595	1786	1869	2154
Le Havre	3042	3073	2798	2853	3195	3280	3049	2614	2732	2772	2777	2152	1997	1830	1663	1512	1219	1155	1137	0	163	171	252	232	431	500	674	769	1015
Felixstowe	3165	3196	3121	2976	3323	3332	3179	2737	2855	2894	2925	2275	2100	1932	1786	1635	1343	1278	1273	163	0	83	141	123	303	360	530	606	872
Zeebrugge	3195	3212	3062	2992	3334	3336	3188	2773	2892	2931	2936	2311	2136	1969	1802	1672	1358	1294	1276	171	83	0	87	64	269	347	519	627	861
Antwerp	3279	3271	3147	3079	3421	3397	3280	2847	2973	3009	3014	2389	2213	2029	1881	1752	1429	1370	1347	252	141	87	0	149	356	405	597	680	965
Rotterdam	3256	3276	3126	3053	3395	3400	3249	2834	2953	2992	3000	2372	2197	2030	1863	1712	1422	1355	1340	232	123	64	149	0	215	305	467	584	809
Bremerhaven	3437	3468	3331	3248	3595	3605	3451	3009	3127	3166	3205	2547	2372	2204	2058	1907	1627	1550	1545	431	303	269	356	215	0	117	344	456	686
Hamburg	3525	3526	3261	3336	3683	3645	3539	3095	3215	3254	3260	2635	2460	2285	2129	1995	1677	1618	1595	500	360	347	405	305	117	0	402	485	770
Gothenburg	3677	3707	3432	3487	3834	3843	3690	3249	3367	3406	3436	2787	2630	2462	2297	2147	1854	1789	1786	674	530	519	597	467	344	402	0	151	368
Aarhus	3771	3701	3527	3582	3929	3938	3785	3343	3461	3500	3531	2881	2706	2538	2392	2241	1949	1884	1869	769	606	627	680	584	456	485	151	0	379
Gdansk	3898	3867	3800	3829	4176	4211	4032	3590	3708	3747	3804	3128	2953	2804	2639	2488	2222	2131	2154	1015	872	861	965	809	686	770	368	379	0

Table 2.4: Distances between ports (2)

Origin \ Destination	Yokohama	Shimizu	Nagoya	Kobe	Busan	Kwangyang	Dalian	Xingang	Qingdao	Liangyungang	Shanghai	Ningbo	Fuzhou	Taipei	Xiamen	Kaohsiung	Shenzhen Yantian	Hong Kong	Shenzhen Chiwan	Shenzhen Da Chan Bay	Vung Tau	Laem Chabang	Singapore	Tanjung Pelepas	Port Klang	Colombo	Jebel Ali	Salalah	Jeddah
Yokohama	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9826	3083	2731
Shimizu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1756	551	488
Nagoya	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7419	2328	2062
Kobe	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7891	2476	2193
Busan	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	41983	13171	11667
Kwangyang	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6358	1995	1767
Dalian	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15986	5015	4442
Xingang	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30552	9585	8490
Qingdao	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	36031	11304	10013
Liangyungang	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10608	3328	2948
Shanghai	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	87801	27546	24399
Ningbo	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	36884	11572	10250
Fuzhou	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4132	1296	1148
Taipei	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3512	1102	976
Xiamen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16436	5157	4568
Kaohsiung	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30136	9455	8374
Shenzhen Yantian	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	42727	13405	11874
Hong Kong	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	73688	23118	20477
Shenzhen Chiwan	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16023	5027	4453
Shenzhen Da Chan Bay	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5341	1676	1484
Vung Tau	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6496	2038	1805
Laem Chabang	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16230	5092	4510
Singapore	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	90837	28499	25243
Tanjung Pelepas	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	21071	6611	5855
Port Klang	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25670	8054	7134
Colombo	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12166	3817	3381
Jebel Ali	3215	575	2428	2582	13737	2080	5231	9997	11790	3471	28730	12069	1352	1149	5378	9861	13981	24112	5243	1748	2126	5311	29723	6895	8400	3981	0	0	0
Salalah	1009	180	762	810	4310	653	1641	3136	3699	1089	9014	3786	424	361	1687	3094	4386	7565	1645	548	667	1666	9325	2163	2635	1249	0	0	0
Jeddah	893	160	675	718	3818	578	1454	2778	3276	965	7984	3354	376	319	1495	2740	3885	6700	1457	486	591	1476	8260	1916	2334	1106	0	0	0
Port Said	1003	179	757	805	4285	649	1632	3118	3678	1083	8962	3765	422	358	1678	3076	4361	7521	1635	545	663	1657	9272	2151	2620	1242	0	0	0
Damietta	321	57	242	257	1370	207	522	997	1176	346	2865	1203	135	115	536	983	1394	2404	523	174	212	530	2964	687	838	397	0	0	0
Izmit	53	9	40	42	226	34	86	164	194	57	472	198	22	19	88	162	230	396	86	29	35	87	489	113	138	65	210	66	58
Istanbul Ambarli	621	111	469	499	2653	402	1010	1931	2277	670	5549	2331	261	222	1039	1904	2700	4657	1013	338	411	1026	5741	1332	1622	769	2469	775	686
Odessa	42	7	31	33	178	27	68	130	153	45	373	156	18	15	70	128	181	313	68	23	28	69	385	89	109	52	166	52	46
Ilyichevsk	32	6	24	26	137	21	52	100	118	35	287	121	14	11	54	99	140	241	52	17	21	53	297	69	84	40	128	40	36
Constantza	201	36	152	161	859	130	327	625	737	217	1796	754	85	72	336	616	874	1507	328	109	133	332	1858	431	525	249	799	251	222
Piraeus	475	85	358	381	2028	307	772	1476	1741	512	4241	1782	200	170	794	1456	2064	3560	774	258	314	784	4388	1018	1240	588	1887	592	524
Rijeka	49	9	37	39	210	32	80	153	180	53	438	184	21	18	82	150	213	368	80	27	32	81	453	105	128	61	195	61	54
Koper	116	21	88	93	496	75	189	361	426	125	1037	436	49	41	194	356	505	870	189	63	77	192	1073	249	303	144	461	145	128
Trieste	94	17	71	75	400	61	152	291	344	101	837	352	39	33	157	287	407	703	153	51	62	155	866	201	245	116	372	117	103
Gioia Tauro	947	169	715	761	4046	613	1541	2945	3473	1022	8462	3555	398	338	1584	2904	4118	7102	1544	515	626	1564	8755	2031	2474	1173	3765	1181	1046
Genoa	519	93	392	417	2216	336	844	1613	1902	560	4635	1947	218	185	868	1591	2256	3890	846	282	343	857	4795	1112	1355	642	2062	647	573
Fos	299	53	225	240	1275	193	486	928	1095	322	2667	1121	126	107	499	916	1298	2239	487	162	197	493	2760	640	780	370	1187	372	330
Barcelona	609	109	460	489	2601	394	991	1893	2233	657	5441	2285	256	218	1018	1867	2648	4566	993	331	403	1006	5629	1306	1591	754	2421	759	673
Valencia	1236	221	933	992	5280	800	2011	3843	4532	1334	11043	4639	520	442	2067	3790	5374	9268	2015	672	817	2041	11425	2650	3229	1530	4913	1541	1365
Malaga	98	18	74	79	419	63	160	305	360	106	876	368	41	35	164	301	426	735	160	53	65	162	906	210	256	121	390	122	108
Algeciras	1029	184	777	826	4397	666	1674	3200	3774	1111	9196	3863	433	368	1721	3156	4475	7718	1678	559	680	1700	9514	2207	2689	1274	4092	1284	1137
Tangiers	338	60	255	272	1445	219	550	1052	1240	365	3022	1270	142	121	566	1037	1471	2536	552	184	224	559	3127	725	884	419	1345	422	374
Le Havre	744	133	562	598	3179	481	1211	2314	2728	803	6649	2793	313	266	1245	2282	3236	5580	1213	404	492	1229	6879	1596	1944	921	2958	928	822
Felixstowe	1048	187	792	842	4480	678	1706	3260	3845	1132	9369	3936	441	375	1754	3216	4559	7863	1710	570	693	1732	9693	2248	2739	1298	4168	1308	1158
Zeebrugge	787	141	595	632	3364	510	1281	2448	2887	850	7036	2956	331	281	1317	2415	3424	5905	1284	428	521	1301	7280	1689	2057	975	3131	982	870
Antwerp	2472	442	1867	1985	10563	1600	4022	7687	9066	2669	22091	9280	1040	884	4135	7582	10750	18540	4031	1344	1634	4084	22855	5302	6459	3061	9829	3084	2731
Rotterdam	3295	589	2488	2646	14080	2132	5361	10247	12084	3558	29446	12370	1386	1178	5512	10107	14330	24713	5374	1791	2179	5443	30465	7067	8609	4080	13102	4110	3641
Bremerhaven	1534	274	1158	1232	6555	993	2496	4770	5625	1656	13708	5759	645	548	2566	4705	6671	11505	2502	834	1014	2534	14182	3290	4008	1899	6099	1914	1695
Hamburg	2371	424	1790	1904	10130	1534	3857	7372	8694	2560	21186	8900	997	847	3966	7271	10310	17780	3866	1289	1567	3916	21918	5084	6194	2936	9426	2957	2619
Gothenburg	277	49	209	222	1182	179	450	860	1014	299	2471	1038	116	99	463	848	1202	2074	451	150	183	457	2556	593	722	342	1099	345	306
Aarhus	231	41	174	186	987	149	376	718	847	249	2064	867	97	83	386	708	1005	1732	377	126	153	382	2136	495	604	286	918	288	255
Gdansk	81	15	61	65	348	53	132	253	298	88	727	305	34	29	136	250	354	610	133	44	54	134	752	175	213	101	324	102	90

Table 2.5: Demand between ports

Origin \ Destination	Port Said	Damietta	Izmit	Istanbul Ambarli	Odessa	Ilyichevsk	Constantza	Piraeus	Rijeka	Koper	Trieste	Gioia Tauro	Genoa	Fos	Barcelona	Valencia	Malaga	Algeciras	Tangiers	Le Havre	Felixstowe	Zeebrugge	Antwerp	Rotterdam	Bremerhaven	Hamburg	Gothenburg	Aarhus	Gdansk
Yokohama	3065	980	138	1622	109	84	525	1240	128	303	245	2473	1355	780	1590	3228	256	2688	883	1943	2738	2057	6457	8606	4007	6192	722	603	213
Shimizu	548	175	25	290	19	15	94	222	23	54	44	442	242	139	284	577	46	480	158	347	489	368	1154	1538	716	1107	129	108	38
Nagoya	2314	740	104	1225	82	63	396	936	97	229	185	1868	1023	589	1201	2437	193	2029	667	1467	2068	1553	4875	6499	3025	4676	545	456	160
Kobe	2462	787	111	1302	87	67	422	996	103	243	196	1986	1088	626	1277	2592	206	2158	709	1561	2199	1652	5185	6912	3218	4973	580	484	171
Busan	13096	4186	590	6929	465	359	2243	5297	547	1295	1045	10567	5788	3331	6794	13790	1094	11484	3774	8303	11700	8787	27587	36772	17118	26456	3086	2578	908
Kwangyang	1983	634	89	1049	70	54	340	802	83	196	158	1600	877	504	1029	2088	166	1739	572	1257	1772	1331	4178	5569	2592	4006	467	390	138
Dalian	4986	1594	225	2638	177	137	854	2017	208	493	398	4024	2204	1268	2587	5251	417	4373	1437	3161	4455	3346	10504	14001	6518	10074	1175	981	346
Xingang	9530	3047	429	5043	339	261	1632	3854	398	943	761	7690	4212	2424	4944	10035	796	8357	2747	6042	8514	6394	20076	26760	12458	19253	2246	1876	661
Qingdao	11239	3593	506	5947	399	308	1925	4546	470	1112	897	9069	4967	2859	5831	11835	939	9855	3239	7126	10041	7541	23676	31558	14692	22705	2648	2212	779
Liangyungang	3309	1058	149	1751	118	91	567	1338	138	327	264	2670	1463	842	1717	3485	276	2902	954	2098	2956	2220	6971	9292	4326	6685	780	651	229
Shanghai	27388	8755	1234	14491	973	751	4691	11077	1144	2709	2186	22100	12105	6966	14209	28840	2288	24016	7893	17364	24468	18376	57694	76903	35801	55329	6453	5391	1899
Ningbo	11505	3678	518	6087	409	315	1970	4653	481	1138	918	9284	5085	2926	5969	12115	961	10089	3316	7294	10279	7719	24236	32305	15039	23243	2711	2265	798
Fuzhou	1289	412	58	682	46	35	221	521	54	127	103	1040	570	328	669	1357	108	1130	371	817	1151	865	2715	3619	1685	2604	304	254	89
Taipei	1095	350	49	580	39	30	188	443	46	108	87	884	484	279	568	1154	92	961	316	695	979	735	2308	3076	1432	2213	258	216	76
Xiamen	5127	1639	231	2713	182	141	878	2074	214	507	409	4137	2266	1304	2660	5399	428	4496	1478	3251	4580	3440	10800	14396	6702	10358	1208	1009	356
Kaohsiung	9400	3005	423	4974	334	258	1610	3802	393	930	750	7585	4155	2391	4877	9899	785	8243	2709	5960	8398	6307	19802	26395	12288	18990	2215	1850	652
Shenzhen Yantian	13328	4260	600	7052	473	365	2283	5390	557	1318	1064	10755	5891	3390	6914	14034	1113	11687	3841	8450	11907	8942	28076	37423	17422	26925	3140	2623	924
Hong Kong	22986	7348	1035	12162	816	630	3937	9296	961	2273	1835	18548	10159	5846	11925	24204	1920	20156	6624	14573	20535	15422	48420	64541	30046	46435	5416	4524	1594
Shenzhen Chiwan	4998	1598	225	2644	178	137	856	2021	209	494	399	4033	2209	1271	2593	5263	418	4383	1440	3169	4465	3353	10528	14034	6533	10097	1178	984	347
Shenzhen Da Chan Bay	1666	533	75	881	59	46	285	674	70	165	133	1344	736	424	864	1754	139	1461	480	1056	1488	1118	3509	4678	2178	3366	393	328	116
Vung Tau	2026	648	91	1072	72	56	347	820	85	200	162	1635	896	515	1051	2134	169	1777	584	1285	1810	1360	4268	5690	2649	4093	477	399	141
Laem Chabang	5063	1618	228	2679	180	139	867	2048	212	501	404	4085	2238	1288	2627	5331	423	4439	1459	3210	4523	3397	10665	14216	6618	10228	1193	997	351
Singapore	28335	9058	1276	14992	1007	777	4853	11460	1184	2802	2262	22864	12523	7207	14700	29837	2367	24847	8166	17965	25314	19012	59689	79562	37039	57242	6676	5577	1965
Tanjung Pelepas	6573	2101	296	3478	233	180	1126	2658	275	650	525	5304	2905	1672	3410	6921	549	5763	1894	4167	5872	4410	13846	18455	8592	13278	1549	1294	456
Port Klang	8007	2560	361	4237	284	220	1371	3239	335	792	639	6461	3539	2037	4154	8432	669	7022	2308	5077	7154	5373	16868	22484	10467	16177	1887	1576	555
Colombo	3795	1213	171	2008	135	104	650	1535	159	375	303	3062	1677	965	1969	3996	317	3328	1094	2406	3390	2546	7994	10656	4961	7667	894	747	263
Jebel Ali	0	0	180	2110	142	109	683	1613	167	394	318	3217	1762	1014	2069	4199	333	3496	1149	2528	3562	2675	8400	11196	5212	8055	940	785	277
Salalah	0	0	56	662	44	34	214	506	52	124	100	1009	553	318	649	1317	105	1097	361	793	1118	839	2635	3513	1635	2527	295	246	87
Jeddah	0	0	50	586	39	30	190	448	46	110	88	894	490	282	575	1167	93	972	319	703	990	743	2334	3111	1448	2239	261	218	77
Port Said	0	0	56	658	44	34	213	503	52	123	99	1004	550	316	645	1310	104	1091	358	789	1111	835	2620	3492	1626	2513	293	245	86
Damietta	0	0	18	210	14	11	68	161	17	39	32	321	176	101	206	419	33	349	115	252	355	267	838	1116	520	803	94	78	28
Izmit	66	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Istanbul Ambarli	770	246	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Odessa	52	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ilyichevsk	40	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Constantza	249	80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Piraeus	589	188	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Rijeka	61	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Koper	144	46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Trieste	116	37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gioia Tauro	1174	375	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Genoa	643	206	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Fos	370	118	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Barcelona	755	241	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Valencia	1533	490	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Malaga	122	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Algeciras	1276	408	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tangiers	419	134	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Le Havre	923	295	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Felixstowe	1300	416	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Zeebrugge	977	312	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Antwerp	3066	980	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Rotterdam	4087	1306	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bremerhaven	1903	608	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Hamburg	2940	940	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gothenburg	343	110	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Aarhus	286	92	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gdansk	101	32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 2.6: Demand between ports (2)

Available ships

In Francesetti and Foschi (2002) an overview of costs related to ships with different sizes is given. The ship sizes given in this article are also used in this study. Furthermore, some additional ship sizes are added in this study. The costs of these added ships are obtained by extrapolation on the costs given in Francesetti and Foschi (2002). The available ship sizes for both the main and feeder services can be found in Tables 2.7 and 2.8. In this study, it is assumed that an unlimited number of feeder ships is available.

Ship Name	Ship Capacity (TEU)	Total Capacity (TEU/year)	Capital Cost ($/year)	Operating Cost ($/year)	Nr available
M1	4,000	208,000	4,500,000	3,600,000	5
M2	5,000	260,000	5,400,000	4,050,000	5
M3	6,000	312,000	6,000,000	4,350,000	5
M4	7,000	364,000	6,500,000	4,600,000	5
M5	8,000	416,000	7,000,000	4,850,000	5
M6	9,000	468,000	7,500,000	5,100,000	5
M7	10,000	520,000	8,000,000	5,350,000	2
M8	14,000	728,000	10,000,000	7,850,000	1

Table 2.7: Liner ship characteristics

Ship Name	Ship Capacity (TEU)	Total Capacity (TEU/year)	Capital Cost ($/year)	Operating Cost ($/year)	Fuel cost ($/nmi)
F1	200	10,400	800,000	1,450,000	16.667
F2	350	18,200	950,000	1,525,000	20.833
F3	500	26,000	1,100,000	1,600,000	25.000
F4	700	36,400	1,400,000	1,750,000	26.667
F5	800	41,600	1,500,000	1,800,000	29.167
F6	900	46,800	1,600,000	1,850,000	31.667
F7	1,000	52,000	1,750,000	1,925,000	33.333
F8	1,250	65,000	2,100,000	2,100,000	41.667
F9	1,500	78,000	2,300,000	2,200,000	50.000
F10	1,750	91,000	2,500,000	2,300,000	58.333
F11	2,000	104,000	2,700,000	2,400,000	66.667
F12	2,250	117,000	2,950,000	2,525,000	75.000
F13	2,500	130,000	3,200,000	2,650,000	83.333
F14	4,000	208,000	4,500,000	3,600,000	91.626
F15	5,000	260,000	5,400,000	4,050,000	104.264

Table 2.8: Feeder ship characteristics

Speed

From Notteboom (2006) it is learned that the speed of container vessels varies between 18 and 26 nautical miles per hour. Therefore, this range of speeds is also considered in this study. Furthermore, it is assumed that the speed can each time be increased by 0.5 nmi per hour. Thus, seventeen different values for liner shipping vessels are considered in this study.

Further, it is assumed that feeder ships sail at a constant speed. This speed is assumed to be 22 nautical miles per hour.

Capital and operating cost

In Francesetti and Foschi (2002), the yearly capital costs are given by 10% of the purchase price of the ship. The factor of 10% is the amortization factor. The purchase prices are given for ships with different ship sizes. The purchase price of the ships considered in this study, that are not given in Francesetti and Foschi (2002) are determined by extrapolation.

The operating costs are defined as 5% of the purchase price of the ship plus 1.5 times the number of crew members times the average yearly wage of the crew. The crew size is multiplied by 1.5 to take illness and holidays into account. The factor 5% of the purchase price of the ship is used to take cost of maintenance, repairs, etcetera into account. On average, 18 crew members with an average yearly wage of about $50,000 are present on a ship. The average yearly wage is obtained by correcting the yearly wage of Francesetti and Foschi (2002) for inflation.

An overview on the yearly capital and operating costs per ship size can be found in Tables 2.7 and 2.8.

Fuel cost

The fuel consumption in ton per day is given for the different ship sizes in Francesetti and Foschi (2002) for a speed of 25 nmi per hour. When this amount is divided by the distance travelled per day, the fuel consumption in ton per nautical mile is obtained. Thereafter, the fuel consumption is multiplied by the oil price in USD/ton to obtain the fuel cost in USD per nautical mile for the different ship sizes. In this study an oil price of 500 USD per ton is used in the calculations.

In Notteboom (2006) a figure is given that shows the fuel consumption in ton per day for different values of the sailing speed for a ship with capacity of almost 8500 TEU. The relation between fuel consumption and sailing speed will be about the same for different ship sizes. Therefore, this figure can be used to determine factors that indicate how much oil is consumed at different sailing speeds. Finally, these factors can be used to determine the fuel cost in USD per nautical mile for the other sailing speeds of the considered ships.

In Table 2.9 an overview of the fuel cost for the different liner ship sizes and sailing speeds is given. The fuel costs for feeder ships are obtained in a similar way and are given in Table 2.8.

Ship	Speed								
Name	18.0	18.5	19.0	19.5	20.0	20.5	21.0	21.5	22.0
M1	85.637	84.925	84.250	83.610	83.002	83.870	84.696	88.242	91.626
M2	97.449	96.639	95.871	95.143	94.451	95.438	96.379	100.413	104.264
M3	109.261	108.353	107.492	106.675	105.900	107.006	108.061	112.584	116.902
M4	121.073	120.067	119.113	118.208	117.348	118.575	119.743	124.755	129.540
M5	132.886	131.780	130.734	129.740	128.797	130.143	131.425	136.927	142.178
M6	144.698	143.494	142.354	141.273	140.245	141.711	143.107	149.098	154.816
M7	156.510	155.208	153.975	152.805	151.694	153.280	154.790	161.269	167.454
M8	203.758	202.063	200.458	198.935	197.488	199.553	201.519	209.954	218.007

Ship	Speed							
Name	22.5	23.0	23.5	24.0	24.5	25.0	25.5	26.0
M1	94.860	101.820	108.483	114.869	120.995	126.875	132.525	137.957
M2	107.944	115.864	123.447	130.713	137.684	144.375	150.804	156.986
M3	121.028	129.908	138.410	146.557	154.372	161.875	169.083	176.014
M4	134.112	143.952	153.373	162.401	171.061	179.375	187.363	195.043
M5	147.196	157.996	168.336	178.245	187.750	196.875	205.642	214.071
M6	160.280	172.040	183.299	194.090	204.439	214.375	223.921	233.100
M7	173.364	186.084	198.263	209.934	221.128	231.875	242.200	252.129
M8	225.701	242.261	258.115	273.310	287.884	301.875	315.317	328.243

Table 2.9: Fuel cost for different speeds and ship sizes

Port, (un)loading and transhipment cost

The port, (un)loading and transshipment cost are obtained from Lachner and Boskamp (2011). Port costs are incurred per port visit and usually vary between ports. Furthermore, the port costs may depend on the ship size. However, the differences in port costs are relatively small, so they are assumed to be constant per route type. In this study, ships are charged 25,000 USD per port visit on a main route and 15,000 USD per port visit on a feeder route. Thus, when a port is visited on a main route $52 \cdot 25,000 = 1,300,000$ USD is charged, because each route is performed once a week. For feeder routes, the port cost per year equals $52 \cdot 15,000 = 780,000$ USD.

(Un)loading and transshipment costs are incurred per TEU (un)loaded or transshipped in a port. These costs can differ between ports and for different ship sizes. However, it is again assumed that these costs are constant per route type. The cost of (un)loading is 175 USD per TEU on main routes and 125 USD per TEU on feeder routes. A transshipment consist of an unloading and a loading movement, so the cost of a transshipment is $2 \cdot 175 = 350$ USD on main routes. Because each port (except the cluster centers) are only visited on one feeder route and no demand exists between ports in the same cluster, no transshipments will take place on feeder routes.

Port and buffer time

The time a ship spends in a port depends on many factors like the number of containers that have to be (un)loaded, the number of cranes available to (un)load, the arrival time, etcetera. However, these factors are uncertain, so it is difficult to determine these times. Therefore, port times are assumed to be constant. The data on these times are obtained from Lachner and Boskamp (2011). In this study, it is assumed that a ship spends 20 hour in a port on a main route and 15 hours in a port on a feeder route.

The buffer time is an additional time that is added to the route time to cover delays. The causes of delays can be divided in four groups: terminal operations, port access, maritime passages and chance (Notteboom 2006). Chance includes weather conditions and mechanical problems. In this study, a buffer time of at least 2 days has to be allocated to each main route. The buffer time on feeder routes is assumed to be 1 day.

2.B Reference network

Table 2.10 shows the routes in the reference network during spring 2010. Next, Table 2.11 shows the different types of ships used on each of the routes.

AE1/AE10	AE10/AE1	AE2	AE3	AE6
Yokohama	Shenzhen Yantian	Busan	Dalian	Yokohama
Hong Kong	Hong Kong	Xingang	Xingang	Nagoya
Shenzhen Yantian	Tanjung Pelepas	Dalian	Busan	Shanghai
Tanjung Pelepas	Le Havre	Qingdao	Shanghai	Ningbo
Felixstowe	Zeebrugge	Kwangyang	Ningbo	Xiamen
Rotterdam	Hamburg	Shanghai	Taipei	Hong Kong
Hamburg	Gdansk	Bremerhaven	Shenzhen Chiwan	Shenzhen Yantian
Bremerhaven	Gothenburg	Hamburg	Shenzhen Yantian	Tanjung Pelepas
Tangiers	Aarhus	Rotterdam	Tanjung Pelepas	Jeddah
Jeddah	Bremerhaven	Felixstowe	Port Klang	Barcelona
Jebel Ali	Rotterdam	Antwerp	Port Said	Valencia
Shenzhen Da Chan Bay	Singapore	Tanjung Pelepas	Damietta	Algeciras
Ningbo	Hong Kong	Busan	Izmit	Tangiers
Shanghai	Kobe		Istanbul Ambarli	Tanjung Pelepas
Kaohsiung	Nagoya		Constantza	Vung Tau
Yokohama	Shimizu		Ilyichevsk	Shenzhen Yantian
	Yokohama		Odessa	Hong Kong
	Shenzhen Yantian		Damietta	Yokohama
			Port Said	
			Port Klang	
			Tanjung Pelepas	
			Dalian	

AE7	AE9	AE11	AE12
Shanghai	Laem Chabang	Qingdao	Shanghai
Ningbo	Tanjung Pelepas	Shanghai	Busan
Xiamen	Port Klang	Fuzhou	Hong Kong
Hong Kong	Colombo	Hong Kong	Shenzhen Chiwan
Shenzhen Yantian	Zeebrugge	Shenzhen Chiwan	Tanjung Pelepas
Algeciras	Felixstowe	Shenzhen Yantian	Port Klang
Tangiers	Bremerhaven	Tanjung Pelepas	Port Said
Rotterdam	Rotterdam	Port Klang	Piraeus
Felixstowe	Le Havre	Salalah	Koper
Bremerhaven	Tangiers	Port Said	Rijeka
Malaga	Salalah	Gioia Tauro	Trieste
Shenzhen Yantian	Colombo	Genoa	Damietta
Hong Kong	Port Klang	Fos	Port Said
Shanghai	Singapore	Genoa	Jeddah
	Laem Chabang	Damietta	Port Klang
		Port Said	Singapore
		Salalah	Shanghai
		Port Klang	
		Singapore	
		Liangyungang	
		Qingdao	

Table 2.10: Routes in the Maersk network

AE1/AE10	Capacity	AE10/AE1	Capacity	AE2	Capacity
Sofie Maersk	8,160	A.P. Moller	8,160	Maersk Seville	8,478
Albert Maersk	8,272	Skagen Maersk	8,160	Maersk Saigon	8,450
Carsten Maersk	8,160	Sally Maersk	8,160	Adrian Maersk	8,272
Maersk Singapore	8,478	Arnold Maersk	8,272	Maersk Salina	8,600
Clementine Maersk	8,648	Svendborg Maersk	8,160	Maersk Savannah	8,600
Maersk Seoul	8,450	Svend Maersk	8,160	Anna Maersk	8,272
Maersk Taurus	8,400	Columbine Maersk	8,648	Arthur Maersk	8,272
Sine Maersk	8,160	Maersk Tukang	8,400	Maersk Stepnica	8,600
Axel Maersk	8,272	Clifford Maersk	8,160	Maersk Semarang	8,400
Cornelia Maersk	8,650	Maersk Salalah	8,600	Maersk Stralsund	8,450
		Maersk Stockholm	8,600		
Average	8,365		8,316		8,439
AE3		AE6		AE7	
Maersk Kinloss	6,500	Mathilde Maersk	9,038	Eugen Maersk	14,770
CMA CGM Debussy	6,627	Maersk Antares	9,200	Elly Maersk	14,770
Maersk Kuantan	6,500	Gunvor Maersk	9,074	Evelyn Maersk	14,770
Maersk Kowloon	6,500	Mette Maersk	9,038	Edith Maersk	14,770
CMA CGM Corneille	6,500	Marit Maersk	9,038	Estelle Maersk	14,770
Maersk Kelso	6,500	Gerd Maersk	9,074	Maersk Algol	9,200
CMA CGM Musset	6,540	Maersk Altair	9,200	Ebba Maersk	14,770
Maersk Kwangyang	6,500	Gudrun Maersk	9,074	Eleonora Maersk	14,770
CMA CGM Bizet	6,627	Marchen Maersk	9,038	Emma Maersk	14,770
Maersk Kensington	6,500	Maren Maersk	9,038	Gjertrud Maersk	9,074
CMA CGM Baudelaire	6,251	Georg Maersk	9,074		
		Grete Maersk	9,074		
		Maersk Alfirk	9,200		
		Margrethe Maersk	9,038		
Average	6,504		9,086		13,643
AE9		AE11		AE12	
Maersk Sembawang	6,478	Charlotte Maersk	8,194	Maersk Kyrenia	6,978
Maersk Sebarok	6,478	Maersk Surabaya	8,400	Safmarine Komati	6,500
Maersk Serangoon	6,478	Maersk Santana	8,478	CMA CGM Belioz	6,627
SL New York	6,420	CMA CGM Faust	8,204	Safmarine Kariba	6,500
Maersk Seletar	6,478	Soroe Maersk	8,160	CMA CGM Balzac	6,251
Maersk Kendal	6,500	Susan Maersk	8,160	Maersk Karachi	6,930
Maersk Sentosa	6,478	Caroline Maersk	8,160	CMA CGM Ravel	6,712
Maers Semakau	6,478	Cornelius Maersk	8,160	CMA CGM Flaubert	6,638
Maersk Senang	6,478	Chastine Maersk	8,160	CMA CGM Voltaire	6,456
Average	6,474		8,230		6,621

Table 2.11: Ships and capacities on the Maersk network

2.C Cluster design

Table 2.12 shows the composition of the ten clusters obtained after aggregation in this study.

Shanghai	Hong Kong	Singapore	Colombo	Jebel Ali
Yokohama	Xiamen	Vung Tau	Colombo	Jebel Ali
Shimizu	Kaohsiung	Laem Chabang		Salalah
Nagoya	Shenzhen Yantian	Singapore		
Kobe	Hong Kong	Tanjung Pelepas		
Busan	Shenzhen Chiwan	Port Klang		
Kwangyang	Shenzhen Da Chan Bay			
Dalian				
Xingang				
Qingdao				
Liangyungang				
Shanghai				
Ningbo				
Fuzhou				
Taipei				

Port Said	Valencia	Rotterdam	Antwerp	Hamburg
Izmit	Gioia Tauro	Zeebrugge	Antwerp	Bremerhaven
Odessa	Genoa	Le Havre		Hamburg
Jeddah	Fos	Felixstowe		Gothenburg
Port Said	Barcelona	Rotterdam		Aarhus
Damietta	Valencia			Gdansk
Istanbul Ambarli	Malaga			
Ilyichevsk	Algeciras			
Constantza	Tangiers			
Piraeus				
Rijeka				
Koper				
Trieste				

Table 2.12: Design of the ten clusters

2.D Best Network

Table 2.13 shows the main routes of the best network, while Table 2.14 shows the feeder routes of this network.

M1	M2	M3	M4
Tanjung Pelepas	Shenzhen Yantian	Busan	Ningbo
Singapore	Shenzhen Chiwan	Qingdao	Busan
Port Klang	Shenzhen Da Chan Bay	Xingang	Qingdao
Colombo	Hong Kong	Dalian	Xingang
Gioia Tauro	Xiamen	Shanghai	Dalian
Valencia	Kaohsiung	Ningbo	Liangyungang
Algeciras	Singapore	Hong Kong	Shanghai
Felixstowe	Tanjung Pelepas	Shenzhen Chiwan	Fuzhou
Zeebrugge	Port Klang	Shenzhen Yantian	Hong Kong
Rotterdam	Port Said	Jeddah	Shenzhen Yantian
Bremerhaven	Felixstowe	Port Said	Xiamen
Hamburg	Le Havre	Bremerhaven	Kaohsiung
Rotterdam	Rotterdam	Hamburg	Algeciras
Damietta	Zeebrugge	Aarhus	Tangiers
Port Said	Port Klang	Gothenburg	Malaga
Jeddah	Tanjung Pelepas	Antwerp	Valencia
Tanjung Pelepas	Singapore	Algeciras	Fos
	Shenzhen Yantian	Valencia	Genoa
		Gioia Tauro	Barcelona
		Port Said	Gioia Tauro
		Jeddah	Ningbo
		Colombo	
		Shenzhen Chiwan	
		Hong Kong	
		Shenzhen Yantian	
		Busan	
9,000 TEU	10,000 TEU	14,000 TEU	9,000 TEU

M5	M6	M7
Shenzhen Yantian	Ningbo	Shenzhen Yantian
Shenzhen Chiwan	Qingdao	Shenzhen Chiwan
Hong Kong	Busan	Shenzhen Da Chan Bay
Xiamen	Xingang	Hong Kong
Kaohsiung	Dalian	Xiamen
Jebel Ali	Shanghai	Kaohsiung
Salalah	Jebel Ali	Jeddah
Antwerp	Valencia	Port Said
Hamburg	Felixstowe	Damietta
Bremerhaven	Le Havre	Shenzhen Yantian
Rotterdam	Rotterdam	
Zeebrugge	Zeebrugge	
Algeciras	Antwerp	
Valencia	Port Said	
Barcelona	Singapore	
Gioia Tauro	Hong Kong	
Port Said	Ningbo	
Port Klang		
Tanjung Pelepas		
Singapore		
Vung Tau		
Shenzhen Yantian		
14,000 TEU	9,000 TEU	9,000 TEU

Table 2.13: Main routes of the best network

Route	Capacity	Ports visited					
F01	350	Rotterdam	Le Havre	Felixstowe	Rotterdam		
F02	900	Shanghai	Liangyungang	Shanghai			
F03	2,000	Shanghai	Ningbo	Nagoya	Yokohama	Shanghai	
F04	500	Shanghai	Fuzhou	Shimizu	Shanghai		
F05	2,000	Shanghai	Taipei	Kobe	Kwangyang	Shanghai	
F06	1,250	Hong Kong	Xiamen	Kaohsiung	Shenzhen Da Chan Bay	Hong Kong	
F07	2,250	Singapore	Vung Tau	Laem Chabang	Singapore		
F08	1,250	Hamburg	Aarhus	Gdansk	Gothenburg	Hamburg	
F09	200	Port Said	Jeddah	Port Said			
F10	1,750	Port Said	Istanbul Ambarli	Izmit	Port Said		
F11	200	Port Said	Ilyichevsk	Odessa	Port Said		
F12	1,750	Port Said	Piraeus	Constantza	Port Said		
F13	200	Port Said	Rijeka	Damietta	Port Said		
F14	700	Port Said	Trieste	Koper	Port Said		
F15	2,250	Valencia	Tangiers	Genoa	Fos	Barcelona	Valencia
F16	200	Valencia	Malaga	Valencia			

Table 2.14: Feeder routes of the best network

2.E Solution approach

In this section, some solution methods are discussed in more detail.

2.E.1 Aggregation

First, we will describe the methods used to aggregate ports into port clusters. Thereto, we will first create lists of central, noncentral and intermediary ports. Thereafter, initial clusters will be designed, which are updated in the next step. After this step, the final clusters are known and the cluster data have to be constructed.

Lists of central, noncentral and intermediary ports

First, construct the lists of central, noncentral and intermediary ports:

$$\begin{aligned}
\mathcal{H}^c &:= \{h \in \mathcal{H} : y \geq M\bar{y}\} && \text{central ports.}\\
\mathcal{H}^{nc} &:= \{h \in \mathcal{H} : y \leq m\bar{y}\} && \text{noncentral ports.}\\
\mathcal{H}^m &:= \{h \in \mathcal{H} : h \notin \mathcal{H}^c \cup \mathcal{H}^{nc}\} && \text{intermediary ports.}
\end{aligned}$$

In this definitions, $\mathcal{H}$ is the set containing all ports, m and M are the minimum and maximum factor respectively. In our case study, we used $m = 0.2$ and $M = 2$. Further, when d_{ab} is the demand from port a to port b, we have:

$$\begin{aligned}
y_h &= \sum_{h' \in \mathcal{H}} d_{hh'} + \sum_{h' \in \mathcal{H}} d_{h'h} && \text{throughput of port } h;\\
\bar{y} &= \frac{1}{|\mathcal{H}|} \sum_{h \in \mathcal{H}} y_h && \text{average throughput per port.}
\end{aligned}$$

Initial clusters

Next, we create initial clusters. For each $h_i \in \mathcal{H}^c$, create a new cluster $C_i := \{h_i\}$ (with $i = 1, \ldots, |\mathcal{H}^c|$) only containing port h_i and the central port of the cluster is $c_i = h_i$. Let n denote the number of clusters. Then, we have $n = |\mathcal{H}^c|$ initial clusters all containing exactly one port. Next, we will add intermediary and noncentral ports to the nearest existing cluster if they are within the maximum cluster distance. We will only compare the distance between the considered port and the central port of the cluster with the maximum cluster distance, because this distance has to be covered on the feeder lines. Thus, it is possible that the distance between two ports in the same cluster exceeds the maximum cluster distance, but these ports will then not be visited on the same feeder service. Thus, for each $h \in \mathcal{H}^{nc} \cup \mathcal{H}^{m}$, we will have

$$\begin{aligned} C_i &= C_i \cup \{h\} \quad \text{if} \quad \delta_{hc_i} = \min_{1 \le j \le n} \delta_{hc_j} \le \delta^{max} \\ C_i &= C_i \quad \text{else,} \end{aligned}$$

where δ_{ab} is the distance between ports a and b and δ^{max} the maximum distance allowed between ports in a cluster. In our case study we used $\delta^{max} = 1250$ nmi, such that direct feeder lines between a port in the cluster and the central port of the cluster can always be served within one week.

Update clusters

Let

$$\begin{aligned} \mathcal{H}^{a} &:= \{h \in \mathcal{H} : h \in \cup_i C_i\} \quad \text{allocated ports.} \\ \mathcal{H}^{na} &:= \{h \in \mathcal{H} : h \notin \cup_i C_i\} \quad \text{nonallocated ports.} \end{aligned}$$

As long as $\mathcal{H}^{na} \cap \mathcal{H}^{m} \neq \emptyset$, take port $h \in \mathcal{H}^{na} \cap \mathcal{H}^{m}$ with the largest throughput ($y_h = \max_{h' \in \mathcal{H}^{na} \cap \mathcal{H}^{m}} y_{h'}$). Create a new cluster, $n = n + 1$ and $C_n = \{h\}$, $c_n = h$. Add intermediary and noncentral ports to the new cluster if they are nearest to this cluster and their distance is within the maximum distance allowed. That is, for each $h \in \mathcal{H}^{nc} \cup \mathcal{H}^{m}$, we will have

$$\begin{aligned} C_n &= C_n \cup \{h\} \quad \text{if} \quad \delta_{hc_n} = \min_{j \le n} \delta_{hc_j} \le \delta^{max} \\ C_n &= C_n \quad \text{else.} \end{aligned}$$

For all ports $h \in C_n$ check whether they were already allocated to a cluster, that is check whether

$$h \in \cup_{1 \le i \le n-1} C_i.$$

If the port was already allocated to a cluster, remove it from the cluster, so if $h \in C_j$ with $1 \le j \le n-1$, then let $C_j = C_j \setminus \{h\}$. Update the sets with allocated and nonallocated ports and repeat this procedure until $\mathcal{H}^{na} \cap \mathcal{H}^{m} = \emptyset$.

If $\mathcal{H}^{na} \cap \mathcal{H}^{nc} \neq \emptyset$, then determine for each $h \in \mathcal{H}^{na} \cap \mathcal{H}^{nc}$ to which cluster it is closest and add it to this cluster:

$$\begin{aligned} C_i &= C_i \cup \{h\} \quad \text{if} \quad \delta_{hc_i} = \min_{1 \le j \le n} \delta_{hc_j} \\ C_i &= C_i \quad \text{else.} \end{aligned}$$

The clusters C_i for $1 \leq i \leq n$ are the final clusters that will be used as input for the cargo routing model. It only remains to determine the relevant port cluster data, which will be explained in the next section.

Determine cluster data

To determine the profitability of a given network, the demand, distance and revenue between port pairs are needed. Furthermore, the (un)loading, transshipment and visit costs are needed for each port, as is the time of a port visit. The distance, costs and port time of a cluster are all incurred in the central port of the cluster, so we set:

$$\begin{aligned} \delta_{C_iC_j} &= \delta_{c_ic_j} \qquad 1 \leq i,j \leq n; \\ c^*_{C_i} &= c^*_{c_i} \qquad 1 \leq i \leq n; \\ t^p_{C_i} &= t^p_{c_i} \qquad 1 \leq i \leq n, \end{aligned}$$

where δ_{ab} is the distance between port/cluster a and b, c^*_a denote the relevant costs of port/cluster a and t^p_a is the port time of port/cluster a. Since it is not possible to determine the origin and destination port of a cargo flow between clusters when solving the cargo routing model, we let the revenue between port pairs be equal to the revenue between the central ports of the relevant clusters. That is, we let:

$$r_{C_iC_j} = r_{c_ic_j} \qquad 1 \leq i,j \leq n.$$

The demand between clusters depend on the demand between the ports in the clusters. Cluster demand equals the sum of all individual port demands in the cluster:

$$d_{C_iC_j} = \sum_{h \in C_i} \sum_{h' \in C_j} d_{hh'} \qquad 1 \leq i,j \leq n,$$

with d_{ab} the demand between ports/clusters a and b.

2.E.2 Disaggregation

In practice, it is necessary to know the exact origin and destination port of each cargo flow. Therefore, the cargo flows between port clusters have to be disaggregated into cargo flows between ports. This section will describe the method to obtain these disaggregated flows.

The disaggregation process can be performed for each combination of port clusters separately. Thus, select two port clusters C_i and C_j and the corresponding total flow over the network

$$x^{tot} = \sum_{s \in \mathcal{S}} x^{tot}_{C_i C_j s}.$$

So, the total cargo flow from cluster C_i to cluster C_j over the network is equal to x^{tot}. Now, we want to determine the origin and destination ports of the flow. Repeat the following until $x^{tot} = 0$. Select the combination (h, h') with $h \in C_i$ and $h' \in C_j$ with the largest expected revenue. Now, allocate as much flow as possible to the combination (h, h'), so

$$x^{tot}_{hh'} = \min(d_{hh'}, x^{tot}),$$

$x^{tot}_{hh'}$ is the allocated flow from port h to port h'. Update the flow to be allocated:

$$x^{tot} = x^{tot} - x^{tot}_{hh'}.$$

2.E.3 Feeder network

In the disaggregation phase, cargo flows between ports are determined. We will use feeder services to ship the cargo from and to ports in the cluster. After an initial feeder network is constructed, some methods are described to improve the network. These methods include reallocating demand in order to reduce the capacity on the feeder lines, exchanging ports between feeder lines and adding ports to the main route network. In this section, a description of these methods will be given.

Initial feeder network

In first instance, for each port in the cluster (except the central port) a direct feeder service is constructed between this port and the central port of the cluster. Let $\mathcal{F}^C$ be set of feeder services in cluster C, then

$$\mathcal{F}^C := \{(c, h, c) : h \in C \setminus \{c\}\} \qquad \text{initial feeder network of cluster } C,$$

where c is the central port of cluster C. The capacity of a line $f \in \mathcal{F}^C$ is given by

$$b_f = \min\left\{b \in Q : b \geq \max(x^{tot}_{ch}, x^{tot}_{hc})\right\},$$

where Q is the set with available capacities.

Reduce feeder capacity

The method to reduce the capacity is performed for each cluster separately. Therefore, we describe the method for a given cluster C. In the algorithm, we will determine and store the difference in profit of reducing the feeder capacity for each feeder service in the cluster separately. Thus, we select one by one the feeder services in the cluster.

Flow over legs

Let F be the selected feeder line. For this service, we first determine the flow on each leg of the feeder line. Let the legs of the line be given by $l_1, \ldots, l_n$, where n is the number of legs of the service and let the feeder route be given by $h_1, \ldots, h_n, h_1$ (leg l_1 corresponds to the leg between ports h_1 and h_2). Furthermore, let

$$x^o_{h_i} = \sum_{h \in \mathcal{H}} x_{h_i h}$$

be the total flow with origin port h_i and

$$x^d_{h_i} = \sum_{h \in \mathcal{H}} x_{h h_i}$$

be the total flow with destination port h_i. Then, the flow on leg $1 \leq i \leq n$ is given by

$$x_{l_i} = \sum_{j > i} x^d_{h_j} + \sum_{j \leq i} x^o_{h_j}.$$

Reduce capacity

Let b^c be the current capacity of the feeder service and b^n the capacity when we reduce this capacity by one size. Then, the reduction needed on leg l_i is equal to

$$y_{l_i} = \max(x_{l_i} - b^n, 0).$$

That is, we only need to reduce the flow over a leg if it is currently larger than the new capacity of the service.

Valid combinations to exchange demand

The flow over a leg can be reduced by changing the origin and/or destination port of a cargo flow. Since cargo flows between ports are obtained from cargo flows between clusters, there are multiple feasible allocations of the flow to the port pairs. We determine the port combinations between which cargo flows can be exchanged in order to reduce the flow over F.

$$\begin{aligned} &\mathcal{L}^o := \{((h_1,h_2),(h_3,h_2)) : h_1 \in F, h_2 \notin C, h_3 \in C \setminus F\} && \text{combinations with origin in } C.\\ &\mathcal{L}^d := \{((h_1,h_2),(h_1,h_3)) : h_1 \notin C, h_2 \in F, h_3 \in C \setminus F\} && \text{combinations with destination in } C.\\ &\mathcal{L} \ := \mathcal{L}^o \cup \mathcal{L}^d && \text{valid port combinations.} \end{aligned}$$

The set $\mathcal{L}$ consists of all valid combinations of port pairs between which cargo can be exchanged to reduce the cargo flow on feeder line F. If $((h_1,h_2),(h_3,h_4)) \in \mathcal{L}$, then the cargo flow over F can be reduced by increasing the satisfied demand between ports h_3 and h_4 and at the same time reducing the demand between ports h_1 and h_2 with the same amount. In this way, the total demand satisfied between clusters does not change as long as ports h_1 and h_3 belong to the same cluster and ports h_2 and h_4 belong to the same cluster. Furthermore, we only want to change the flow over the feeder network in cluster C, so we will add the restriction that the port that does not belong to C is not allowed to be changed. In $\mathcal{L}^o$ the origin port belongs to C and thus we see that the destination port in both port pairs is the same (namely h_2) and, similarly, for $\mathcal{L}^d$ the origin port (h_1) is the same for both pairs. Furthermore, we want to reduce the cargo flow over line F, so we do not want to shift cargo from one leg of service F to another leg. Therefore, we add the restriction that the new port in cluster C is not allowed to be on line F. Thus, port h_3 in the definition of both $\mathcal{L}^o$ and $\mathcal{L}^d$ is an element of $C \setminus F$ (all ports in C that are not visited on line F).

Exchange demand between port pairs

For each combination $((h_1,h_2),(h_3,h_4))$ of port pairs in $\mathcal{L}$, the revenue decrease per unit of exchanging cargo from the first demand pair to the second is given by $r_{h_1h_2} - r_{h_3h_4}$, where r_{ab} denotes the revenue of satisfying one unit of demand from port a to port b.

Select the combination with the lowest decrease in revenue. We want to exchange as much cargo as possible from the first port pair to the second pair. Clearly, the maximum amount that can be exchanged is bounded by the amount of cargo that is currently transported between the first port pair and the unsatisfied demand between the new port pair. Furthermore, it is bounded by the free capacity of the feeder line over which the new flow has to be transported.

Let h^o denote the port in C of the first port pair, h^n the port in C of the second port pair and h' the port not in C that is part of both port pairs. Furthermore, let d_{ab} and d_{ab}^{sat} be the demand and satisfied demand between ports a and b respectively. The demand to be exchanged is bounded by

$$d_d^{ex} = \begin{cases} \min(d^{sat}_{h'h^o}, d_{h'h^n} - d^{sat}_{h'h^n}) & \text{if } h' \text{ is the origin port,} \\ \min(d^{sat}_{h^o h'}, d_{h^n h'} - d^{sat}_{h^n h'}) & \text{if } h' \text{ is the destination port.} \end{cases}$$

Exchanging demand will also change the flow on the feeder lines containing h^o and h^n, but only the flow on the new feeder line is relevant in this case, because this flow will be increased. Let F^n be the new feeder service. The definition of $\mathcal{L}$ guarantees that $F^n \neq F$. If h' is the origin port of the pairs, then the flow between the port pairs will be on the feeder line for all legs before port h^n, while it will be on the feeder line for all legs after port h^n if h' is the destination port. Let h^n be the k-th port of feeder line F^n and let n the length of the feeder line. Furthermore, let $x^n_{l_i}$ be the flow on feeder line F^n over leg l_i. Then, the amount of cargo that can at most be exchanged is bounded by

$$d_f^{ex} = \begin{cases} \min\limits_{1 \le i < k} b_{F^n} - x^n_{l_i} & \text{if } h' \text{ is the origin port,} \\ \min\limits_{k \le i \le n} b_{F^n} - x^n_{l_i} & \text{if } h' \text{ is the destination port.} \end{cases}$$

Thus, the maximum amount of cargo that can be exchanged between the combinations of port pairs is given by:

$$d^{ex} = \min(d_d^{ex}, d_f^{ex}).$$

The flow over the feeder services has to be updated when we exchange this amount. Thereto,

$$x^{old}_{l_i} = \begin{cases} x^{old}_{l_i} - d^{ex} & 1 \le i < j & \text{if } h' \text{ is the origin port,} \\ x^{old}_{l_i} - d^{ex} & j \le i \le m & \text{if } h' \text{ is the destination port,} \end{cases}$$

where port h^o is the j-th port on feeder line F and $x^{old}_{l_i}$ is the flow over leg l_i of feeder line F and m is the length of feeder service F. Similarly,

$$x^{new}_{l_i} = \begin{cases} x^{new}_{l_i} + d^{ex} & 1 \le i < j & \text{if } h' \text{ is the origin port,} \\ x^{new}_{l_i} + d^{ex} & j \le i \le m & \text{if } h' \text{ is the destination port.} \end{cases}$$

The costs of exchanging the demand is given by:

$$C^{ex} = C^{ex} + \begin{cases} (r_{h'h^o} - r_{h'h^n} + c^h_{h^n} - c^h_{h^o})d^{ex} & \text{if } h' \text{ is the origin port,} \\ (r_{h^o h'} - r_{h^n h'} + c^h_{h^n} - c^h_{h^o})d^{ex} & \text{if } h' \text{ is the destination port,} \end{cases}$$

where c_h^h is the handling cost per unit in port h and C^{ex} is initialized at 0 each time we consider a new feeder service F.

Next, we can update the reduction needed on each leg and repeat this procedure until either all valid combinations of port pairs are considered or the reduction needed equals zero on each leg. If the capacity of the feeder service can be reduced (the reduction needed equals zero for each leg of the service), then the profit is given by

$$P^{ex} = c^c_{b_c} + c^f_{b_c} - (c^c_{b_n} + c^f_{b_n}) - C^{ex},$$

where c_b^c is the capital and operating costs on the feeder line F when a ship with capacity b is used and c_b^f is the fuel costs on F for a ship with capacity b.

Exchange port between feeder services

Next, we describe the method to exchange a port between two feeder services. The cargo allocation is not changed in this method, so we can consider the different clusters separately. Thereto, we first select a cluster C. For each combination of two feeder lines in cluster C and each port on the first feeder line, we will consider the increase in profit when we exchange this port from the first service to the second service. Thus, we select two feeder service F and F' in C. Furthermore, let N be a noncentral port on feeder line F. Then, we will determine at which location it is most profitable to add port N to line F' and how large the profit increase is.

Let (P, P') be a consecutive port combination on feeder service F'. First, determine the cost of the feeder services F and F' as they are before we exchange a port:

$$C^{old} = c^c_{b_f} + c^f_{b_f} + c^c_{b_{f'}} + c^f_{b_{f'}},$$

where b_f and $b_{f'}$ are the capacities of F and F' respectively and c_b^c and c_b^f are the capital and operating and fuel costs for a ship with capacity b respectively.

Let N now be visited in between ports P and P' on feeder service F', that is, remove leg (P, P') from F' and add legs (P, N) and (N, P') to F'. Since N will now be visited on line F', we remove it from line F. The method described in Section 2.E.3 can be used to determine the new flows on the feeder services F and F', because the satisfied demand between port pairs is known. When the flow over each leg is known, the capacity of the feeder service can be determined by:

$$b_f = \min\left\{b \in Q : b \geq \max_{1\leq i\leq n} x_{l_i}\right\},$$

where the feeder service is given by $l_1, \ldots, l_n$.

The costs of the feeder services F and F' after exchanging port N can again be calculated by:

$$C^{new} = c^c_{b_f} + c^f_{b_f} + c^c_{b_{f'}} + c^f_{b_{f'}},$$

where b_f and $b_{f'}$ are now the new capacities on the feeder lines. The increase in profit is given by:

$$P^N = C^{old} - C^{new}.$$

Repeat this procedure until all consecutive port combinations on F' are considered. Then, repeat until all noncentral ports on F are considered.

Add ports to main routes

In the aggregation phase, we decided to create port clusters in order to reduce the computation time of the cargo-routing model. Central ports of the clusters are visited on the main route network, while all other ports are currently only visited on the feeder network. However, it might be profitable to visit some of those ports on the main network. Ports are clustered based on distance to the central port, so if the central port is visited on a main route, the additional distance that has to be sailed in order to include a noncentral port to the main route will in general be quite small. In this section, we describe a method to add noncentral ports to the main routes.

First, select a main route r and determine the clusters $C_1, \ldots, C_n$ that are visited on r. If a cluster is visited twice on a route, we consider it to be two different clusters. So, a distinction is made between the cluster when it is visited on the eastbound part of the route and the cluster when it is visited on the westbound part of the route. So, each cluster that is visited on a route is unique for the route. Consider a cluster C on route r and let N be a port that belongs to cluster C and is not yet visited on route r, that is, $N \in C \setminus r$. Let (P, P') be a consecutive port combination on main route r, satisfying $P \in C$ and/or $P' \in C$. Furthermore, let F be the feeder service on which port N is currently visited ($N \in F$).

Cargo reallocation

Let N now be visited in between ports P and P' on main route r, that is, remove leg (P, P') from r and add legs (P, N) and (N, P') to r. Since it is probably not feasible to (un)load all cargo from/to port N on route r (some cargo might be on a different route, or the ship capacity will not suffice to transport all cargo directly via route r), we cannot remove port N from the feeder line F. We want to reallocate as much cargo as possible from feeder service F to main route r, because the handling and transshipment costs will be reduced in this way. The amount of cargo that can be reallocated is first of all restricted by the total amount of cargo from/to port N that is present on the ship. Furthermore, it depends on the unused capacity of the ship on the additional legs. To determine how much cargo can be reallocated according to the unused capacity of the ship, first the position of the inserted port N with respect to the center of the cluster has to be determined.

Two situations can be distinguished: the central port of the cluster is already visited when port N is visited on the main route, or the central port of the cluster has still to be visited when port N is visited. Figure 2.3 shows the two possibilities. In the figures, only the central ports of the clusters and port N are considered, but all conclusions that will be drawn, will also hold when more ports are on the route.

Now, consider the left figure, where port N belongs to cluster C and is visited after the central port of the cluster. In the original route, the ship visits first the central port c of cluster C and directly thereafter the central port c' of the cluster C'. Thus, the cargo flows from and to port N are (un)loaded in c. Now, let $x_{cc'}$ be the flow from port c to port c'. The cargo flow with destination port N will be unloaded in port c, so this flow is not included in flow $x_{cc'}$. On the other hand, the cargo flow with origin port N is included in flow $x_{cc'}$, because it is loaded in port c.

When port N is added to the main route after the central port c in the cluster, flows x_{cN} and $x_{Nc'}$ have to be determined. The difference with the original situation is that the cargo flow from and to port N is now (un)loaded in port N instead of in port c. Thus, the cargo flow to port N is included in flow x_{cN}, while the flow from port N is not included. Combining this with the flows included in flow $x_{cc'}$, it can be seen that $x_{cN} = x_{cc'} - N_f^{out} + N_f^{in}$, where N_f^{in} is the amount of cargo flow unloaded in port N (flow with port N as destination) and N_f^{out} is the amount of cargo flow loaded in port N (flow with origin port N). In flow $x_{Nc'}$ the cargo flow to port N is not included, where the flow from port N is included, so it holds that $x_{Nc'} = x_{cc'}$. Thus, N_f^{out} is not included in x_{cN} and included in both $x_{cc'}$ and $x_{Nc'}$, so all cargo with origin port N on ship route r can be loaded in port N, without exceeding the capacity of the ship. However, the amount of cargo that can be unloaded in port N is bounded by $N_f^{in} \leq b_r - x_{cc'} + N_f^{out}$, since this cargo is included in the new flow x_{cN} but not in the old flow $x_{cc'}$.

The other situation is shown in the right figure. In this case, port N belonging to cluster C is visited before the central port c of the cluster. The flow on the initial route between central ports c' and c is denoted by $x_{c'c}$. In this case, the flow to port N is included in flow $x_{c'c}$, while the flow from port N is not included, because it will be loaded

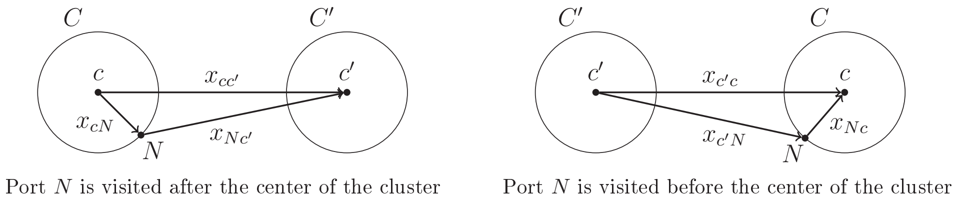


Figure 2.3: Example of the positioning of port N with respect to the central ports c and c' of the cluster.

in the central port c of cluster C. Now, consider flow $x_{c'N}$ between central port c' and port N. In this flow, the cargo flow to port N is included and the cargo flow from port N is not included. Thus, in this case $x_{c'N} = x_{c'c}$. The cargo flow to port N is now unloaded in port N, so this flow is not included in x_{Nc}. However, the flow from port N is already loaded in port N, so is included in x_{Nc}. Together with the flows included in $x_{c'c}$, it can be found that $x_{Nc} = x_{c'c} - N_f^{in} + N_f^{out}$, where N_f^{in} and N_f^{out} have the same definitions as above. Now, all cargo to port N can be unloaded without exceeding the capacity of the ship when sailing to port N, but the amount of flow that is loaded is bounded by $N_f^{out} \leq b_r - x_{c'c} + N_f^{in}$, since this cargo is included in the new flow x_{Nc} but not in the old flow $x_{c'c}$.

The amounts of flow (un)loaded in port N are equal to

$$N^{out} = \min(N_f^{out}, N_s^{out})$$

and

$$N^{in} = \min(N_f^{in}, N_s^{in}),$$

where N_s^{out} and N_s^{in} are the amounts of cargo present on the ship on route r with port N respectively as origin and destination port. Thereafter, the new flows can be determined using the formulas for x_{ab} given above. Furthermore, N^{in} and N^{out} can be used to update the flows on the feeder service visiting port N by subtracting the flows from the legs over which it should be transported. When no flows are loaded and unloaded anymore in port N on the feeder service, the port can be deleted from the feeder service.

Cost reduction

Which costs incurred at the main and feeder network are relevant in this method depend on where in the solution algorithm this procedure is executed. When the methods to reduce the feeder network are not yet performed, only the handling costs on the feeder network are considered to be relevant. However, when these methods are already performed also the capital, operating, fuel and port costs of the feeder network are relevant. The handling, port, capital, operating and fuel costs of the main network are relevant in both cases.

The route costs c^r_r and $c^r_{r'}$ consisting of the capital, operating, port visit and fuel costs of the main routes r and r' can be obtained from the method to determine the optimal speed, which is described in Section 2.3.2. The new capacity needed on the feeder route can be obtained from the method described in Section 2.E.3. Thereafter, the new route costs of the feeder route can easily be computed by adding the capital, operating, port visit and fuel costs, because the route duration and speed are fixed. Furthermore, the difference in handling costs can be obtained using N^{in} and N^{out}.

After the cost reduction is determined, repeat this procedure for a new consecutive port combination (P, P') on route r with $P \in C$ and/or $P' \in C$ as long as they are not all considered yet. Next, repeat until all noncentral ports $N \in C \setminus r$ are considered, all clusters $C \in r$ are considered and finally until all routes r are considered. Then, add the port for which the cost reduction is largest to the main route and at the location where this cost reduction will be obtained. This method is repeated until no cost reduction can be obtained anymore by adding a port to a main route.

Chapter 3

Will liner ships make fewer port calls per route?[1]

3.1 Introduction

The growth in container trade has led to substantial increases in container ship sizes. Larger ships are well known to benefit from economies of scale at sea, but they may suffer from diseconomies of scale in ports. Cullinane and Khanna (1999) performed a study to investigate the (dis)economies of scale in large container ships both in port and at sea. They considered container ships varying in capacity from 200 to 8,000 TEU. Their findings show that diseconomies in ports exist for ships larger than 1,500 TEU, but that the magnitude is quite small. Furthermore, they show that economies of scale at sea exist at least for ships up to 8,000 TEU and the economies of scale at sea clearly outweigh the diseconomies of scale in ports. Although the data used by the authors is outdated, the general observations will most probably still be valid.

The increase in container ship size also has its consequences for the network structure in liner shipping. Not all ports are capable of handling large container ships. Furthermore, it might not be profitable to call at relatively low-demand ports with large-size container ships. Therefore, traditional liner networks consisting of so-called circular, butterfly and pendulum routes may shift to more hub-and-feeder like networks. In circular routes, ports are all visited exactly once on each rotation, while in butterfly routes ports can be called at twice or more during a rotation. Pendulum routes are a special type of butterfly routes in which the same ports are called at on the east- and westbound trip, only in reversed

[1]This chapter is based on the online report of the Econometric Institute Report Series with small modifications: Mulder, J. and Dekker, R. Will liner ships make fewer port calls per route? Report number: EI 2016-04.

order. Hub-and-feeder networks consist of a small number of large hub ports, which are connected to each other. All other (smaller) ports in the networks are called spokes or feeder ports and are only visited on routes originating from and destined for their closest hub. In South America, shipping routes have recently been reconstructed towards networks similar to hub-and-spoke networks. Furthermore, non-stop services between regions with high demand have been introduced (Sanchez and Wilmsmeier 2011).

Call sizes at terminals have increased as well, as a consequence of the total growth in container trade. This poses problems for terminals, as they face a larger peak load for the stack. Yet larger call sizes on bigger ships also benefit terminal quay crane productivity as cranes can work longer on a bay.

All in all, this raises the question whether container carriers should reduce the number of port visits on a string in order for terminals to be more productive and reduce unproductive port time. In this research, we will also investigate whether a change to hub-and-feeder networks is to be anticipated. Next, we will review literature; first, we will discuss literature related to hub-and-feeder networks and thereafter literature related to traditional liner shipping networks.

Fagerholt (2004) considers the problem of determining the optimal regional network design. The proposed solution approach consists of two stages. First, all feasible routes are generated and then an integer programming problem is solved to select the optimal routes from the set of feasible routes. Our regional network design solution approach is based on this approach.

Imai et al. (2009) examined the profitability of two different types of service networks under several scenarios. They compared a multi-port network with conventional ship sizes with a hub-and-spoke network with mega-ships. The research shows that multi-port networks are more profitable than hub-and-spoke networks except for European shipping companies serving the Asia-Europe trade lane. The hub-and-spoke model in Imai et al. (2009) only allows for direct feeder routes between the hub and the other ports. In our networks, multiple port calls on a feeder route are allowed, which will most probably increase the profitability of a hub-and-spoke network. Further, Imai et al. (2009) use different cost structures and ship types for the multi-port and hub-and-spoke networks. In our research, we will show that even with the same cost structures and ship types, hub-and-spoke systems can be more profitable than multi-port networks. Furthermore, Hsu and Hsieh (2007) consider a ship allocation, sailing frequency and cargo routing problem on a predefined hub-and-spoke network. A two-objective model is used for which Pareto optimal solution curves are presented. The authors compare the performance of the network when cargo is routed directly from a feeder origin port to the destination port with the performance when the cargo is routed via a hub. For some ports direct shipping is preferred over shipping via a hub, while for other ports routing via a hub is less costly.

Gelareh and Pisinger (2011), Gelareh et al. (2013) and Zheng et al. (2015) use mixed integer programming models to formulate the simultaneous hub location, feeder port allocation, fleet deployment and network design problem. In both Gelareh and Pisinger (2011) and Gelareh et al. (2013), networks can contain only one hub route visiting all hubs exactly once. Furthermore, in Gelareh and Pisinger (2011) only direct feeder services are allowed, but feeder ports can be connected to multiple hubs. In Gelareh et al. (2013) multiple feeder ports can be visited on one feeder route, but each feeder port is only visited exactly once in the network. Respectively, a Benders and a Lagrangian decomposition approach are proposed in these works to solve problems that cannot be solved using existing solvers. Zheng et al. (2015) solve the problem using a genetic algorithm embedded with a multi-stage decomposition approach. The latter three papers do not give any indication of whether hub-and-feeder systems are better than multi-port networks.

In the previous works, the determination of which ports are used as hubs and the allocation of the other ports to the hubs are both incorporated in the mixed integer programming model. In Mulder and Dekker (2014) on the contrary, hub selection and port allocation are solved as separate problems before considering the network design and fleet deployment problems. Thereafter, the network design and fleet deployment problem is solved using a genetic algorithm approach. This work is comparable to the work in Mulder and Dekker (2014) as it uses the same idea: we first cluster ports before considering the network design problem. However, this research uses an improved formulation of the cargo routing problem, leading to better solutions in less computational time. Furthermore, in this work, we use an iterative MIP-based algorithm that guarantees that the network improves in each iteration instead of a genetic algorithm based approach. The iterative solution algorithm is easier to understand and needs less computational time. Finally, in this work, we analyze the effect of adding a large non-hub port to the main network, but we do not include it in the solution algorithm, because the cost reductions are only marginal, while Mulder and Dekker (2014) include different methods to add smaller ports to the main route network to their solution approach in order to improve the network profit.

Xia et al. (2015) study the joint fleet deployment, speed optimization and cargo routing problem. The authors incorporate a new fuel consumption function based on both speed and load of the ships. The problem is solved by clustering ports into a few large regions and construct routes between the regions. The authors do not consider the routes to the individual ports within each region.

A lot of research has been performed on network design in traditional liner shipping networks (for reviews on these works, see e.g. Ronen 1983, 1993, Christiansen et al. 2004, Meng et al. 2014). Brouer et al. (2014a) provide a benchmark model and data set based on real data from Maersk Line, which makes it possible to compare networks. The current best results for this benchmark data for Europe-Asia instances are found using the method of Brouer et al. (2014b). In this paper, we will use the Europe-Asia data from this benchmark paper and compare our results to the results of Brouer et al. (2014b) of which some corrections are reported in Brouer (2015).

In this paper we will investigate the profitability of making fewer port calls in two ways. First we will describe the considered problem in Section 3.2. In this section, we will also discuss the influence of making fewer port calls per route on different aspects that cannot always easily be captured in a mathematical model. Next, Section 3.3 describes the methods used to design a hub-and-feeder network. In Section 3.4, we conduct a case study for the Asia-Europe trade lane by applying the optimization methods of Section 3.3 on top of a pre-specified hub-and-feeder system, using demand data published by Brouer et al. (2014a). Next, Section 3.5 provides a improvement heuristic that adds an additional port to a route, whereafter conclusions are drawn in Section 3.6.

3.2 Problem description

In this paper we will focus on designing networks with a special structure. Our networks will be similar to so-called hub-and-feeder networks in which first a route network is constructed between a few large hub ports. Thereafter, all other ports are allocated to one of the hub port and for each hub port a regional network (also called feeder network) is constructed calling at the allocated hub ports. The hub-and-feeder structure of the considered networks is inspired by the big container ships that are used these days. These big ships are expected to be most profitable when large call sizes are realised during port calls, because port visits are expensive and calls with large call sizes will probably be more efficient than small calls. Ideally, big ships will be as full as possible during the complete route and provide (approximately) direct connections between multiple ports. A hub route network with a limited number of hubs is likely to satisfy these conditions, because the total regional demand will be transported between the different hubs.

The problem studied in this paper can now be described as follows. Consider a given set of demands between origin and destination ports (also referred to as OD-pairs) and a set of available ships. The goal is to design a liner shipping network that can be used to transport the demand from the origin to the destination ports. The network will consist of a set of routes that will be sailed. Each route specifies which ports will be called at the route and in which order. A ship type has to be allocated to each route in such a way that each port on the route is visited an integer number of times each week. Clearly, only available ships can be used to allocate to routes. Furthermore, the sailing speed for each ship needs to be decided upon and should be in between the minimum and maximum allowed sailing speed of the allocated ship. Finally, the exact routing of cargo over the routes in the network needs to be determined. Transshipments might be used in order to satisfy a demand against a given cost. The liner company receives a revenue for each container transported, but also incurs loading and unloading cost of the container. The company is allowed to reject containers against a given cost, denoting the loss of goodwill.

3.2.1 Analysis of influence of fewer port calls

A hub-and-feeder network will differ from traditional liner shipping networks in multiple aspects. An important difference is that traditional networks have more port calls per route than the networks that we will generate in this research. In this section, we will describe the influence of fewer port visits on the demand structure, transit time, (dis)economies of scale, uncertainty in port time, CO_2 emission, cost allocation and flexibility and competition.

Demand structure

Reducing the number of port calls per route will result in large fast container flows between the hubs. In this paper we assume that all demand has to be transported from its given origin to the given destination port. However, when for example Rotterdam will become a hub port with non-stop connections to Asia, it is inevitable that the demand at ports close to Rotterdam, like Antwerp and Hamburg, will (partly) shift towards Rotterdam. Containers can be transported by truck, train or barge from Rotterdam to Antwerp and Hamburg or directly to the real destination of the container. In general, a demand shift towards the hub ports is to be expected, because hub ports offer fast and frequent connections to other hub ports. Hinterland connections will be used to deliver containers at the hub ports, increasing the demand at hubs. This results in less transshipments in the networks, which is beneficial for the hub networks. Hence, the hub networks might perform better than indicated by the results in this paper, where we did not consider the demand change.

Transit time

The influence of introducing a hub-and-feeder network on transit time is considered by distinguishing between two different types of demand pairs: pairs of which both ports are hub ports in their region and pairs of which at least one port is a regional port.

Hub services provide more efficient transport between hub ports than traditional liner network, because fewer port stops per route are made. This will probably decrease the transit time for cargo demand between two hub ports. However, the second category is more difficult to evaluate. In the hub-and-feeder network, many of these demand pairs will need at least one transshipment, because at least one of the ports is not a hub. Assuming weekly port calls in liner shipping networks, the containers may have to wait up to one week for the connecting ship. In traditional liner shipping networks, the ports might be called at the same route, in which case no transshipment is needed. However, also in traditional networks, transshipments are likely, especially for small ports that are usually only visited at a few routes in the network. Furthermore, a connection between two ports on the same route is probably more efficient in hub-and-feeder networks than in traditional networks because of the reduction in port calls. With fewer port calls, the total distance of a route will probably decrease because ships need to sail less additional distance in order to make port calls. Furthermore, the route time decreases, because fewer port visits also means less port time. Hub networks consist of only a few ports, so the same port combination might be visited on multiple routes, increasing the frequency between hubs. Therefore, the additional time required for additional port calls on traditional routes and consequently the additional distance to be sailed can easily become larger than the transshipment time needed in a hub-and-feeder service. A disadvantage of hub-and-feeder networks is that regional origin (respectively destination) ports might be located closer to the destination (respectively origin) port than the hub port where the cargo has to be transshipped, so some backtracking has to occur. In this case, the distance between the origin (respectively destination) port and the hub has to be sailed twice. This disadvantage can partly be accounted for in the clustering algorithm, where this additional distance can be incorporated in the decision to allocate the regional port to a hub port.

In conclusion, the effect of a hub-and-feeder network on the transit time is not necessarily negative, but more research has to be done to draw exact conclusions. Transit time can be incorporated in the network design problem, but will increase the complexity of the problem even more, since trade-offs need to be made between costs and transit time. Therefore, we decided not to include the transit time as decision variables in our problem.

Economies of scale

Third, (dis)economies of scale influences the profitability of networks. Larger ships are well-known to be more efficient at sea, but they might be less efficient in ports (see for example Cullinane and Khanna 1999). However, in this research we investigate the reduction in port calls per route using the same ship types as are currently used by liner companies. The advantage of making fewer port stops compared to services calling at multiple ports is that more containers on the ship have to be unloaded during a port call. This reduces the complicated problem of container placement on ships and decreases the probability that containers are blocking other containers. However, the disadvantage of having fewer port calls is that more containers have to be loaded and unloaded during a port call, which might influence the port time. In the case study, we do not incorporate this aspect; all port calls take the same amount of time consistent with the approach in Brouer et al. (2014a). Furthermore, high container volumes are transported over a hub service. The high volumes justify the use of even larger container ships on hub services. In this way, shipping lines can benefit even more from economies of scale.

Uncertainty in port time

The next aspect we will consider is the uncertainty in port time. Port time uncertainties can arise from many different factors such as port/terminal congestion, unexpected waiting times before berthing or before starting loading/discharging and port/terminal productivity below expectation (Notteboom 2006). The probability of obtaining one of these delays increases when ships arrive delayed in a port, thereby missing their allocated time slots. Delay management is a very important issue in liner shipping. Shipping lines face high operational costs per day, so delays can be very costly (Vernimmen et al. 2007). Furthermore, shippers are faced with the possibility of losing customers. Therefore, shipping lines will try to maximize their schedule reliability. Ships sailing on services with fewer port calls will spend relatively more time at sea. Fewer port calls will clearly lead to a lower probability of incurring delays in ports. Furthermore, average sailing distances per sea leg increase when fewer ports are called per route, which gives rise to the possibility of recovering earlier obtained delays by increasing the sailing speed. Clearly, delays can always be recovered (at least partially) by increasing the sailing speed, but larger sailing distances imply smaller speed increases to capture the same amount of delay. Since daily bunker costs are usually assumed to be proportional to the third power of the sailing speed (Stopford 2009), larger increases in sailing speed can have disastrous consequences on the bunker costs. Hence, services with fewer port calls are likely to recover from incurred delays in a less costly manner compared to other liner services, resulting in more timely port arrivals and thus lower probabilities of incurring port delays. Another factor influencing the port time uncertainty is the variation in call size. A hub service will typically be used to transport the cargo demand from one region to another region. Individual port demand is thus aggregated to port region demand, which will in general also increase the call size uncertainty. However, this effect is compensated by the decrease in number of port calls on the service. Which of these two effects will dominate the other depends on multiple factors, like correlation between call sizes.

Concluding, in general hub services might be able to decrease the port time uncertainty, because on-time arrivals can better be managed. However, uncertainty in call size can endanger the port time reliability.

CO_2 emissions

Next, we will consider the difference in CO_2 emissions between transport using hub services and transport using traditional liner services. Again, the total effect is difficult to estimate beforehand and depends on the exact network structure. On one hand, more transshipment movements are needed to transport the cargo from the origin to the destination port using hub-and-feeder networks, adding more CO_2 emissions to the process. On the other hand, hub services are more efficient at sea, decreasing the total CO_2 emissions. Therefore, no exact conclusions can be drawn with respect to the influence of hub-and-feeder networks on the CO_2 emissions. The amount of CO_2 emissions for a given route will be proportional to the bunker consumption at the route. Hence, after obtaining different networks, the amount of CO_2 emissions can easily be compared to each other. However, it is much more difficult to incorporate the amount of emissions in the optimization model, since trade-offs between total network costs and emissions have to be made.

Cost allocation

Liner companies are using shipping networks to provide connections between ports to deliver demand. The total network performance can easily be determined by determining the total costs and revenues of the network. In practice liner companies are often also interested in the cost or benefit of a single OD-pair. However, these costs are much more difficult to estimate, because multiple OD-pairs share connections and it is not obvious which part of the costs should be allocated to which OD-pair. This problem is referred to as the cost allocation problem. In traditional networks, this is a very difficult problem, because OD-pairs can usually be serviced using multiple different connections, which makes it unclear which OD-pairs should contribute to the costs of an individual route. Cost allocation will become more straightforward when using hub-and-feeder networks, because there will be fewer connections between OD-pairs. The extreme case in which the hub route consists of a non-stop connection between two ports, will clearly reveal the exact route of an OD-cargo.

Flexibility and competition

Finally, we compare the flexibility and competition potentials between hub-and-feeder and traditional liner shipping networks. Clearly, flexibility increases when more ports are visited on a route. Since the distance between two consecutive ports is usually smaller on routes with more port calls, ships have more opportunity to for example swap port calls if no berth is available in one of the ports at the expected arrival time. Furthermore, more port calls in a region might allow transport providers to change the origin port in case they are going to miss their connection, which also increases the potential for competition between ports. If only one port is visited in each region, the competitive position of all other (smaller) ports will weaken strongly. Hence, traditional networks provide both more flexibility and more competition potentials.

3.3 Solution methodology

This section discusses the hub-and-feeder network design problem. The goal is to construct a network satisfying the hub-and-feeder design. We propose an iterative solution algorithm in which the hub and feeder routes are iteratively updated given that the other routes are fixed. Algorithm 4 gives a description of the iterative solution approach. We first need to select the potential hub ports from the data. Then, given this set of hubs, clusters need to be designed. This is done in Step 1 of the algorithm using Algorithm 5, which will be described in Section 3.3.1. Next, we want to construct a route network consisting of hub and feeder routes. The feeder routes are also referred to as regional routes. Each route is denoted by a string of ports. The order of the ports denote the order in which they are visited on the route. Ports can be visited multiple times on a route. We require a weekly frequency of each route in the network. The hub and feeder subnetworks can be generated separately from each other. That is, given a set of feeder routes, the connecting hub routes can be optimized, while the optimal regional routes can be found given a realization of the satisfied demand obtained with fixed hub routes. Sections 3.3.2 and 3.3.3 describe respectively the network design of the hub and regional route networks.

The initial hub network in Step 2 is constructed as will be described in Section 3.3.2. Then, the initial regional route network can be constructed by solving the regional route network design (RRND) problem that will be introduced in Section 3.3.3 with as initial demand the real demand between each OD-pair. That is, in the construction of the initial regional route network, we assume that all demand should be satisfied. Furthermore, each cluster is allowed to use all available ships in the dataset. After constructing the initial hub and regional route network, the ship allocation and cargo routing (SACR) problem that will be described in Section 3.3.4 can be solved with a time limit. In this problem, ships are allocated to the routes in the network and the cargo allocation over the network is determined in order to determine the profitability of the network. Next, the demand satisfied in the network can be used as a new input in the regional route network design problem. At this point, also more information about the availability of ships is known. For each cluster, the ships allocated to this cluster in the solution of the SACR problem will be available to use in the MIP formulation of the considered cluster. Furthermore, each ship that is not used in the SACR will be initially available in the RRND problem. After solving each MIP, these available ships will be updated: if one of the ships that was not used in the SACR is now allocated to a regional route, it is removed from the available ships, while ships that were initially allocated to a regional route, but are not allocated in the new optimal solution, are added to the available ships. In this way, a new regional route network is found and can be used to resolve the SACR problem. This can be repeated until no improvement is found. Note that in each iteration, the new regional

Algorithm 4: Iterative algorithm

1. Run Algorithm 5 from Section 3.3.1 to obtain hubs and clusters.
2. Design hub network as described in Section 3.3.2.
3. Repeat as long as an improvement is found
   (a) Design regional route network as described in Section 3.3.3.
   (b) Solve ship allocation and cargo routing problem as described in Section 3.3.4 to determine new hub network and satisfied demand realization.

route network will perform at least as good as the previous regional route network, because the previous network is always a feasible solution to the RRND problem. Hence, by using a MIP start for the SACR problem, with the ship allocation to the hub routes as in the previous solution to the SACR problem and the ship allocation to the regional routes as in the new optimal solution of the RRND problem, we are guaranteed to find a solution with a profit that is at least as high as the profit in the previous iteration.

The ship allocation and cargo routing problem is a well-known problem in liner shipping. Formulations used in existing literature to solve this problem, can be distinguished in two groups: flow-based formulations and path-based formulations. In general, path-based formulations outperform the flow-based formulations for small instances. However, the number of variables in path-based formulations will grow exponentially in the input size, while this growth is linear for flow-based formulations. Path-based formulations are often solved using column generation techniques. However, the number of variables in our instances are small enough to be able to generate all variables beforehand. Since it still becomes more difficult to find good solutions for instances with more than five hubs, we propose a new type of formulation: we combine the flow-based and path-based formulations in order to benefit from the advantages of both formulations. Section 3.3.4 will describe both the path-based formulation and this new formulation. In the case study, the performance of these formulations are compared and the best formulation is used in the iterative algorithm.

3.3.1 Hub selection and clustering

The clustering process is done in two parts: first hubs are determined with initial clusters, which are partitioned in smaller clusters in the second part. Algorithm 5 describes the clustering process. The first step uses a variant of the k-centroid clustering algorithm to find the initial clusters and corresponding hubs. These hubs are the potential hubs in our solution algorithm. In the k-centroid clustering algorithm, one starts with k initial ports that are used as hub ports. Then, each port is allocated to the closest hub port using a distance function. After the allocation, the distance function is used to determine the average distance of each port in a cluster to all other ports in this cluster. The port with the smallest average distance is chosen and used as new hub. Then the allocation and hub determination steps are repeated until the clusters are converged. Instead of the distance between two ports, we will use a different distance function in the k-centroid clustering algorithm. Since all cargo, except the cargo to and from a hub, has to be transshipped, hub ports are preferred to have large demand, because this will reduce the transshipment cost for the cluster. Furthermore, the location of the hub in the cluster is important, because all other ports are visited from the hub port. Therefore, we define the following distance function δ^c_{ph} for $p, h \in \mathcal{P}$:

$$\delta^c_{ph} = \delta_{ph} c^{nm} \left(\sum_{p' \in \mathcal{P}} (d_{pp'} + d_{p'p}) + \sum_{p' \in \mathcal{P}} \left(I_{\delta_{pp'} \leq \delta_{hp'}} d_{pp'} + I_{\delta_{p'p} \leq \delta_{p'h}} d_{p'p} \right) \right) + \sum_{od \in \mathcal{D}} d_{od} c^t_h I^t_{odh}. \tag{3.1}$$

In (3.1) $\mathcal{P}$, and $\mathcal{D}$ are the set of ports and OD-pairs respectively, where h denotes the potential hub port in the cluster. δ_{ij} and d_{ij} are respectively the distance in nautical miles and demand in TEU between ports i and j and c^{nm} and c^t_i are respectively the average cost of transporting one container per nautical mile and the transshipment cost of port i, both in USD. Furthermore, I_a returns 1 if statement a is true and 0 otherwise, while I^t_{odh} equals 1 if OD-pair od needs a transshipment at hub h and 0 otherwise. The distance used for the clustering algorithm δ^c includes the cost of transshipping at the hub and the sailing cost from the hub port to the regional port. If the hub port is located further away from the destination (respectively origin) port than the origin (respectively destination) port, the sailing distance between the hub and the regional port is added twice in order to reduce the additional distances travelled by containers because of transshipments at hubs. The average sailing cost per container per nautical mile is estimated by taking an average over all ship types assuming that they will sail at design speed. Finally, we do not allow to sail through the Suez Canal in regional routes, since costs are associated to each passage of the Suez Canal.

Tighter draft restrictions lead to less ship types that are able to berth in that port. Since larger ships usually have larger drafts, tighter draft restrictions prevent larger ships from being able to berth. Hence, ports with high demand in which large container ships are able to berth are most likely not visited on the same route as ports with only little demand or with limited draft restrictions. Hence, we do not want to allocate these ports to the same cluster. Thereto, Step 2 splits the ports in each cluster into large, medium and small ports based on draft limit and total port demand (sum of demand and supply in the data set). The current cluster is replaced by three groups each containing one of these groups of ports. The sets $\mathcal{P}^s$, $\mathcal{P}^m$ and $\mathcal{P}^l$ denote the sets with small, medium and large ports respectively. In the algorithm γ_i denotes the draft of port i and $\underline{\gamma}$, $\bar{\gamma}$, m and M are lower and upper bounds on the draft and port demand. Figure 3.1 shows an example of this step of the clustering algorithm.

A limit P on the total number of ports in a cluster is imposed. In Step 3 of the algorithm, clusters are split into two new clusters as long as the number of ports exceeds this limit. Splitting is based on a 2-centroid clustering algorithm. First, the two ports

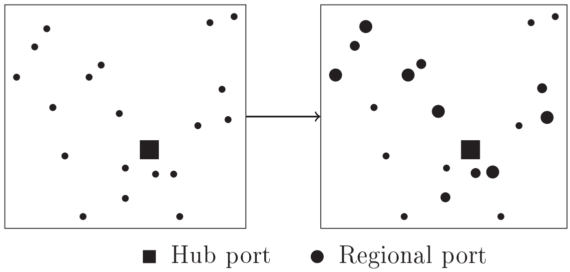


Figure 3.1: Example of Step 2 of Algorithm 5. The left part shows a hub port together with the allocated regional ports. In the right part, the regional ports are divided in small, medium and large regional ports.

that are located most far apart from each other are used to initialize the two cluster centers. Ports are allocated to the closest cluster and the port that is most close to the geographical cluster center is used as new center. Then, clusters are updated using the same procedure, until they do not change any more or a predetermined number of updates have been performed. Figure 3.2 shows an example of Step 3 of the algorithm.

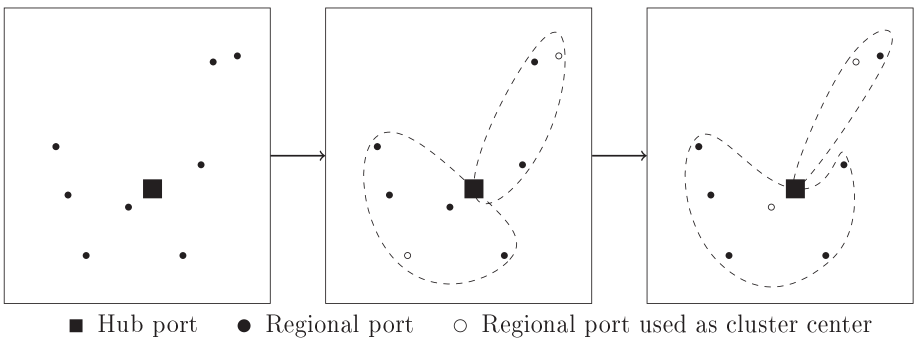


Figure 3.2: Example of Step 3 of Algorithm 5. The left part shows a hub port together with the allocated small, medium or large regional ports. In the middle part, the two regional ports that are located most far apart are used as cluster centers and all other regional ports are allocated to the closest cluster center. The right part shows the updated cluster centers and the new clusters.

Finally, some empty clusters might exist after the first steps of the algorithm, so Step 4 removes these empty clusters if they exist.

Algorithm 5: Clustering algorithm

1. Initialize clusters and determine hubs using a k-centroid clustering algorithm with distance function δ^c as defined in (3.1) and using the k largest ports as initial hubs.

2. For each hub, split the cluster into three new clusters with small, medium and large ports respectively:

   - $\mathcal{P}^s = \left\{ p \in \mathcal{P} : \sum_{p' \in \mathcal{P}} (d_{pp'} + d_{p'p}) \leq \frac{2m}{|\mathcal{P}|} \sum_{od \in \mathcal{D}} d_{od} \vee \gamma_p \leq \underline{\gamma} \right\}$.
   - $\mathcal{P}^m = \left\{ p \in \mathcal{P} : p \notin \mathcal{P}^s \wedge \left(\sum_{p' \in \mathcal{P}} (d_{pp'} + d_{p'p}) \leq \frac{2M}{|\mathcal{P}|} \sum_{od \in \mathcal{D}} d_{od} \vee \gamma_p \leq \bar{\gamma} \right) \right\}$.
   - $\mathcal{P}^l = \{ p \in \mathcal{P} : p \notin \mathcal{P}^s \cup \mathcal{P}^m \}$.

3. As long as $\exists i : | \{ p \in \mathcal{P} : p \in c_i \} | \geq P$, repeat for these clusters:

   - Split the cluster into two new clusters using the 2-centroid clustering algorithm with distance function δ and initial centers the two ports that are located most far apart.

4. Remove all empty clusters if applicable.

3.3.2 Hub network

The hubs are sorted based on geographical location starting in the Far East and ending in Northern Europe. Then, hub routes are generated by complete enumeration of all routes that visit the hubs in geographical order. Hence, routes can visit a hub on the eastbound voyage, on the westbound voyage, on both the eastbound and the westbound voyage or on neither of the two voyages.

3.3.3 Regional network

In this section we will discuss the regional route network design problem. The clusters will be used in the generation of the regional route network. Only ports that belong to the same cluster might be visited on the same route. Hence, the regional route network design problem can be split into a separate problem for each cluster. Furthermore, we assume that all regional ports will be visited on at most one route. Since almost no regional demand is included in the dataset, almost the total supply and demand of a port will be delivered to and from the hub respectively. Hence, the regional network design problem is very similar to a vehicle routing problem with simultaneous pickups and deliveries and a heterogeneous fleet. The maximum number of ports in a cluster will be chosen in such a way that all routes starting and ending at the hub can be generated and included in a mixed integer programming (MIP) model. For each route, the port on the route with the smallest draft restricts which ship types can be used for this route. Hence, the feasible ship types for each route can be calculate beforehand and is input for the MIP model.

We introduce the following notation in order to define the RRND model.

$\mathcal{P}^c$ set of ports in the considered cluster (except the hub port).

$\mathcal{R}$ set of routes.

$\mathcal{R}_p$ set of routes containing port $p \in \mathcal{P}^c$.

$\mathcal{S}^c$ set of available ships for the considered cluster.

$\mathcal{V}_{rs}$ set of speeds that ship $s \in \mathcal{S}^c$ can sail at route $r \in \mathcal{R}$ to obtain a weekly duration.

c^r_{rsv} weekly route cost of sailing route $r \in \mathcal{R}$ per ship of type $s \in \mathcal{S}^c$ with speed $v \in \mathcal{V}_{rs}$.

t_{rsv} duration of route $r \in \mathcal{R}$ if ship type $s \in \mathcal{S}^c$ is used at speed $v \in \mathcal{V}_{rs}$.

$\tilde{c}^t_r$ transshipment cost of route $r \in \mathcal{R}$ for satisfying demand between two ports that are allocated to the same cluster.

n_s number of available ships of type $s \in \mathcal{S}^c$.

ξ_{prsv} fraction of demand of port $p \in \mathcal{P}^c$ satisfied when route $r \in \mathcal{R}_p$ is sailed once a week with ship type $s \in \mathcal{S}^c$ and speed $v \in \mathcal{V}_{rs}$.

y_{rsv} number of weekly port calls on route $r \in \mathcal{R}$ using ship type $s \in \mathcal{S}^c$ with speed $v \in \mathcal{V}_{rs}$.

z_r binary variable indicating whether route $r \in \mathcal{R}$ is used or not

k constant equal to the number of available ships.

Note that t_{rsv} not only denotes the duration of a route, but also the number of ships that need to be allocated to the route in order to obtain a weekly frequency. The mixed integer programming formulation is given by:

$$\min \sum_{r\in\mathcal{R}}\sum_{s\in\mathcal{S}^c}\sum_{v\in\mathcal{V}_{rs}} c^r_{rsv} t_{rsv} y_{rsv} + \sum_{r\in\mathcal{R}} \tilde{c}^t_r z_r \tag{3.2}$$

$$\text{s.t.} \quad \sum_{s\in\mathcal{S}^c}\sum_{v\in\mathcal{V}_{rs}} \frac{1}{k} y_{rsv} \le z_r \qquad r\in\mathcal{R} \tag{3.3}$$

$$\sum_{r\in\mathcal{R}_p}\sum_{s\in\mathcal{S}^c}\sum_{v\in\mathcal{V}_{rs}} \xi_{prsv} y_{rsv} \ge 1 \qquad p\in\mathcal{P}^c \tag{3.4}$$

$$\sum_{r\in\mathcal{R}_p} z_r = 1 \qquad p\in\mathcal{P}^c \tag{3.5}$$

$$\sum_{r\in\mathcal{R}}\sum_{v\in\mathcal{V}_{rs}} t_{rsv} y_{rsv} \le n_s \qquad s\in\mathcal{S}^c \tag{3.6}$$

$$y_{rsv} \in \mathbb{N} \qquad r\in\mathcal{R} \quad s\in\mathcal{S}^c \quad v\in\mathcal{V}_{rs} \tag{3.7}$$

$$z_r \in \{0,1\} \qquad r\in\mathcal{R}. \tag{3.8}$$

The objective is to minimize the total costs associated with the selected routes. Since all demand has to be satisfied, (un)loading costs will be constant and do not need to be included in the total costs. Also most transshipment costs will be incurred in every feasible route network. For example, if the closest hub to the origin port of a demand pair is different from the closest hub to the destination port of this demand pair, the transshipment cost at the hubs will always be incurred. Furthermore, if the hubs are the same, but the origin and destination port are allocated to different clusters, the demand is also always transshipped at the hub, because the route networks are solved per cluster. So, the only transshipments that might or might not be included in the total costs are transshipments of demand pairs with the origin and destination both allocated to the same cluster. Therefore, these are the only transshipment costs included in the objective. In order to include these transshipment cost exactly once per container, the transshipment costs are only included in $\tilde{c}^t_r$ if the origin and destination port of the demand pair are both allocated to the same cluster and the origin port is part of route $r\in\mathcal{R}$, while the destination port is not. Finally, the first part of the objective gives the total route costs of the selected routes. In the route cost c^r_{rsv}, the weekly fixed cost of ship type $s\in\mathcal{S}^c$, the weekly port call costs and the weekly fuel cost (both fuel cost at sea as during port stays) of sailing route $r\in\mathcal{R}$ with ship type $s\in\mathcal{S}^c$ at speed $v\in\mathcal{V}_{rs}$ are included. Hence,

$$c^r_{rsv} = 7c^f_s + \frac{e\tilde{f}_s\left(\frac{v}{\tilde{v}_s}\right)^3 t^s_{rsv} + e\tilde{f}^p_s t^p_r + \sum_{p\in r} c^p_{ps}}{t_{rsv}},$$

where c_s^f is the daily fixed cost of ship type $s \in \mathcal{S}^c$, e is the bunker price in USD per ton, $\tilde{f}_s$ and $\tilde{f}_s^p$ are the fuel consumption in ton per day for ship type $s \in \mathcal{S}^c$ when sailing at design speed and when berthing in a port respectively. Further, $\tilde{v}_s$ is the design speed in knots of ship type $s \in \mathcal{S}^c$ and t_r^p and t_{rsv}^s are respectively the port and sailing times in days of sailing route $r \in \mathcal{R}$ with ship type $s \in \mathcal{S}^c$ at speed $v \in \mathcal{V}_{rs}$. The cost of calling at port $p \in \mathcal{P}^c$ with ship type $s \in \mathcal{S}^c$ is denoted by c_{ps}^p. Finally, routes are generated such that

$$t_{rsv} = \frac{t_r^p + t_{rsv}^s}{7} \in \mathbb{N}.$$

Constraints (3.3) guarantee that z_r will take value 1 if route $r \in \mathcal{R}$ is used by some ship type. Constraints (3.4) ensure that all demand is satisfied, while Constraints (3.5) make sure that each port is visited on exactly one route. The limited availability of ships is imposed by Constraints (3.6). Finally, Constraints (3.7) and (3.8) ensure the integrality and binary conditions on the decision variables.

After solving the MIP model (3.2)-(3.8) for each cluster, the routes that are used in the optimal solutions are added to the regional network.

3.3.4 Ship allocation and cargo routing

The hub network and regional network form together the input network for the ship allocation and cargo routing problem. Note that the regional network already consists of routes visiting each regional port exactly once, such that only the ship allocation to these routes still has to be solved (unless some ports will not be visited at all in the network). The hub network, on the other hand, consists of all possible routes visiting only (some of) the hub ports, so we still need to decide which routes will be used. The ship allocation and cargo routing problem can also be modelled using a MIP formulation. Thereto, we first introduce some additional notation:

$\mathcal{L}$	set of legs.
$\mathcal{D}$	set of origin-destination demand pairs (OD-pairs).
$\mathcal{Q}$	set of paths.
$\mathcal{Q}_l$	set of paths containing leg $l \in \mathcal{L}$.
$\mathcal{Q}_{od}$	set of paths satisfying demand $od \in \mathcal{D}$.
$\mathcal{R}_l$	set of routes containing leg $l \in \mathcal{L}$.
c_{od}^{ls}	cost per TEU of not satisfying demand $od \in \mathcal{D}$.
p_q	profit of transporting one TEU over path $q \in \mathcal{Q}$.
b_s	capacity in TEU of ship type $s \in \mathcal{S}$.
d_{od}	demand of OD pair $od \in \mathcal{D}$.
x_q	amount in TEU transported over path $q \in \mathcal{Q}$.
L_{od}	amount in TEU of unsatisfied demand between $od \in \mathcal{D}$.

Note that a path consists of a certain number of legs, so for example $q_1 = (l_1, l_2, l_3)$ denotes a path containing three legs. Each leg consists of two consecutive ports visited on a route together with the route they are visited on, for example $l_1 = (p_1, p_2, r_1)$ denotes the leg between ports p_1 and p_2 visited consecutively on route r_1. The profit of a path is defined as revenue of satisfying the demand of the OD-pair minus the sum of the loading, unloading and possible transshipment costs. Each OD-pair has one or multiple paths associated to it (the paths that start at the origin and end at the desination port of the OD-pair). The structure of the input network guarantees that the number of paths per OD-pair is limited: the subpath from the origin port to the first hub port (only applicable if the origin port is not a hub) is unique, because every regional port is on exactly one regional route. Similarly, the subpath from the last hub port to the destination (if applicable) is unique. Hence, the number of different OD-paths is equal to the number of different subpaths between the two hubs, which results in a limited number of total paths. This is important because in general the number of paths in a network can grow very fast. Since each path is associated with a variable in the mixed integer programming formulation, this would most likely result in too many variables, such that more advanced techniques, like column generation, are needed to solve the MIP. However, the special structure of our input network guarantees that it is possible to include all variables in the formulation without the necessity of using additional techniques.

The mixed integer programming formulation is given by:

$$\max \sum_{q \in \mathcal{Q}} p_q x_q - \sum_{r \in \mathcal{R}} \sum_{s \in \mathcal{S}} \sum_{v \in \mathcal{V}_{rs}} c^r_{rsv} t_{rsv} y_{rsv} - \sum_{od \in \mathcal{D}} c^{ls}_{od} L_{od} \tag{3.9}$$

$$\sum_{q \in \mathcal{Q}_{od}} x_q + L_{od} = d_{od} \qquad od \in \mathcal{D} \tag{3.10}$$

$$\sum_{q \in \mathcal{Q}_l} x_q \leq \sum_{r \in \mathcal{R}_l} \sum_{s \in \mathcal{S}} \sum_{v \in \mathcal{V}_{rs}} b_s y_{rsv} \qquad l \in \mathcal{L} \tag{3.11}$$

$$\sum_{r \in \mathcal{R}} \sum_{v \in \mathcal{V}_{rs}} t_{rsv} y_{rsv} \leq n_s \qquad s \in \mathcal{S} \tag{3.12}$$

$$x_q \geq 0 \qquad q \in \mathcal{Q} \tag{3.13}$$

$$y_{rsv} \in \mathbb{N} \qquad r \in \mathcal{R} \quad s \in \mathcal{S} \; v \in \mathcal{V}_{rs}. \tag{3.14}$$

The objective of the model is to maximize the total revenue minus the costs. Note that negative objective values in Brouer et al. (2014a) correspond to positive profits and hence, they are actually also maximizing total profit. The revenue is included in the profit of the paths. The total costs consists of (un)loading and transshipment costs (included in the path profit), penalties associated to lost sales and total route costs (fixed ship costs, port call costs and fuel costs during sailing and berthing in ports). Constraints (3.10) ensure that all demand is either satisfied using one of the OD-paths or lost. Further, Constraints (3.11) denote the capacity constraints on each of the legs in the network. The limited availability of ships is modelled in the same way as in the regional route network design problem by Constraints (3.12). Finally, Constraints (3.13) and (3.14) ensure the nonnegativity and integrality of the path flows and number of allocated ships respectively.

Reducing the number of variables

Although the total number of paths is small enough to be included in the MIP formulation, we can improve upon the formulation by using a modelling trick. As already explained above, the start and end subpath of each OD-pair is unique. By introducing an artificial leg between the two hub ports, we can reduce the number of paths to one for each OD-pair. Hence, each path consists of the unique start subpath, the artificial leg between the two hubs and the unique end subpath. This reduces the number of paths and thus the number of variables even further. However, additional constraints are needed to ensure that the flow over the artificial leg is sent over existing legs in the network. Thereto, we introduce:

$\mathcal{L}'$ set of artificial legs added to the network.

$\mathcal{Q}'_{l'}$ set of paths in the network that can be used to transport flow over artificial leg $l' \in \mathcal{L}'$.

Furthermore, the set of paths in the new formulation will now consist of the unique OD-path for each OD-pair and all possible paths between two hubs in the network. Hence, if q is a unique OD-path, then $q \in \mathcal{Q}$ and if q is a path between two possible hubs, then $q \in \mathcal{Q}'_{l'}$ with l' denoting the two hubs that are the start and end ports of the path. The following constraints can now be added to the formulation (3.9)-(3.14):

$$\sum_{q \in \mathcal{Q}_{l'}} x_q = \sum_{q \in \mathcal{Q}'_{l'}} x_q \qquad l' \in \mathcal{L}'. \tag{3.15}$$

Constraints (3.15) ensure that all flow that is sent over an artificial leg is also allocated to one of the existing paths between the first and last port of the artificial leg. Hence, the constraints ensure that all flow is allocated over the real network. The resulting mixed integer program (3.9)-(3.15) has less variables and generates better solutions in shorter computational times. Note that transshipment on hub routes can be handled by adding constraints of type (3.15) twice to the model. The first time, the flow balance between artificial legs in OD-paths and artificial legs in transshipment paths is modelled. The second time, they model the flow balance between the artificial transshipment legs and the real hub legs.

3.4 Case study

In this section, the results of the optimization methods applied to a case study on the Asia-Europe trade lane are discussed.

3.4.1 Data

We use the Asia-Europe data from LINERLIB based on Maersk data (Brouer et al. 2014a, LINERLIB 2014), where also the low, base and high capacity scenarios are introduced. The data contains information on 114 ports on the Asia-Europe trade lane. Furthermore, demand for 4000 OD-pairs is given together with the availability and characteristics of six different ship types. The data contains some unlikely values for the transshipment cost. Some ports have transshipment cost of 0 or almost 0, which results in a large underestimation of the costs of these ports. Therefore, we decided to use the average transshipment costs for these ports, since this will advantage ports with large throughput to be selected as hub ports. Table 3.1 shows the economies of scale in the six different ship types in the data set. The costs per TEU are all calculated under the assumption of full utilization, so scale economies will be less when ships are not fully loaded. The table shows that for larger ships have smaller fixed cost per TEU, lower fuel consumption per TEU both when sailing at design speed and when berthing and have to pay lower fees per TEU capaity for the Suez Canal passage. The only exception is the fuel consumption at design speed of ship type 3 compared to the smaller type 2, but this is cause by the much higher design speed of type 3 (18 knots compared to 14 knots for type 2).

Ship type	Capacity (TEU)	Daily fixed cost (USD/TEU)	Design speed (knots)	Daily fuel usage sailing (ton/TEU)	Daily fuel usage port (ton/TEU)	Suez fee (USD/TEU)
1	900	11.11	12	0.087	0.005	390.60
2	1,600	10.00	14	0.053	0.003	273.06
3	2,400	9.17	18	0.061	0.003	222.68
4	4,800	8.75	16	0.037	0.002	172.31
5	8,400	8.33	16.5	0.030	0.002	150.72
6	15,000	7.33	17	0.025	0.001	138.05

Table 3.1: Economies of scale in ship types

3.4.2 Hubs

We decided to include at most seven different hubs in our network. On one hand, the seven largest ports constitute for more than one third of the total demand, while on the other hand the hub network design subproblem is still managable with seven hubs. When six hubs are included, 453 potential hub routes need to be considered, while this increases to 1818, 7279 and 29124 routes for seven, eight and nine hubs respectively. The potential hubs are determined using the 7-centroid clustering algorithm with distance function as given in (3.1) and $c^{nm} = 0.075$. Recall that c^{nm} denotes the average cost of transporting one container over one nautical mile. We calculated this value using the two types of feeder ships from the data and weighted the costs by the amount of available ships of each type. Note that in general these costs will be overestimated, because of the economies of scale of larger ships that will most probably also be used on the feeder network. This results in the following seven potential hubs with their corresponding region: Bremerhaven (Northern

Europe), Rotterdam (Northern Europe), Algeciras (Southern Europe), Jebel Ali (the Middle East), Tanjung Pelepas (the Singapore region), Shenzhen (Southern China) and Shanghai (the Far East). We will run several experiments including 2-7 of the above potential hub ports as hubs in the network. When including only 2, 3 or 4 hubs, we select respectively from the largest 4, 5 and 6 hubs (in total port demand).

3.4.3 Effect of reducing variables

In this section, we compare the mixed integer formulations (3.9)-(3.14) (SACR model) and (3.9)-(3.15) (reduced SACR model) for the SACR problem. To this purpose, we perform one step of the iterative algorithm for the base case for both formulations and compare the number of variables, best lower and upper bound and optimality gap. Tables 3.2 and 3.3 show the averages of these characteristics for all instances with 3, 4, 5, 6 and 7 hubs for the two formulations. We did not include instances with 2 hubs, since the formulations are the same for these instances. The tables clearly show that

#Hubs	#Variables	LB ($\times 10^6$ USD)	UB ($\times 10^6$ USD)	Gap (%)
3	16,989	21	25	23.08
4	55,204	21	30	48.78
5	219,348	4	35	3,754.44
6	900,238	−77	3,762	4,989.76
7	3,671,793	−77	105,517	137,234.30

Table 3.2: Characteristics for the SACR model

#Hubs	#Variables	LB ($\times 10^6$ USD)	UB ($\times 10^6$ USD)	Gap (%)
3	8,628	22	24	7.46
4	9,436	26	28	6.70
5	13,282	30	32	8.48
6	31,269	32	36	15.77
7	114,093	32	41	26.06

Table 3.3: Characteristics for the reduced SACR model

formulation (3.9)-(3.15) is better suitable for our solution algorithm than the formulation without Constraints (3.15): both the obtained solutions within the time limit and the upper bounds are better using this formulation. Furthermore, the reduced model is able to find feasible solutions for all number of hubs within the time limit, while the original model could not find any feasible solution for instances with six and seven hubs.

3.4.4 Results

Table 3.4 shows the profits of the route networks found using Algorithm 4 for low, base and high capacity scenarios with different potential hubs as input. The table only shows the combinations of potential hubs that resulted in the most profitable networks. The first column of Table 3.4 shows which potential hubs are included while running Algorithm 4: Bremerhaven (Br), Rotterdam (Ro), Algeciras (Al), Jebel Ali (JA), Tanjung Pelepas (TP), Shenzhen (She) and Shanghai (Sha). The second, third and last columns show respectively the profit in million USD of the network found for the low, base and high capacity scenarios. All instances are run with 10 and 12 meters as lower and upper draft bounds respectively. The lower and upper demand bounds are respectively given by 0.25 and 1.5. We allow for at most 6 ports and at most 10 changes per cluster. Finally, the running time of the ship allocation and cargo routing model is set to 180 sec.

	Weekly profit in million USD		
Included hubs	Low	Base	High
Al, JA, TP, She, Sha	19.646	**37.578**	46.176
Ro, Al, JA, TP, She	20.817	36.127	44.656
Br, Al, JA, TP, She	20.559	36.306	45.542
Br, Ro, Al, JA, TP	19.197	35.661	45.663
Ro, Al, JA, TP, She, Sha	20.726	36.453	46.238
Br, Al, JA, TP, She, Sha	20.140	36.772	**46.445**
Br, Ro, Al, JA, TP, She	20.886	36.870	45.886
Br, Ro, Al, JA, TP, She, Sha	**20.981**	36.731	43.765
Current best network	14.951	30.371	37.499

Table 3.4: Weekly profit for the low, base and high capacity scenarios

Table 3.4 shows the networks obtained for different configurations of hub ports. In general, the weekly profit increases when a potential hub port is added. The best network with two hubs for the base instance has a weekly profit of 31 million USD. This increases to over 32, 34 and 37 million USD for networks with three, four and five hubs included. Remarkably, adding the sixth and seventh hub, decreases the profit of the found network. The last row of the table shows the profit of the current best network in literature for the low, base and high capacity scenarios of the Asia-Europe LINERLIB data (Brouer et al. 2014b, Brouer 2015). Note that the current best networks contain some paths in which containers are transshipped twice in the same port. The weekly profits given in Table 3.4 are corrected values for which these double transshipments do not occur anymore and hence are slightly higher than the values reported in Brouer et al. (2014b), Brouer (2015).

Our best networks provide an increase in profit of 40.3%, 23.7% and 23.9% respectively for the low, base and high capacity scenario with respect to the current best networks in literature.

Table 3.5 shows some characteristics of the best route networks, i.e. the networks corresponding with the bold profits in Table 3.4. The first part of the table shows cost characteristics of the networks. All costs and revenues are given in million USD per week. Since vessels are more expensive in lower capacity scenarios, the fleet cost is decreasing when the capacity scenario increases from low to high. Furthermore, the lost sales cost also decreases, because more capacity results in more satisfied demand. The table also shows that the bunker costs are decreasing, while the port costs are increasing when the capacity scenario increases from low to high. Hence, more direct routes are chosen when the vessel costs increases, leading to less port visits. The second part of the table shows some other characteristics of the networks. With less capacity, less ships are available which results in less routes. The hub routes have in general high utilization, resulting in high best peak and average utilization in all three capacity scenarios. Finally, the number of transshipments is relatively high in the network, because many OD-pairs need one or two transshipments in order to be satisfied.

	Low	Base	High
Revenue	130.443	135.012	139.722
Fleet cost	27.944	25.151	22.624
Bunker cost	26.743	18.961	17.858
Move cost	25.008	26.180	26.806
Transshipment cost	8.606	9.274	8.892
Port cost	4.046	4.411	5.118
Canal fees	9.667	9.667	10.494
Lost sales cost	7.449	3.790	1.485
Profit	20.981	37.578	46.445
Fleet deployment	100.00	99.43	92.45
Nr routes used	57	64	71
Average port calls per week	2.04	1.76	1.77
Best peak utilization	100.00	100.00	100.00
Worst peak utilization	26.75	10.50	4.75
Best average utilization	97.83	97.06	99.83
Worst average utilization	14.29	8.13	4.73
Average nr transshipments	1.14	1.29	1.26
Percentage rejections	9.68	4.93	1.93

Table 3.5: Characteristics for the best networks with low, base and high capacity

3.5 Local improvement heuristic: adding a stop to a hub route

In this section, we propose a local improvement heuristic to improve the obtained networks. Thereto, we investigate whether it is profitable to add an additional port to a hub route. In this way we get more insight into the trade-offs. We take one of the hub routes used in the best network of the base case found when including Algeciras, Jebel Ali, Tanjung Pelepas, Shenzhen and Shanghai as hubs and show with some calculations whether it is profitable to also visit another port.

Given the route distance, the sailing time in days and speed in knots can be calculated by respectively:

$$t^s_{rsv} = 7t_{rsv} - t^p_r \tag{3.16}$$

and

$$v = \frac{\delta_r}{24t^s_{rsv}} \tag{3.17}$$

where δ_r is the route distance of route $r \in \mathcal{R}$ in nmi, t_{rsv} is the route time of sailing route $r \in \mathcal{R}$ with ship type $s \in \mathcal{S}$ at speed $v \in \mathcal{V}_{rs}$ in weeks and t^p_r is the total port time of route $r \in \mathcal{R}$ in days. Note that v should be in between a ship-dependent minimum speed v^{min} and maximum speed v^{max}. The bunker cost c^b_{rsv} of route $r \in \mathcal{R}$ sailed with ship type $s \in \mathcal{S}$ at speed $v \in \mathcal{V}_{rs}$ can be computed using the formula:

$$c^b_{rsv} = e\tilde{f}_s \left(\frac{v}{\tilde{v}_s}\right)^3 t^s_{rsv} + e\tilde{f}^p_s t^p_r, \tag{3.18}$$

where e is the bunker price in USD per ton, $\tilde{f}_s$ is the fuel consumption in ton per day at design speed of the vessel type $s \in \mathcal{S}$, $\tilde{v}_s$ is the design speed in nmi/hr of ship type $s \in \mathcal{S}$ and $\tilde{f}^p_s$ is the fuel consumption of the ship $s \in \mathcal{S}$ in ton per day when idling at a port.

Using (3.18) we can calculate the bunker cost for each route and hence the difference in the bunker cost after adding an additional stop. Furthermore, an additional stop results in additional port dues, which can be calculated by:

$$c^p_{ps} = f^p_p + v^p_p b_s, \tag{3.19}$$

where f^p_p is the fixed port call cost at port $p \in \mathcal{P}$ and v^p_p the variable port cost at port $p \in \mathcal{P}$ per TEU capacity of the ship berthing at the port. Finally, transshipment costs decrease by making an additional port call. The transshipment cost made to transship the containers at the nearest hub will be saved for the containers that can now directly be loaded or unloaded at the new port, which results in a savings on transshipment cost s_h^t at hub $h \in \mathcal{H}$ of.

$$s_h^t = N c_h^t, \tag{3.20}$$

where N denotes the number of TEU that can now directly be loaded or unloaded in the new port instead of needing a transshipment at the nearest hub and c_h^t denotes the transshipment cost per container at the nearest hub $h \in \mathcal{H}$.

We consider the hub route Shanghai - Algeciras - Shanghai. 8 post panamax ships with a capacity of $8,400$ TEU are allocated to this route. Each ship needs 8 weeks to complete the route once, so that every week a port call is realized on this route. Since only two ports are called on the route, the total port time of the route equals two days. We will investigate whether it is profitable to make an additional stop in Rotterdam and Hong Kong, which are also ports with a relatively large demand. We assume that the port rotation time remains the same and accommodate the extra stop by changing the ship speed. Through this choice we make the comparison insightful, though also other options exist. Yet these may require a complete network change.

The current route distance is equal to $18,280$ nautical miles (nmi). Using (3.17), we can now determine the sailing speed on the shuttle route. The sailing speed is equal to 14.1 nmi/hr. The design speed of a super panamax vessel is 16.5 nmi/hr, the bunker price is assumed to be 600 USD/ton, the fuel consumption at design speed is 82.2 ton/day and the fuel consumption when idling at a port equals 7.4 ton/day. The bunker costs can then be found using (3.18) and equal $1,672,594$ USD.

The next sections describe the effect of additional port calls at Rotterdam and Hong Kong. An overview of the cost differences for these two ports can be found in Table 3.6.

3.5.1 Additional stop in Rotterdam

The route distance increases with 15.1% to $21,048$ nmi when Rotterdam is added to the route. Furthermore, an additional port call results in an increase in port time from two to three days. The new speed that has to be sailed to complete the route in nine weeks can be determined using (3.16) and (3.17) and is equal to 16.5 nmi/hr. Using (3.18), we find that the new bunker cost are equal to $2,649,762$ USD. Hence, making an additional port call at Rotterdam will lead to an increase in bunker costs of $977,169$ USD. Furthermore, additional port costs are incurred by the additional stop in Rotterdam. The fixed port cost of Rotterdam is equal to $19,187$ USD and the variable cost is 8 USD/TEU. The capacity of the ships is given above and is equal to $8,400$ TEU, which results in an additional port cost of $86,387$ USD by (3.19). The savings made by adding Rotterdam to the hub route depend on the number of TEU to be directly (un)loaded at Rotterdam instead of Algeciras. Containers originating at or destined for Rotterdam needed to be transshipped at the port of Algeciras in the old scenario. The transshipment cost at Algeciras is 68 USD/TEU. In total, $5,040$ TEU, which originate from or destine for Rotterdam, can be shipped by the considered hub service per week. The total savings can be found using (3.20) and are equal to $342,720$ USD. Hence, adding Rotterdam to the hub route will result in additional costs of $720,836$ USD and will thus not be profitable.

3.5.2 Additional stop in Hong Kong

Next, we will perform the analysis for the port of Hong Kong. Adding Hong Kong after Shanghai to the original hub route will result in a smaller detour than when adding Rotterdam. The total route distance will increase with 0.35% to $18,344$ nmi. Using (3.16) and (3.17) we then find a new speed of 14.4 nmi/h. The bunker route costs can again be calculated using (3.18) and increase with an amount of $86,018$ to $1,758,611$ USD. The fixed port cost of Hong Kong is equal to $6,809$ USD and the variable cost is 1 USD/TEU. Hence, we find additional port costs of $15,209$ USD using (3.19). In total $8,102$ TEU originating from or destined for Hong Kong are transported on the hub route per week. The transshipment cost of Shanghai is equal to 31 USD/TEU. Hence, by (3.20) a total saving in transshipment cost of $251,162$ is found. In total, adding Hong Kong to the hub route will result in cost savings of $149,935$ USD. Hence, adding Hong Kong to the hub route will be profitable.

	Rotterdam	Hong Kong
Difference in bunker cost	977,169	86,018
Difference in port call cost	86,387	15,209
Difference in transshipment cost	–342,720	–251,162
Total cost of additional stop	720,836	–149,935

Table 3.6: Cost differences in USD for additional stops in Rotterdam and Hong Kong

3.5.3 Conclusion

The above proposed procedure can be used as a local improvement heuristic to improve the networks found in Section 3.4. However, adding regional ports to hub routes destroys the nice structure we are exploiting in Section 3.3. In this section, we use the property that regional ports are only visited on one regional route to limit the number of paths needed in the mixed integer program to formulate the SACR problem. When we will start adding regional ports to hub routes, the number of paths will increase, especially when multiple regional ports are visited on hub routes. Hence, the complexity of the SACR model will increase, making it even harder to find good solutions. We have also seen in this section that it is not evident that adding a regional port to a hub route will result in a cost reduction. Furthermore, if we are able to find cost reductions, the improvements are only marginal. Therefore, we decide not to include the improvement heuristic in our solution algorithm.

3.6 Conclusion

Container ship sizes have increased during the last few years. Bigger ships might incur higher costs in ports; hence fewer port visits might increase the route efficiency. The goal of this research has been to investigate the profitability of a hub-and-feeder network with only a few port calls per route.

The influence of a hub-and-feeder network on the demand structure, transit time, (dis)economies of scale, port time uncertainty, delay management and flexibility and competition is discussed. Hub ports will in general attract more demand, because they offer fast and frequent services with other hub ports. Hence, a demand shift towards hub ports is to be expected when introducing hub-and-feeder services. Hence, the hub networks might even be more profitable than already indicated by the results in this paper. Hub-and-feeder networks will in general result in more transshipments, which can result in longer transit times. However, in hub-and-feeder systems, hub routes are relatively fast routes and are sailed frequently, which will probably balance the increase in time caused by the increase in transshipments. Furthermore, hub services benefit more from the economies of scale at sea than traditional liner routes, because the fraction of sailing time with respect to route duration is higher for hub services. Moreover, hub services will most likely justify the use of even larger container ships. The effect on port time uncertainty is more difficult to estimate, because on one hand on-time arrivals can better be managed for hub services, which significantly reduces the probability of incurring delays in ports. On the other hand, larger uncertainty in call size endangers the port time reliability. However, these possible delays are easier and most probably cheaper to manage for hub services, because ships have to cover large distances at each sea leg of a hub service. The increase in sailing speed needed to capture a certain amount of time is then smaller compared to shorter sea legs. Although all these studies have their limitations, they do give an indication that hub-and-feeder networks are an interesting and efficient concept, like they are in airlines.

An iterative solution approach is proposed to solve the problem. In the iterative approach, first the clusters corresponding to each hub are determined. Thereafter, all possible hub routes are generated. The initial regional route network can be generated under the assumption that all demand will be satisfied. Given a fixed regional route network, the ship allocation and cargo routing problem can be formulated and solved. However, since the problem has to be solved multiple times, we do not necessarily solve the problem to optimality, but impose a time limit. The solution to this problem results in a new realization of the satisfied demand, which can be used to reoptimize the regional route network. The iterative solution approach repeats these steps until no improvement is found.

A new formulation is proposed to solve the ship allocation and cargo routing problem. This formulation basically combines ideas from the existing flow-based and path-based formulations. We compare our new formulation with a typical path-based formulation and conclude that the new formulation clearly outperforms the path-based formulation: both the found solution and the best bound obtained after a fixed time are better for the new formulation.

A case study is performed using the data benchmark set as introduced in Brouer et al. (2014a). In the case study, the profitability of route networks using combinations of seven potential hub ports (Bremerhaven, Rotterdam, Algeciras, Jebel Ali, Tanjung Pelepas, Shenzhen and Shanghai) is investigated using the iterative algorithm. Three scenarios with low, base and high capacity are considered. The results show that including all seven hubs results in the best network for the low capacity case, while for the high capacity case all hubs except for Rotterdam and in the base case all hubs except for Rotterdam and Bremerhaven should be included to obtain the best networks. Finally, the results show that our networks perform better than the reference network as discussed in Brouer et al. (2014b) and Brouer (2015) with profit increases of respectively 40.3%, 23.7% and 23.9% for the low, base and high capacity scenarios.

Next, an improvement heuristic is proposed in which the profitability of an additional port call at a hub service is investigated. In the example, the hub service between Shanghai and Algeciras is considered and the costs and savings of adding Rotterdam and Hong Kong respectively to the route are calculated. It is concluded that adding Hong Kong to the hub service is beneficial, while adding Rotterdam is not profitable. The profitability mainly depends on the additional distance that has to be covered, the number of TEU for which the number of transshipments can be reduced and the transshipment cost of the closest hub. However, the changes in profit are very small. Furthermore, adding ports to hub services will destroy the structure of the MIP models. Therefore, the improvement heuristic is not added to the solution algorithm.

Chapter 4

Designing robust liner shipping schedules Optimizing recovery actions and buffer times[1]

4.1 Introduction

In liner shipping networks ships follow a fixed round tour with a fixed time schedule. However, ships can encounter delays both when they are sailing between ports and when they are berthing in a port. Shipping companies buy certain time slots in terminals in ports for a specified number of berths, which terminals use to make their own planning regarding time and berths. Hence, delayed ships have consequences for both ships and ports. For ships, the consequences depend on the amount of delay and on the number of required berths.

When a ship arrives delayed in a port, the terminal can encounter demurrage costs. Demurrage costs are charged to compensate for several aspects. These costs can for example be incurred because the berth schedule has to be adapted. The containers that have to be loaded to the ship are placed near the scheduled berth in the initial berth planning. If the delayed arrival results in a berth change, these containers are most probably located further away from the new allocated berth, hence higher transportation costs are incurred. The terminal can charge the shipping company for these costs, because they are

[1] An earlier version of this chapter is published online as report of the Econometric Institute Report Series: Mulder, J., Dekker, R. and Sharifyazdi, M. Designing robust liner shipping schedules: Optimizing recovery actions and buffer times, report number: EI 2012-30.

caused by the late arrival. Furthermore, discharged containers have to be transported to their destination. A delayed arrival can result in missed connections and can also affect the berthing times of other ships, which also results in additional costs. Finally, ships can encounter costs that represents liquidated damages for delays. These costs occur when the ship is prevented from (un)loading the containers within the scheduled time.

Delays can have many causes, including terminal operations delay, port access delay, maritime passage delay and chance (Notteboom 2006). As explained above, a shipping company will encounter additional costs because of delays. Besides, delayed ships are also more likely to have a delayed arrival in the next port of call. Furthermore, the incurred delay can increase during the round tour, because new delays are incurred and because delayed ships might have to wait until they can enter a port. Delayed ships can reduce their delay by taking recovery actions, such as increasing the sailing speed or port handling capacity, against certain costs. Furthermore, a buffer time can be incorporated in the schedule to capture (a part of) the delay. The total buffer time of a route depends on the time needed to perform one round tour of the route. Since liner shipping routes are usually serviced once a week, the round tour time is rounded up to an integer number of weeks. The total buffer time is chosen in such a way that the time of a single round tour including buffer times equals an integer number of weeks, implying calls at specific weekdays.

The goal of the problem studied in this paper is to determine a recovery policy and buffer time allocation that minimizes the costs associated with delays and recovery actions for a given liner shipping route.

4.1.1 Literature review

Notteboom (2006) analyses the effects of delays on liner shipping networks. The author provides insights in the causes of delay and mentions some actions that can be taken to recover from the obtained delays. One of the proposed recovery actions is to increase the sailing speed on the sea leg. Notteboom and Vernimmen (2009) discuss the effect of high fuel costs on liner shipping networks. They conclude that high bunker prices have a significant influence on the route design in liner shipping networks. For example, liner shipping companies decided to increase their fleet size in order to be able to reduce the sailing speed on the route network. This also results in more buffer times in ports, however the authors do not study the influence of the allocation of the buffer times over the ports.

Lately, research on minimizing fuel costs and emissions by decreasing the sailing speed has become more popular (see for example Corbett et al. 2009, Fagerholt et al. 2010, Cariou 2011). From these studies it follows that the sailing speed of ships can have significant effect on fuel costs and emissions. However, these studies do not consider situations in which ships can encounter delays.

Wang and Meng (2012a) and Wang and Meng (2012b) consider the problem of designing liner shipping schedules when there is uncertainty in the port and sailing times. In their models, they try to adjust the speed in order to capture delays. A trade-off is made between the round tour duration (number of ships needed) and fuel cost associated with the required speed to maintain these round tours. Qi and Song (2012) propose a model to minimize the fuel emissions when port times are uncertain. Again, delays are captured by adjusting the sailing speed on the next sea leg. The authors compare the on-time arrivals under different conditions. Furthermore, the authors provide some managerial insights regarding the uncertain port times. Only in Wang and Meng (2012b), the effect of the buffer time allocation on the uncertainty and fuel costs is considered. However, their method can only handle linear recovery rules.

Brouer et al. (2013) consider a problem with disruption scenarios. They introduce a time horizon after which the vessel has to be back on schedule. Ships can adjust their sailing speed, omit port calls or swap the order of port calls in order to return to the planned time schedule. Since they consider a real-time problem, short computational times are important. The authors show that real life problems can be solved fast using the proposed mixed integer programming model.

Finally the problem of finding an optimal allocation of buffer times also obtained attention in the railway sector, where it was framed as running time supplements. Kroon et al. (2007) and Kroon et al. (2008) use a stochastic optimization model to solve the problem of optimally allocating running time supplements in a railway network. Kroon et al. (2007) consider the problem of a single train for which they found that in an optimal allocation supplements are mainly allocated to the middle part of the trip and not much to the first and last parts of the trip. Kroon et al. (2008) extended this research for multiple trains and a cyclic timetable. Both studies show that the punctuality (percentage of trains that arrive within 3 minutes of the schedule time) can be increased by reallocating the time supplements. Since trains usually already run at maximum speed and dwell times in a station are needed to let passengers enter and leave the train, it is not possible to include recovery actions in a railway network.

4.1.2 Contribution and outline

Very little research has been performed on developing robust routes in liner shipping or other systems with round tours. In current literature, delays can only be reduced by increasing the sailing speed on the next sea leg. Furthermore, it is often assumed that the sailing speed can be adjusted in such a way that the total amount of delay can be captured on this single leg. However, since very high costs are incurred when the sailing speed has to be increased to the maximum, it is probably cheaper to use multiple sea legs to capture large delays (especially when the next sea leg is relatively small). Furthermore, the effect of the buffer time allocation on delays has not been studied yet.

In this article, we propose a model that can be used to find the optimal recovery policy and buffer time allocation at the same time. To the best of our knowledge, such a formulation does not exist in the literature yet. The proposed formulation can be used for a wide variety of problem specific issues. For example, the recovery policy can include both actions in ports and actions between ports. Furthermore, it is also possible to include actions that will take more time, such as decreasing the sailing speed, in order to minimize the cost of the liner shipping company. Delay can be penalized by including delay costs, but it is also possible to add chance constraints on the delay in the model or a combination of both. The distribution of delay that will be additionally incurred, might depend on the current port position and amount of delay. In the standard formulation a maximum on the amount of buffer that can be allocated to ports on the route is imposed. However, when the arrival times in certain ports on the route cannot be changed, it is also possible to add a maximum amount of buffer that can be allocated to parts of the route in order to satisfy the required arrival times in those ports. Finally, by extending the state space by adding so-called way points to the route, more flexibility in the recovery policy can be obtained. In this way, the sailing speed can be adjusted during a sea leg based on the current actual delay with respect to the schedule.

The remainder of this paper is organized as follows. In Section 4.2 the problem is described in more detail. First, in Section 4.2.1 the so-called ship delay recovery problem is introduced, after which this problem is extended to include the buffer time allocation in Section 4.2.2. Section 4.2.3 describes two extensions to the standard formulation as introduced in Section 4.2.2. Next, in Section 4.3 two structural properties of the problem are given and proved. Section 4.4 proposes four heuristics to solve the problem and Section 4.5 describes results from a case study. Finally, in Section 4.6 the main conclusions from this research are drawn.

4.2 Problem description

4.2.1 Ship delay recovery problem

As mentioned before, the goal of this paper is to derive methods to optimize recovery actions and the buffer time allocation simultaneously. We will start with a simplified version of the problem in which we only optimize the recovery actions. In this so-called ship delay recovery problem, the ship schedule and vessels operating the schedule are known beforehand. Hence, the buffer allocation is fixed and we know the planned arrival and departure times for each port on the route. During the execution of the route, ships might encounter delays. Delays will propagate through the network if no action is taken. This will lead to high costs and is thus undesirable. Therefore, recovery actions should be taken to recover from the incurred delay. The ship delay recovery problem tries to find the recovery policy that leads to lowest total expected costs. Costs consist of delay dependent costs and action dependent costs.

Markov decision process

The ship delay recovery problem will be formulated as a Markov decision process. A recovery policy describes for every possible state of the Markov process which action has to be performed given that the process is in that specific state. The states of the Markov process denote the position of the ship and the amount of delay with respect to the original schedule encountered by the ship. The position of the ship contains both the port in which the ship is berthing and whether it is arriving or leaving the port. Furthermore, since a finite number of possible states are needed in the Markov process to allow efficient computations, we discretize delay and fix a maximum amount of delay, d^{max}, that can be encountered by a ship.

Markov property

The Markov property states that the transition probabilities of moving from one state to another are independent on the past states of the process. In other words, in order to satisfy the Markov property, the additional delay incurred in a port or between two ports depends only on the current port position and the current amount of delay with respect to the schedule. In this section, we will validate this assumption. We will do this using the pie chart on sources of schedule unreliability as given in Notteboom (2006). From this chart, we can conclude that most delays are caused by port/terminal congestion (65.5%). This category includes delays caused by unexpected waiting times before berthing or before loading/discharging. These delays can of course depend on the current delay of the ship (when a ship arrives with a high delay in a port, the probability that in the mean time another ship arrived increases). However, it is not relevant in which states the current delay before the transition is obtained. To satisfy the Markov property, additional delays are allowed to depend on the current delay, so delays in this category will probably be captured under the Markov property. The second category of delay is port/terminal productivity below expectations (20.6%). Again, there is no reason to assume that the port/terminal productivity will be lower or higher when ships incurred more delay during their last sea leg, so also this category will probably not violate the Markov property. In fact, only the third category (unexpected waiting times due to weather or on route mechanical problems, 5.3%) can presumably violate the Markov property. For on route mechanical problems a similar reasoning as for the previous two categories holds, but weather conditions can sometimes depend on the incurred delay on the previous sea leg. In case of prolonged bad weather, one could argue that the incurred delay will be correlated with the previously incurred delay. However, we do not see this as a problem for the Markov assumption because of two reasons. First, delay caused by weather conditions is only a very minor part of the total delay (less than 5.3%). Second, in the proposed Markov process, there is no fixed time interval between two consecutive states. In general, berthing in a port will take at least one day, while sailing between ports will take a few days or even weeks. The weather conditions may change in the mean time and the ship also covers a large distance in that time period. These reasons combined make us believe that correlated additional delays are such a minor part of total delay, that the Markov assumption is a valid one.

Furthermore, we want to stress that few alternative ways exist to model this stochastic problem. Most other papers on similar problems use a two-stage stochastic programming approach in which it is also assumed that delays are independent of each other. Finally, we note that dependencies in delays because of weather conditions might also be captured by adding an additional dimension to the state space, denoting the weather conditions. In this way, delays are allowed to depend on this dimension, so the probability of incurring a certain delay may differ for good and bad weather conditions. However, the state space of the Markov process will significantly increase using this formulation, which may result in longer computational times.

Formulation

Above we defined the states of the process and we validated the Markov property for the state definition. Furthermore, in each state of the Markov process, a decision is made about which recovery action to take in that state. This process is referred to as a Markov decision process. We introduce the following sets:

$\mathcal{P}$	set of possible port positions (port name and arriving/leaving).
$\mathcal{D}$	set of possible units of delay.
$\mathcal{I}$	set of possible states of the Markov process, $\mathcal{I} = \mathcal{P} \times \mathcal{D}$.
$\mathcal{K}$	set of possible actions.
$\mathcal{K}(i)$	set of possible actions that can be performed when in state i.

A Markov decision process can be defined as a discrete time Markov chain in which after each transition an action $k \in \mathcal{K}(i)$ has to be chosen from a set of available actions (Hillier and Lieberman 2001). A cost C_{ik} is associated with performing recovery action $k \in \mathcal{K}(i)$ in state $i \in \mathcal{I}$. As already mentioned, the probability of a transition from state $i \in \mathcal{I}$ to state $j \in \mathcal{I}$ only depends on the current state i and the recovery action $k \in \mathcal{K}(i)$ chosen in state i. These transition probabilities are denoted by p_{ijk}.

As discussed before, the maximum allowable delay is bounded in order to obtain a finite number of states. This means that all ships that exceed the maximum allowable delay are assumed to have the maximum allowable delay instead. Since this might lead to undesirable side effects, we impose a large penalty per unit of delay that is disregarded. This can be done by introducing some additional states with delay values exceeding the maximum allowable amount of delay and including the penalty cost in these states. We can assume that a ship will sail at maximum speed if the delay in the current state is high enough to have a positive probability to arrive with more than the maximum allowable delay in the next state under the maximum speed action, since it would incur the large penalty with a larger probability if it will sail slower. Hence, we will use some states with more delay than the maximum allowable amount of delay, but the amount of additional states needed is limited, since the maximum delay to be incurred is bounded.

Transition probabilities

Next, the probability of a transition from one state to another can be determined for all combinations of states. The Markov property implies that additional incurred delays only depend on the current port position and delay with respect to the schedule. Furthermore, in each state of the Markov process a decision is made about which recovery action to take in that state. Thus, the transition probabilities also depend on the chosen recovery actions. Let $\bar{p}_{i\bar{d}}$ be the probability that $\bar{d}$ time units of additional delay are incurred when $i = (p, d)$ is the current state of the Markov process. Furthermore, let g_k be the gain (in time units) of recovery action k and B_p be the amount of buffer (in time units) of port position p. A transition from $i = (p, d)$ to $j = (p', d')$ occurs if port position p' is the direct successor of port position p in the model and $d' = d + \bar{d} - g_k - B_p$. Thus, the transition probability p_{ijk} of going from state $i = (p, d)$ to state $j = (p', d')$, where p' is the direct successor of p, when recovery action k is chosen, will equal:

$$p_{ijk} = \begin{cases} \bar{p}_{i\bar{d}} & \text{with } \bar{d} = d' - \min\{d, d^{max}\} + g_k + B_{p'} \quad \text{if } d' > 0 \\ \sum\limits_{\bar{d} \in D} \bar{p}_{i\bar{d}} & \text{with } D = \{\bar{d} | d + \bar{d} - g_k - B_{p'} \leq 0\} \quad \text{if } d' = 0. \end{cases}$$

Note that d^{max} is the maximum allowable delay, hence $\min\{d, d^{max}\}$ ensures that the delay above the maximum allowable delay is disregarded.

Performance measure

Usually, the (long-run) expected average cost per unit time is used as a performance measure in Markov decision processes. The underlying assumption in calculating the expected average cost per time unit is that a transition between states takes always place after a fixed time interval. However, in this model the order in which the ports are visited is fixed, while the time needed in a state varies over the states. Since costs are only incurred once in each state, the (long-run) average cost per port position can be computed as

$$\mathbb{E}[C] = \sum_{i \in \mathcal{I}, k \in \mathcal{K}(i)} \pi_i C_{ik} D_{ik},$$

where D_{ik} denotes the current policy; $D_{ik} = 1$ if action k is chosen in state i and 0 otherwise, and π_i represents the steady-state probability of being in state i under the evaluated policy.

The next section describes a linear programming formulation that can be used to solve a Markov decision process. The linear programming formulation is obtained from Hillier and Lieberman (2001). They also describe a few iterative algorithms that can be used to solve a Markov decision process, such as policy improvement.

Linear programming formulation

The Markov decision process can be formulated as a linear programming model if randomized policies are allowed. In a randomized policy the D_{ik} variables are allowed to be continuous variables instead of integer variables. The definition of D_{ik} then becomes

$$D_{ik} = \mathbb{P}(\text{decision} = k | \text{state} = i).$$

In other words, D_{ik} is the probability that action k is selected given that the process is in state i. The decision variables of the linear programming model are

$$\pi_{ik} = \mathbb{P}(\text{state} = i \text{ and action} = k).$$

Thus, π_{ik} is the steady-state unconditional probability that the process is in state i *and* decision k is made (Hillier and Lieberman 2001). Note that π_{ik} and D_{ik} are closely related to each other. It follows that

$$\mathbb{P}(\text{state} = i \text{ and action} = k) = \mathbb{P}(\text{action} = k|\text{state} = i)\mathbb{P}(\text{state} = i),$$

or

$$\pi_{ik} = D_{ik}\pi_i,$$

with π_i the steady-state probability that the Markov chain is in state i:

$$\pi_i = \sum_{k\in\mathcal{K}(i)} \pi_{ik}.$$

From the previous formulas it can be concluded that

$$D_{ik} = \frac{\pi_{ik}}{\pi_i} = \frac{\pi_{ik}}{\sum_{k'\in\mathcal{K}(i)} \pi_{ik'}}. \tag{4.1}$$

The linear programming model can now be formulated as:

$$\min \sum_{i\in\mathcal{I}} \sum_{k\in\mathcal{K}(i)} C_{ik}\pi_{ik} \tag{4.2}$$

$$\text{s.t.} \quad \sum_{i\in\mathcal{I}} \sum_{k\in\mathcal{K}(i)} \pi_{ik} = 1. \tag{4.3}$$

$$\sum_{k\in\mathcal{K}(j)} \pi_{jk} - \sum_{i\in\mathcal{I}} \sum_{k\in\mathcal{K}(i)} \pi_{ik}p_{ijk} = 0 \qquad j \in \mathcal{I}. \tag{4.4}$$

$$\pi_{ik} \geq 0 \qquad i \in \mathcal{I}, \quad k \in \mathcal{K}(i). \tag{4.5}$$

The objective of the model (4.2) is to minimize the average expected cost. Constraint (4.3) ensures that the probabilities of being in a state sum up to 1 and Constraints (4.4) are the transition constraints ensuring that the process moves from one state to another. Finally, Constraints (4.5) make sure that all π_{ik} take nonnegative values.

After solving the linear programming model, the values of D_{ik} can be obtained using (4.1). The optimal solution of the model will contain $|\mathcal{I}|$ basic variables $\pi_{ik} > 0$. Since it can be shown that $\pi_{ik} > 0$ for at least one $k \in \mathcal{K}(i)$ for each $i \in \mathcal{I}$, it follows that $\pi_{ik} > 0$ for exactly one $k \in \mathcal{K}(i)$ for each $i \in \mathcal{I}$. Therefore, each D_{ik} has an integer value and the optimal policy is deterministic instead of randomized (Hillier and Lieberman 2001).

4.2.2 Ship delay recovery problem with buffer time allocation

Next, we consider the ship delay recovery problem in which we also need to allocate the buffer times. In this problem, the ship route and vessels operating the route are known beforehand: we only know which ports are visited and in which order they are visited by the ships. Based on this information, we can determine the minimum time needed between an arrival and departure in the same port and between a departure in one port and an arrival in the next port. However, since liner shipping routes are usually serviced once a week, the round tour time is rounded to an integer number of weeks. This means that there is probably some time left when we only use the minimum times needed in the schedule. We will call this additional time to guarantee a round tour time of an integer number of weeks the total amount of buffer time in the model.

In the ship delay recovery problem with buffer time allocation, we also still want to determine an optimal recovery policy. But now an additional decision regarding the buffer time is included in the problem. We want to determine for each port position which amount of buffer can best be included in the time schedule to minimize the costs associated with delays and recovery actions. Note that contrary to the recovery policy, the buffer time should be determined beforehand and should be independent on the amount of delay incurred in a port position. Hence, buffer times can also be modeled as an action in a Markov decision process. However, we lose the property that actions in different states (that is, different amount of delay) can be taken independent of each other. Although policy improvement can still be applied, it does not necessarily result in improvements in each step and hence it does not necessarily result in the optimal policy, which is experimentally confirmed because we found cases in which cycling appeared. In general it is unknown how to address such a problem in Markov decision chains. We were able to formulate a mixed integer programming model by adding additional (integer) constraints to the linear programming formulation of the Markov decision process.

Mixed integer formulation

First, we introduce the following sets needed to formulate the new model:

$\mathcal{B}$	set of possible values of buffer time per ship position.
$\mathcal{A}$	set of possible actions in the new Markov decision problem, $\mathcal{A} = \mathcal{K} \times \mathcal{B}$.
$\mathcal{A}(i)$	set of possible actions in the new Markov decision problem that can be performed when in state i, $\mathcal{A}(i) = \mathcal{K}(i) \times \mathcal{B}$.

Furthermore, let M be the amount of buffer time available in the route and let B_b denote the value in time units of buffer $b \in \mathcal{B}$.

The ship delay recovery problem with buffer time allocation can now be formulated as:

$$\min \sum_{i\in\mathcal{I}} \sum_{a\in\mathcal{A}(i)} C_{ia}\pi_{ia} \tag{4.6}$$

$$\text{s.t.} \quad \sum_{i\in\mathcal{I}} \sum_{a\in\mathcal{A}(i)} \pi_{ia} = 1. \tag{4.7}$$

$$\sum_{a\in\mathcal{A}(j)} \pi_{ja} - \sum_{i\in\mathcal{I}} \sum_{a\in\mathcal{A}(i)} \pi_{ia}p_{ija} = 0 \qquad j \in \mathcal{I}. \tag{4.8}$$

$$\sum_{p\in\mathcal{P}} \sum_{b\in\mathcal{B}} B_b y_{pb} = M. \tag{4.9}$$

$$\sum_{b\in\mathcal{B}} y_{pb} = 1 \qquad p \in \mathcal{P}. \tag{4.10}$$

$$\sum_{d\in\mathcal{D}} \sum_{k\in\mathcal{K}(pd)} \pi_{(pd),(kb)} \leq y_{pb} \qquad p \in \mathcal{P}, \quad b \in \mathcal{B}. \tag{4.11}$$

$$\pi_{ia} \geq 0 \qquad i \in \mathcal{I}, \quad k \in \mathcal{K}(i). \tag{4.12}$$

$$y_{pb} \in \{0, 1\} \qquad p \in \mathcal{P}, \quad b \in \mathcal{B}. \tag{4.13}$$

Objective (4.6) of the model is still to minimize the average expected cost and Constraints (4.7) and (4.8) are also still the same as in the ship delay recovery problem. However, using only these constraints will result in multiple buffer times for a ship position, since for each value of delay another buffer time can be chosen. Therefore, additional constraints have to be added that ensure that the same buffer time is chosen for all different values of delay for each ship position. Thereto, the binary decision variables y_{pb} are introduced, which take the value 1 if buffer time $b \in \mathcal{B}$ is allocated to ship position $p \in \mathcal{P}$ and 0 otherwise. Constraint (4.9) makes sure that the total allocated buffer time equals the maximum amount of buffer time available (M). Furthermore, exactly one possible buffer time has to be allocated to each ship position, which is ensured by Constraints (4.10). Finally, Constraints (4.11) state that the steady-state probabilities π_{ia} can only be positive for combinations of ship position and buffer time for which $y_{pb} = 1$. The right-hand side of these constraints can be strengthened to $y_{pb}/|\mathcal{P}|$ if ports are always visited in the same order (i.e. actions that skip ports are not considered), since ships will then be with equal probability in each of the ports (in which case the constraint will even hold with equality).

Note that the above mixed integer programming model simplifies to the linear programming model discussed in the previous section when the buffer time allocation is fixed. For a feasible fixed buffer time allocation, Constraints (4.9) and (4.10) always hold. Furthermore, only states for which $y_{pb} = 1$ can be visited in this case, which ensures that also Constraints (4.11) are automatically satisfied when a feasible fixed buffer time allocation is given.

In the mixed integer programming (MIP) model all feasible buffer time allocations can be formed using the y_{pb} variables. Since for a fixed y_{pb} the mixed integer programming model simplifies to the linear programming model discussed in the previous section, the mixed integer programming model minimizes the costs over all feasible buffer time allocations. Thus, the mixed integer programming model will indeed return the buffer time allocation that results into the lowest cost of delay and recovery actions. Furthermore, the optimal recovery policy for this buffer time allocation is returned by the model.

The linear programming relaxation of the above MIP provides in general a very weak bound. Constraints (4.11) can easily be made feasible by letting y_{pb} be the sum of the probabilities in which a buffer of b units is allocated to port position p. Constraints (4.10) only ensures that the sum of the allocated buffer variables corresponding to each port is equal to one, which can still be made feasible in this case. However, this allows for much more freedom in the choice of buffer than in the MIP, resulting in a weak bound.

4.2.3 Extensions

In this section, we will discuss two extensions of the standard model described above. In the first extension a chance constraint on the delay will replace the delay cost in the model. This can for example be used to minimize the emissions of ships without incurring too much delay. The second extension is to fix the arrival times in some of the ports that are visited on the route. This extension can for example be used to fix the arrival times at the Suez Canal in order to join a convoy.

Chance constraint on delay

In the standard formulation of the model, (4.6)-(4.13), the objective is to minimize the total costs related to delays and recovery actions. However, costs of delays are not always very clearly defined or liner companies may have other objectives, such as minimizing total emissions. Removing the delay costs from the objective function will obviously result in high delays, which is undesirable. Adding one or more chance constraint(s) to the formulation will ensure that the delay will remain within reasonable limits. Recall that the steady-state variables π_{ia} denote the unconditional probability that the process is in state i and decision a is made. Thus, for example,

$$\sum_{p,a} \pi_{(p0),a}$$

denotes the unconditional probability that the process is in a state without any delay. By adding the constraint

$$\sum_{p,a} \pi_{(p0),a} \geq 1 - \alpha$$

to the model for a suitable value of α with $0 \leq \alpha \leq 1$, we can restrict the probability of being in a state with delay to $100\alpha\%$. Similarly, the constraint

$$\sum_{p,a,d \leq \bar{d}} \pi_{(pd),a} \geq 1 - \alpha$$

limits the probability of having more than $\bar{d}$ time units of delay. Alternatively, the constraints can be imposed per port position:

$$\sum_{a,d \leq \bar{d}} \pi_{(pd),a} \geq 1 - \alpha_p \quad \forall p \in \mathcal{P}.$$

Fixed arrival times in some ports

In the standard formulation, we assume that we can choose the arrival times in all ports that are visited on the route freely. When we fix the arrival time in one port, the remainder of the route is also fixed by the output of the model. However, for some ports it might be difficult to change an existing time slot or there can be limited options to arrive, such as in the Suez Canal, where it is highly desirable to arrive on time for a Suez convoy. In these cases, it would be useful to adapt the formulation in such a way that the arrival times in some ports can already be fixed beforehand. Note that fixing the arrival times in two ports limits the amount of freedom to allocate buffer times in the ports in between.

Consider an example in which we want to fix two ports, A and B, on a route that will take 25 time units including a total buffer time of 5 time units. Assume that the total sailing and port time between A and B is 12 time units and 8 time units between B and A. If we fix the arrival time in port B to be 15 time units after the arrival time in port A, we know that the total buffer time to be allocated to the part of the route from A to B will be $15 - 12 = 3$ time units, while $(25 - 15) - 8 = 2$ time units of buffer can be allocated to the part of the route from port B to port A. Hence, we can add constraints

$$\sum_{p \in \mathcal{P}_s} \sum_{b \in \mathcal{B}} B_b y_{pb} = M_s$$

to the model to ensure that the total amount of buffer allocated to subpart s of the routes is equal to the available buffer of that subpart (M_s). The values of M_s can be calculated using the fixed arrival times and the known sailing and port times as shown in the example above and the set $\mathcal{P}_s$ contains the port positions that are visited on subpart s of the route. If it is essential to also arrive without or with a limited amount of delay in one of the ports, the chance constraints from the previous section can be additionally added to the model to ensure timely arrivals.

4.3 Structural results

Before we introduce heuristics to solve the ship delay recovery problem with buffer time allocation, we prove the correctness of two structural results.

Theorem 1.

1. *If the total buffer time might also be allocated only partially, the total cost incurred in the ship delay recovery problem with buffer time allocation is non-increasing in the total amount of buffer time that can be allocated.*

2. *If the total amount of buffer time has to be allocated completely, the total cost incurred in the ship delay recovery problem with buffer time allocation is non-increasing in this total amount.*

Proof. First, we prove Part 1 of Theorem 1. Let (π_B^*, y_B^*) be an optimal solution to the problem with B units of buffer. Let z_B^* be the costs corresponding to this solution. Then (π_B^*, y_B^*) is also a feasible solution to the problem with $B + 1$ units of buffer. Thus, it holds that

$$z_B^* \geq z_{B+1}^*,$$

where z^*_{B+1} is the cost corresponding to the optimal solution of the problem with $B+1$ units of buffer.

Now, we prove Part 2 of Theorem 1. Let (π^*_B, y^*_B) be an optimal solution to the problem with B units of buffer. Let z^*_B be the costs corresponding to this solution and let k^*_B denote the optimal actions taken in each of the states. We can randomly select an initial state and construct a sample path σ_B by randomly selecting a delay for each port position. The costs of σ_B can be calculated and compared to the costs of a sample path σ_{B+1} that has the same initial state, actions and delays. The buffer time allocation of σ_{B+1} is also equal to that of σ_B, except for one port position p for which the buffer time is one unit higher than it is for σ_B. Since the sample paths σ_B and σ_{B+1} are obtained using the same actions, delays and buffer (except for port position p), the costs of σ_B and σ_{B+1} are equal until the ship arrives in port position p. If the ship arrives according to σ_B with a delay of d units in port position p ($d > 0$), the ship will have $d-1$ units of delay in this port according to sample path σ_{B+1}. Thus, in sample path σ_B a cost of $C_{(p,d),k}$ is incurred, while for sample path σ_{B+1} a cost of $C_{(p,d-1),k}$ is incurred. Since the costs are monotonically increasing in the amount of delay, we know that $C_{(p,d-1),k} \leq C_{(p,d),k}$. If, on the other hand, the ship arrives without delay in port position p, then the ship will also arrive without delay in port position p according to sample path σ_{B+1}, so the costs of both sample paths are equal in this case. Now, for the remaining port positions, the delay of sample path σ_B will be larger than or equal to the delay of sample path σ_{B+1}. Since the costs are non-decreasing in the amount of delay for all port positions, this means that the total costs of sample path σ_{B+1} will be at most as high as the total costs of sample path σ_B. Thus, we have found a feasible solution to the problem with $B+1$ units of buffer time in which all buffer time is allocated and the total costs of a randomly generated sample path are at most as high as the total costs of the corresponding sample path associated with the optimal solution of the problem with B units of buffer time. Since this holds for every sample path, the total costs of the problem with $B+1$ units of buffer time (z_{B+1}) will be at most as high as the optimal costs of the problem with B units of buffer. Hence, $z^*_{B+1} \leq z^*_B$ and the total costs are monotonically decreasing when the amount of buffer time is increasing. □

For the next theorem we need to make some assumptions. Note, that these assumptions are not always realistic, but we only use the theorem to illustrate the idea of some of the heuristics we will introduce in the next section. The first assumption is a quite common one and is often satisfied in practice. The second and third assumption are not always satisfied. Probably the optimal recovery actions change at some states when the buffer distribution is changed, in which case the third assumption will be violated. Furthermore, the additional delay incurred during a transition may depend on the current delay, in which case the second assumption is violated. These are some of the reasons that the proposed greedy buffer heuristic in the next section does not always return the optimal solution.

Assumption 1. *The cost function is convex in the amount of delay.*

Assumption 2. *The additional delay incurred during a transition does not depend on the current delay with respect to the schedule.*

Assumption 3. *The recovery policy is the same for every buffer time allocation.*

Theorem 2. *If Assumptions 1-3 are satisfied, then for each port, the reduction in total cost is monotonically decreasing in the amount of buffer time that is allocated to that port.*

Proof. Let k^*_B be the optimal recovery policy for a fixed buffer time allocation of B units. Consider increasing the buffer time in port position p^* with one or two time units. Let σ_B be a randomly generated sample path that uses recovery policy k^*_B. Let σ_{B+1} and σ_{B+2} be the sample paths obtained using the same initial state, additional delays and recovery policy as in σ_B, but in which the buffer in port position p^* is increased with respectively one and two time units compared to the buffer allocation in σ_B. Let the delay in a port position p of sample path σ_B be denoted by d_p^B. Similarly, d_p^{B+1} and d_p^{B+2} are the delays in port position p of sample paths σ_{B+1} and σ_{B+2} respectively. Since the sample paths σ_B and σ_{B+1} are obtained using the same actions, delays and buffer except for port p^*, the costs of σ_B and σ_{B+1} are equal until the ship arrives in port p^*. If the ship arrives with a delay of d units in port p^* ($d > 0$) according to σ_B, the ship will have $d-1$ units of delay in this port according to sample path σ_{B+1}. A similar reasoning can be given for sample paths σ_{B+1} and σ_{B+2}. Thus, we know that both $d_p^{B+2} - d_p^{B+1}$ and $d_p^{B+1} - d_p^B$ are either equal to 0 or to 1. Furthermore, we know that if $d_p^{B+1} - d_p^B = 0$ then also $d_p^{B+2} - d_p^{B+1} = 0$, because the delay in port position p will only be the same if the ship encounters no delay in this port position. Thus, by increasing the buffer time with one unit, we either find no cost reduction, only a cost reduction for adding the first unit of buffer or we find both a cost reduction for adding the first unit of buffer and for adding the second unit of buffer time. The cost reduction of adding the first unit of buffer is clearly larger or equal to the cost reduction of adding the second unit of buffer in the first two cases. In the last case, the cost reduction of adding the first unit of buffer is equal to $C_{(p,d),k} - C_{(p,d-1),k}$ and the cost reduction of adding the second unit of buffer is equal to $C_{(p,d-1),k} - C_{(p,d-2),k}$. Since the cost function is convex in the amount of delay, we know that $C_{(p,d),k} - C_{(p,d-1),k} \geq C_{(p,d-1),k} - C_{(p,d-2),k}$. Hence, the cost reduction of adding the first unit of buffer is larger than that of adding the second unit of buffer. Again, since this holds for all port positions and sample paths and the additional delay incurred during a transition does not depend on the current delay (thus all three sample paths associated with the three different buffer time allocations occur with equal probability), we have proven that the cost reduction of adding consecutively units of buffer to a certain port for an existing buffer time allocation when the recovery policy is fixed, is monotonically decreasing in the amount of buffer that is added. □

4.4 Heuristics

The proposed mixed integer programming model can be solved using standard mixed integer programming solvers. However, for larger instances, the solution time increases exponentially and faster methods are necessary to obtain satisfactory results in a reasonable amount of time. Therefore, we propose four heuristics that can be used to solve the ship delay recovery problem with buffer time allocation in this section.

4.4.1 Greedy buffer heuristic

The idea of the greedy buffer heuristic (GBH) is to start without any buffer time and step by step allocate units of buffer until the maximum amount of buffer is reached. In each iteration of the greedy buffer heuristic, the cost reduction of increasing the buffer time with one unit in a certain port compared to the current buffer allocation is calculated for each port, where the buffer time for all other ports remains the same as in the current allocation. Since the problem reduces to a Markov decision process when the buffer time allocation is fixed, $|\mathcal{P}|$ Markov decision processes have to be solved in each iteration. The algorithm is summarized in Algorithm 6.

Algorithm 6: Greedy buffer heuristic

1. *Initialization:* Each port position has 0 units of buffer.
2. Allocate the next unit of available buffer time until all units are allocated:
   - *Determine savings:* Determine for each port position the cost reduction when one unit of additional buffer is added to this position.
   - *Buffer allocation:* Allocate the n-th unit of buffer to the first port position that corresponds to the highest cost reduction.

Notice that it follows from Theorems 1.2 and 2 that the greedy buffer heuristic would return the optimal solution if the cost reduction of adding one unit of buffer time to a port position is independent on the allocated buffer times to other port positions and the underlying assumptions of the theorems are valid. In this case, the cost reduction of adding one additional unit of buffer would only change for the port position to which the last unit of buffer is allocated. From Theorem 2 it then follows that this new cost reduction will be at most as high as the previous one. Thus, the heuristic selects the M best positions to allocated the buffer time to. However, since the cost reduction of adding one unit of buffer is in reality dependent on the allocated buffer times to other port positions and we already concluded in the previous section that the underlying assumptions of Theorem 2 are not always satisfied, the greedy buffer heuristic will not be guaranteed optimal in reality.

4.4.2 Buffer exchange heuristic

In the buffer exchange heuristic (BEH), we start with a feasible buffer time allocation and exchange in each iteration units of buffer between two port positions in order to obtain a better buffer allocation. The heuristic starts with exchanging as much buffer as possible between the two port positions for which the change is most profitable. In this way, the solution space is quickly explored. When it is not longer profitable to make a change, the amount of buffer to be exchanged in each iteration is decreased to find the best solution in the direct neighborhood of the current solution. Algorithm 7 gives a description of the buffer exchange heuristic.

Algorithm 7: Buffer exchange heuristic

1. *Initialization:* Start with a feasible buffer allocation and set b equal to 2^z for $z \in \mathbb{N}$ such that $2^z \leq b^{max}$ and $2^{z+1} > b^{max}$, with b^{max} the largest amount of buffer in one of the port positions.

2. Repeat the following as long as the $b \geq 1$

   - *Determine savings:* For each combination of two port positions exchange b units of buffer and compute new total costs if the new buffer allocation is feasible.
   - *Buffer exchange:* If a profitable exchange exists, make the first most profitable exchange. Else, $b = b/2$.

4.4.3 Value function

Instead of solving a Markov decision process to evaluate the exact costs of a change in the buffer allocation, it is also possible to use the value function of the MDP in order to estimate the costs of the change. The value function denotes for every state the total expected average costs incurred when following the current (optimal) recovery policy starting at the considered state. The value function of state i will be denoted by ν_i and can be obtained as the dual variables of (4.2)-(4.5). The following relation will hold for the value function:

$$\nu_i = C_{ik^*(i)} + \sum_{j \in \mathcal{I}} p_{ijk^*(i)} \nu_j$$

where k^* denotes the optimal recovery policy, hence $k^*(i)$ is the optimal action to be performed when the process is in state $i \in \mathcal{I}$.

In order to estimate the value function for a slightly changed buffer allocation given the initial allocation, we assume that the optimal recovery policy also only changes slightly. If the buffer in a port position is increased by one unit, there are two possibilities: either the action can be reduced by one unit in which case the delay in the next port position will remain the same or the action will remain the same in which case the delay in the next port position will be reduced by one unit. Let $k'(i)$ be the recovery action that performs one unit less action if possible: $g_{actI'(i)} = g_{k^*(i)} - 1$ and let $C_{ik} = \infty$ for all $k \notin \mathcal{K}(i)$ and $\nu_{p,-1} = \nu_{p,0}$ for all $p \in \mathcal{P}$. Then, the value function for the new buffer allocation can be estimated using:

$$\nu_i^+ = \min \left\{ C_{ik'(i)} + \sum_{j \in \mathcal{I}} p_{ijk^*(i)} \nu_j, C_{ik^*(i)} + \sum_{j \in \mathcal{I}} p_{ijk^*(i)} \left(\nu_j - \nu_{j-1} \right) \right\},$$

where $\nu_{i-1} = \nu_{p,d-1}$ with $i = (p, d)$. Note that the transition probabilities are still the same as in the original value function.

Furthermore, if the buffer in a port position is decreased with one unit, a similar reasoning holds. In this case, we can either increase the action by one unit in which case the delay in the next port position will remain the same or the action will remain the same in which case the delay in the next port position is increased by one unit. In cases where the delay in the next port position is already at the highest allowable level, we need to pay one unit of penalty. For this case, let $k'(i)$ be such that $g_{k'(i)} = g_{k^*(i)} + 1$ and $\nu_{p,d^{max}+1} = \nu_{pd^{max}} + c^p$, where c^p is the penalty per unit of exceeding the maximum delay.

The value function in case of a reduction in the buffer can be estimated using:

$$\nu_i^- = \min \left\{ C_{ik'(i)} + \sum_{j \in \mathcal{I}} p_{ijk^*(i)} \nu_j, C_{ik^*(i)} + \sum_{j \in \mathcal{I}} p_{ijk^*(i)} \left(\nu_j - \nu_{j+1} \right) \right\},$$

where in this case $\nu_{i+1} = \nu_{p,d+1}$ with $i = (p, d)$.

Using these formulas, the gain of adding one unit of buffer in a port position can be estimated using $\nu_i - \nu_i^+$ and the costs of removing one unit of buffer by $\nu_i^- - \nu_i$. These formulas only provide estimations because we only consider local updates after a change. However, using these estimates is much faster than resolving the Markov decision process for every possible change.

Using these estimates, we can make a variant of both the greedy buffer heuristic and the buffer exchange heuristic. In the variant of the greedy buffer heuristic, the only change is that the savings are not exactly determined, but estimated using the value function. We will refer to this variant as the greedy buffer heuristic using the value function (GBHVF). The variant of the buffer exchange heuristic will be referred to as the value iteration heuristic (VIH). For the value iteration heuristic, the savings of adding an additional unit of buffer time and costs of reducing the buffer time by one unit are estimated instead of exactly determined, but now also the amount of buffer to be exchanged is limited to one unit, in order to avoid large estimation errors. If the expected costs are larger than the expected savings, the change is still made, but the best solution is always stored. The value iteration heuristic stops when a cycle is detected and the best solution found in one of the iterations is returned. Algorithms 8 and 9 provide a schematic overview of these heuristics.

Algorithm 8: Greedy buffer heuristic using the value function

1. *Initialization:* Each port position has 0 units of buffer.
2. Allocate the next unit of available buffer time until all units are allocated:
   - *Determine savings:* Determine for each port position the cost reduction when one unit of additional buffer is added to this position using the value function estimate.
   - *Buffer allocation:* Allocate the n-th unit of buffer to the first port position that corresponds to the highest expected cost reduction.

Algorithm 9: Value iteration heuristic

1. *Initialization:* Start with a feasible buffer allocation and set $b = 1$.
2. Repeat the following as long as no cycle is detected:
   - *Determine savings:* For each combination of two port positions, use the value function to estimate the total costs when b buffer units are exchanged between these port positions.
   - *Buffer exchange:* Make the most profitable exchange
   - *Evaluate:* Determine the cost of the new buffer allocation and store the cost and allocation when it is better than the current best allocation.

4.5 Computational Experiments

4.5.1 Case study

A case study has been conducted at a liner shipping company. From the data they provided, real transition probabilities could be determined. The liner shipping company was interested in several objectives, like maintaining punctuality levels and minimizing costs. The fuel costs of sailing one round trip are substantial and typically several millions of USD. By optimizing recovery actions, like increasing the sailing speed, and the buffer time allocation to the ports substantial savings can be made. Also, the value of adding one hour of buffer time in a certain port can be determined, which can help the liner shipping company in negotiations with port terminals. However, since the data of the liner shipping company is confidential, we cannot use it in this paper. Therefore, we used publicly available data combined with randomly generated probability distributions of incurring additional delays as explained in the next section, to show the performance of our methods.

4.5.2 Data

To test our methods, we use the ME1 route in September 2012 of the Maersk Line network, which we have slightly adapted in order to obtain valid input for our model. First of all, we have to discretize the time in order to obtain a finite number of states in our model. We decided to discretize time in time units of 4 hours with a maximum delay of one week or 42 time units. In the case study we only include recovery actions and buffer times between ports. However, it is also possible to include recovery actions and buffer times in ports in the model. Table 4.1 shows the order in which the ports are visited and some other characteristics of the route. The second column of the table shows the total time planned in the port to load and unload the ship. In the third column of Table 4.1 the distances between the ports in nautical miles are presented. Distances are obtained from SeaRates (2015). This column shows for each port the distance that the ship has to cover to sail from that port to the next port. For example, the distance between Algeciras and Felixstowe can be found in the row of Algeciras and is equal to 1476 nautical miles. The fourth column shows the scheduled sailing time in hours and the fifth column the sailing time needed to sail the distance using a speed of 23 knots (1 knot = 1 nmi/hr = 1.852 km/hr). Sailing times in both columns are rounded to multiples of four hours, such that integer time units will occur in the model. The scheduled and expected sailing times for Felixstowe are respectively 16 and 8 hours, meaning means that a ship needs 8 hours to sail from Felixstowe to Antwerp, when it sails at 23 knots and does not encounter a delay during its trip, but that 16 hours is planned for this trip. The expected sailing times can be computed by dividing the distance between two ports by the sailing speed. The last column shows the buffer times for the ports assuming that the ship will sail at 23 knots during the round tours. These buffer times are subtracted from the encountered delay to determine the delay with respect to the time schedule when the ship arrives in the port. The time needed to make one full round tour is 1176 hours (7 weeks). The total bunker cost of one round tour are \$3.93 million for a ship that follows the published schedule.

Port	Port time (hr)	Distance (nmi)	Scheduled sailing time (hr)	Sailing time at $v = 23$ (hr)	Buffer time (hr)
Jebel Ali	31	1,329	72	60	12
Jawaharlal Nehru	33	443	24	20	4
Mundra	16	1,122	56	52	4
Salalah	14	1,553	68	68	0
Jeddah	11	778	36	36	0
Suez Canal	16	2,283	100	100	0
Algeciras	18	1,476	88	68	20
Felixstowe	24	156	16	8	8
Antwerp	16	366	32	16	16
Bremerhaven	24	283	24	16	8
Rotterdam	20	3,829	192	168	24
Suez Canal	22	395	20	20	0
Aqaba	20	656	40	32	8
Jeddah	19	2,648	124	116	8

Table 4.1: Characteristics of the route

We include both the possibility to increase speed and to decrease speed. For each sea leg, we include actions to decrease or increase the sailing speed in such a way that the required speed will be in between the minimum and maximum speed limit and an integer amount of time will be gained or lost by performing the action. The fuel cost in US dollars per day incurred when sailing at v knots is given by (Brouer et al. 2014a):

$$C_f(v) = \tilde{f} \cdot p_{bunker} \left(\frac{v}{\tilde{v}}\right)^3 \tag{4.14}$$

in which $\tilde{v}$ the design speed of the ship, $\tilde{f}$ the fuel consumption in ton per day of the ship when sailing at design speed, p_{bunker} the bunker price in US dollars per ton. We use a bunker price per ton of \$600, a design speed of 16.5 knots, minimum and maximum speeds of 12 and 23 knots respectively and a fuel consumption of 82.2 ton per day. These values are all obtained from Brouer et al. (2014a).

Using (4.14) and the scheduled speed of each sea leg, which can be obtained from Table 4.1, we can determine the fuel cost for each sea leg when the ship sails according to schedule or takes recovery actions:

$$C_f^i(a) = \tilde{f} \cdot p_{bunker} \left(\frac{\delta_i}{(t_i - g_a)l \cdot \tilde{v}}\right)^3 \cdot \frac{t_i}{24}, \tag{4.15}$$

where $C_f^i(a)$ is the fuel cost in US dollars on sea leg i under recovery action a, δ_i the distance to be covered on sea leg i in nautical miles, t_i the scheduled sailing time for leg i in time units, g_a is the gain of recovery action a in time units and $\tilde{f}$, p_{bunker} and $\tilde{v}$ as defined above. Now, $C_a = C_f^i(a) - C_f^i(0)$ gives the cost of the recovery action a, where we assume that the sailing times at a fixed speed of 23 knots correspond to recovery action 0. Finally, the cost per time unit of delay is assumed to be \$10,000 per time unit (\$2,500 per hour) and the penalty incurred for every unit of delay that exceeds the maximum allowable delay is set to a very large number.

In Section 4.5.3, the performance of the route schedule with optimized buffer times is determined using a uniform distribution of additional delay for both the sea legs and the port stays. For port stays it holds that $\bar{d} \sim U(0,2)$ and for sea legs $\bar{d} \sim U\left(0, \lfloor 2 + \frac{\delta_p}{800} \rfloor\right)$. The results of this optimized schedule are compared to the current schedule and to the current schedule with optimized recovery actions. These distributions are also used in the results of the extensions described in Section 4.2.3.

In Section 4.5.5, the performance of the heuristics is compared to the performance of the mixed-integer linear programming model. In this section, fifty test cases are generated. For each port stay and sea leg, a uniform probability distribution of incurring additional delay is generated at random. The maximum amount of additional delay has to be between a lower and an upper bound for each port stay and sea leg. Furthermore, three different problem settings are compared in this section. The first setting is the setting as described above with convex (linear) delay costs and only speed related recovery actions. In the second setting, the delay costs are truncated after six time units (one day). Hence, the costs are linear increasing as described above for the first six time units of delay, but will remain at the same level if the delay becomes larger. We will refer to this cost function as the concave cost setting. Finally, in the last setting, the costs are also assumed to be concave and an additional recovery action is considered: port skipping is also possible for some ports. In this setting, three ports can be skipped: Mundra, Antwerp and Aqaba. The distances of the sea legs that will be sailed in case these ports are skipped are respectively 1266, 1560, and 778 nautical miles. If a port is skipped, the port time of this port is added to the available sailing time of the new sea leg.

4.5.3 Results optimized buffer time distribution

The average expected costs incurred during a round tour when sailing according to the current schedule is 3.969 million USD. These costs reduce with 11.7% to 3.505 million USD when the recovery policy is optimized given the current buffer allocation. Furthermore, the average expected round tour costs, when both the buffer allocation and recovery policy are optimized, are equal to 3.364 million USD, which is a reduction of 15.2% with respect to the original schedule. The optimized schedule is found in 1381 seconds, while the average costs of the current schedule and the schedule with optimized recovery policy can be calculated in less than a second.

Tables 4.2 and 4.3 show the buffer allocation, the average arrival delay, on time probabilities and average fuel consumption for the current schedule, the current schedule with optimized recovery actions and the schedule with optimal buffer allocation and recovery actions. The average arrival delays and on time probabilities are more evenly distributed for the optimal schedule than for the current schedule. In the current schedule a lot of buffer time is allocated to for example the sea legs from Algeciras to Felixstowe and from Antwerp to Bremerhaven, resulting in lower average arrival delays and higher on time probabilities in Felixstowe and Bremerhaven than in respectively Algeciras and Antwerp. Furthermore, no buffer time is allocated to the sea legs from Salalah tot Jeddah, from Jeddah to the Suez Canal and from the Suez Canal to Algeciras in the current schedule, which results in relatively high average arrival delays and low on time probabilities in Jeddah, the Suez Canal and Algeciras. This is an indication that a part of the buffer time is unused in some ports of the route, while adding more buffer time to other ports can reduce the delay significantly. Therefore, it is probably more efficient to reallocate the allocated buffer times in the route. In the optimized schedule, we see indeed that the amount of buffer is more evenly spread between these ports.

		Max speed			Opt speed		
Port	Buffer time (hr)	Average delay (hr)	On time prob	Avg fuel cons (ton)	Average delay (hr)	On time prob	Avg fuel cons (ton)
Jebel Ali	12	0.00	1.00	497	4.17	0.24	391
Jawaharlal Nehru	4	0.00	1.00	166	2.00	0.50	140
Mundra	4	0.00	1.00	398	2.00	0.50	398
Salalah	0	0.00	1.00	618	1.00	0.75	618
Jeddah	0	2.00	0.50	277	3.00	0.38	277
Suez Canal	0	4.00	0.25	907	5.00	0.19	801
Algeciras	20	8.00	0.06	530	15.50	0.00	412
Felixtowe	8	0.00	1.00	45	6.75	0.02	35
Antwerp	16	0.00	1.00	146	2.42	0.45	57
Bremerhaven	8	0.00	1.00	68	0.12	0.97	44
Rotterdam	24	0.00	1.00	1,517	0.00	1.00	1,211
Suez Canal	0	0.00	1.00	117	2.67	0.50	117
Aqaba	8	2.00	0.50	210	4.67	0.25	181
Jeddah	8	0.00	1.00	1,052	1.83	0.58	947

Table 4.2: Statistics for the current route schedule

Table 4.4 shows the optimal recovery policy for the schedule with optimized buffer times. The number in the table corresponds to the sailing time in time units of the selected recovery action. Thus, when a ship for example departs from Rotterdam with a delay of 3 time units, it will determine its speed in such a way that it will use 44 time units to sail to the Suez Canal. Hence, the ship will decrease its speed, because at 23 knots it will need 42 time units to reach the Suez Canal. Since 23 knots happens to be the maximum speed of the ship, the used time is always larger than the planned time at $v = 23$ knots, denoting a speed reduction.

4.5.4 Results extensions

Results chance constraint on delay

We performed some experiments in which we minimized the costs related to recovery actions with respect to a punctuality constraint as explained in Section 4.2.3. Table 4.5

Port	Buffer time (hr)	Average delay (hr)	On time prob	Avg fuel cons (ton)
Jebel Ali	8	2.63	0.48	421
Jawaharlal Nehru	4	2.00	0.50	140
Mundra	4	2.00	0.50	371
Salalah	8	2.00	0.50	523
Jeddah	4	2.00	0.50	251
Suez Canal	16	2.00	0.50	723
Algeciras	8	2.50	0.50	456
Felixtowe	4	2.00	0.50	45
Antwerp	4	1.00	0.75	107
Bremerhaven	4	2.00	0.50	68
Rotterdam	24	1.00	0.75	1,224
Suez Canal	4	2.67	0.50	100
Aqaba	4	2.67	0.42	192
Jeddah	16	2.33	0.46	866

Table 4.3: Statistics for the optimized route schedule

Port	Delay (time units)							Min sailing time
	0	1	2	3	4	5	≥ 6	
Jebel Ali	17	16	15	15	15	15	15	15
Jawaharlal Nehru	6	5	5	5	5	5	5	5
Mundra	14	13	13	13	13	13	13	13
Salalah	19	18	17	17	17	17	17	17
Jeddah	10	9	9	9	9	9	9	9
Suez Canal	28	28	27	26	25	25	25	25
Algeciras	19	18	17	17	17	17	17	17
Felixtowe	2	2	2	2	2	2	2	2
Antwerp	5	4	4	4	4	4	4	4
Bremerhaven	4	4	4	4	4	4	4	4
Rotterdam	47	46	45	44	43	43	42	42
Suez Canal	6	5	5	5	5	5	5	5
Aqaba	9	8	8	8	8	8	8	8
Jeddah	32	32	31	30	29	29	29	29

Table 4.4: Sailing time in time units under the optimized schedule

shows the average expected costs per round tour in millions USD and the solution times in seconds for the problems with $\alpha \in \{0.25, 0.20, 0.15, 0.10, 0.05\}$. If α is for example equal to 0.2, we added a constraint to the model that the punctuality (percentage of ships arriving without delay in a port) is at least equal to $1 - \alpha = 0.8$. If a higher punctuality is required (a lower value of α), more recovery actions should be performed to satisfy the punctuality constraint and thus higher costs will be incurred. Indeed, in the table we see that the average costs per port increase when we reduce the value of α. Furthermore, the increase in average costs caused by the reduction of α increases when α becomes smaller. The reduction from $\alpha = 0.25$ to $\alpha = 0.2$ only leads to an average cost increase of 0.65%, while the next reductions lead to increases of respectively 0.68%, 0.70% and 0.88% in average costs. The reason for this is that is becomes more difficult to increase the punctuality, because ships never leave before the scheduled time. The average solution time decreases when the minimum punctuality is increased, since the feasible regions will also be significantly decreased. Finally, the optimal solutions have the minimal accepted punctuality for all values of α.

α	Costs in millions USD	Solution time in seconds	Punctuality in %
0.25	3.372	489	75.0
0.20	3.394	269	80.0
0.15	3.417	224	85.0
0.10	3.441	204	90.0
0.05	3.471	118	95.0

Table 4.5: Optimal solutions under different punctuality constraints

Results fixed arrival times

Next, we run an experiment in which two arrival times are fixed in the schedule. We fix the arrival times at the Suez Canal passage, because these arrival times are usually limited based on the departure times of Suez convoys. From the current schedule, we conclude that 19 time units of buffer (76 hours) are allocated between the passage of the Suez Canal in western direction and the passage in eastern direction and 9 time units of buffer (36 hours) are allocated to the other part of the route. The average expected costs per round tour become 3.385 million USD when the arrival times at both Suez Canal passages are fixed. This is an increase of only 0.6% with respect to the average costs of the optimized schedule in which we could choose all buffer times freely. Furthermore, the average costs per port are still 14.7% lower than in the current schedule. The optimal schedule with fixed arrivals at the Suez Canal passage is found in 272 seconds.

Table 4.6 shows the buffer times, average arrival delays, on time probabilities and the average fuel consumption of the optimized schedule with fixed arrival times at the Suez Canal passage. The average arrival delays between the Suez Canal passage in western direction and the Suez Canal passage in eastern direction are typically lower than the arrival delays in the other part of the route (on average 1.28 hours and 2.27 time units respectively). Indeed, in the optimal schedule determined in Section 4.5.3 more buffer time is allocated to the part of the route between the Suez Canal passage in eastern direction and the passage in western direction (52 instead of 36 hours).

Port	Buffer time (hr)	Average delay (hr)	On time prob	Avg fuel cons (ton)
Jebel Ali	4	1.13	0.73	453
Jawaharlal Nehru	4	2.04	0.49	141
Mundra	4	2.02	0.50	371
Salalah	8	2.01	0.50	523
Jeddah	0	2.00	0.50	277
Suez Canal	24	4.00	0.25	675
Algeciras	8	1.00	0.75	437
Felixtowe	4	2.00	0.50	45
Antwerp	4	1.00	0.75	107
Bremerhaven	4	2.00	0.50	68
Rotterdam	32	1.00	0.75	1,174
Suez Canal	0	0.67	0.83	117
Aqaba	4	2.67	0.42	192
Jeddah	12	2.33	0.46	955

Table 4.6: Statistics for the optimized schedule with fixed arrival times at the Suez Canal

4.5.5 Comparison of the methods

This section describes the results obtained when comparing the different solution methods: the mixed integer programming model and the four heuristics introduced in Section 4.4 (the greedy buffer heuristic, the buffer exchange heuristic, the greedy buffer heuristic using the value function and the value iteration heuristic). Recall that we constructed fifty different uniform probability distributions that denote the probabilities of obtaining additional delays for three problem settings: convex cost setting, concave cost setting and concave cost setting with the possibility to skip ports. For each problem setting, fifty instances of the problem are constructed (differing in their delay distribution) and solved using the different solution methods. The maximum amount of buffer time that can be allocated to a port is limited to the maximum possible incurred delay value if the possible buffer allocation for a port become too large otherwise, such that the mixed integer program is able to find good solutions within an hour for almost all cases. If the mixed integer programming model cannot be solved to optimality, we do not have another method to obtain the optimal solution, so we can make a better comparison when we limit the maximum amount of buffer per port. The gap with the best bound is calculated as the difference between the best solution returned by a method and the best lower bound obtained from the MIP divided by the solution of the method and transformed to percentages.

Tables 4.7-4.9 show some characteristics of the solutions of fifty instances of the route network. First of all, the number of proven optimal solutions is shown in the table. Because the time limit is set to 3600 seconds, the MIP model will not always return the optimal solution. For all methods, the number of proven optimal solutions correspond to instances in which the mixed integer programming model was solved to optimality within 3600 seconds. The buffer exchange heuristics always returns the optimal solution if the MIP is able to find the optimal solution. However, we can see from the maximum difference with the MIP solution, that the buffer exchange heuristic returns not always the optimal solution. The heuristics using the value function as estimate for the cost change are much faster than the heuristics that consider all changes in order to determine the exact cost change. Furthermore, the performance of the heuristics that use the value function is comparable to the corresponding heuristic that does not use the value function. When comparing all instances and problem settings, the value iteration heuristic is preferred because it computes good solutions in little time.

In the convex and concave problem settings, the MIP model is not always able to find a good solution and lower bound within an hour, resulting in large maximum gaps with the bound for all methods. Adding the possibility to skip ports strongly improves the gaps with the bound obtained in the MIP and hence in the other methods. However, three out of four heuristics clearly outperform the MIP in this setting. The computational times of the heuristics sharply increase for the concave problem setting with the possibility to skip ports, because the complexity of the Markov decision problems that have to be solved in each iteration increases. The time limit of the heuristics is also set to one hour and the buffer exchange heuristic did not always terminate within one hour.

	GBH	BEH	GBHVF	VIH	MIP
Number of proven optimal solutions	0	22	0	6	22
Average gap with bound (%)	8.62	6.15	9.27	6.35	6.17
Maximum gap with bound (%)	100.00	100.00	100.00	100.00	100.00
Average difference with MIP solution (%)	29.26	-0.45	34.27	-0.12	-
Maximum difference with MIP solution (%)	964.82	0.02	1,136.66	1.00	-
Number of times better than MIP solution	0	14	0	6	-
Number of times worse than MIP solution	42	6	50	37	-
Average computational time (s)	48.8	276.3	3.9	1.6	2,730.4
Number of times fastest	0	0	0	50	0

Table 4.7: Characteristics of the solutions for the convex cost setting

	GBH	BEH	GBHVF	VIH	MIP
Number of proven optimal solutions	10	15	0	10	15
Average gap with bound (%)	6.11	5.81	6.61	5.96	5.87
Maximum gap with bound (%)	90.79	88.99	92.15	89.38	89.95
Average difference with MIP solution (%)	0.74	-0.53	2.29	-0.20	-
Maximum difference with MIP solution (%)	9.05	0.00	28.03	0.99	-
Number of times better than MIP solution	0	14	0	6	-
Number of times worse than MIP solution	15	0	50	21	-
Average computational time (s)	58.5	153.6	3.8	1.0	2,985.8
Number of times fastest	0	0	0	50	0

Table 4.8: Characteristics of the solutions for the concave cost setting

4.6 Conclusion

In liner shipping networks ships encounter problems when they are delayed. Since companies buy time slots and space in ports, delays will result in additional costs. Some actions, such as increasing the sailing speed of the ship or increasing the port handling capacity, can decrease the future costs of delay.

We presented a Markov decision process that can be used to determine an optimal recovery policy for a given liner shipping route. The recovery policy specifies for each combination of port position and delay which recovery action should be performed when a ship is at that specific combination. Since Markov decision processes can be solved as a linear programming model, optimal recovery policies for a given buffer time allocation can easily be determined.

Furthermore, we presented a mixed integer linear programming model to formulate the ship delay recovery problem with buffer time allocation. To the best of our knowledge,

	GBH	BEH	GBHVF	VIH	MIP
Number of proven optimal solutions	0	0	0	0	0
Average gap with bound (%)	10.72	10.69	11.48	10.69	11.32
Maximum gap with bound (%)	20.40	20.04	20.95	19.98	21.61
Average difference with MIP solution (%)	-0.70	-0.73	0.17	-0.73	-
Maximum difference with MIP solution (%)	0.03	0.18	1.17	0.06	-
Number of times better than MIP solution	46	48	15	46	-
Number of times worse than MIP solution	2	2	35	4	-
Average computational time (s)	586.2	3,505.4	77.6	90.6	3,600.0
Number of times fastest	0	0	30	20	0

Table 4.9: Characteristics of the solutions for the concave cost setting with the possibility to skip ports

such a formulation does not exist in the literature yet. Furthermore, we introduced two extensions of the problem: optimizing the recovery policy and buffer time allocation under a punctuality constraint and when the arrival times in some ports of the schedule are fixed. For small problems, the mixed integer programming model can be solved quite fast. However, for larger instances, the computational times increase exponentially. Therefore, we also presented four heuristical methods to solve the problem.

We constructed a data set based on a liner shipping route as used by Maersk Line in September 2012 with uniform probabilities of additional delay for each port position. We included recovery actions that increase or decrease the sailing speed on a sea leg and used the fuel consumption and bunker oil price to determine the cost of these actions. The route with optimal buffer time allocation and recovery policy resulted in a cost decrease of 15.2% compared to the current route. Furthermore, the results showed a more evenly distributed on time probability in each of the ports.

Next, we performed some experiments with constraints on the minimum punctuality. The minimum punctuality is varied between 75% to 95%. If the minimum punctuality increases, the average costs per round tour also increase, but the solution time decreases. Furthermore, the increase in average costs per round tour also increase for stricter punctuality requirements.

Thereafter, we have run an experiment in which we fixed the arrival times at both Suez Canal passages in the schedule. The average costs per round tour are only 0.6% higher than in the schedule with freely optimized buffer times. Furthermore, the average costs are decreased by 14.7% with respect to the original schedule.

Finally, we generated fifty test instances to compare the results of the heuristics on three problem settings: convex cost setting, concave cost setting and concave cost setting with the possibility to skip ports. The value iteration heuristic is able to find reasonably good solutions for the ship delay recovery problem with buffer time allocation in very little time. The buffer exchange heuristic returns on average slightly better solutions than the value iteration heuristic, but is much slower than the value iteration heuristic. If the problem instances increase, the value iteration heuristic will be less affected in computational time and will hence be preferred over the buffer exchange heuristic.

Chapter 5

Joint optimization of speed and buffer times in transportation systems

5.1 Introduction

Timetables are used in container shipping, airlines and public transport to communicate planned arrival and departure times in advance to customers. However, delays are inevitable while executing the timetable, making the arrival times uncertain. Maintaining timetable reliability despite these delays is crucial: The timetable is relied upon by passengers and freight forwarders.

Transport companies combine two main methods to ensure a reliable schedule. Firstly, during timetable *development*, a more delay-resistant planning may be obtained by including buffer or slack time. In liner shipping, for example, the planned arrival at the port of Jeddah could be 9 days after the planned departure from Rotterdam, while the trip takes only 8 days on average when sailing at design speed. The 24 hours buffer time can capture (part of) a delay. But buffers increase the nominal travel time and therefore costs. So limited buffer time is available, and strategic allocation along a route is key. Secondly, during *execution* of the timetable, a ship may sail faster to recover from a delay with respect to the timetable. But increasing speed is very costly: Figure 5.1 shows that sailing at 28 knots instead of 14 knots increases fuel consumption per nautical mile by about 350% for a 8,000 TEU ship. For a trip from Rotterdam to Jeddah, this corresponds to over 1 million USD at a bunker price of 600 USD/ton, or over $6,000$ tons of CO_2 (Psaraftis and Kontovas 2013). Speed adjustments also have significant impact for other transport modes: For example doubling the average speed of a metro on a track roughly quadruples energy consumption (Binder and Albrecht 2012).

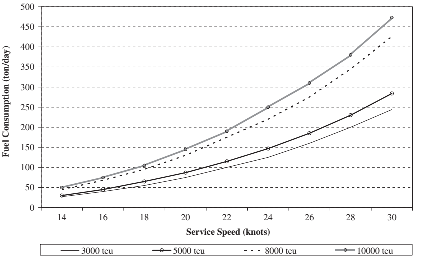


Figure 5.1: Fuel (bunker) consumption of several container ships at different travel speeds (from Notteboom and Vernimmen 2009).

We will focus on the liner shipping application throughout the paper. However, the methodology that we develop applies to timetabling for metro's, trains, and for planning airline operations. Further research investigating the strengths and limitations of the methodology may benefit such applications.

Our model consists of two levels. On the tactical level, construction of the timetable involves the allocation of buffers as we keep the total time constant. On the operational level, the timetable is executed: random events cause (additional) delay, travel speed is optimized, and late arrivals and departures are penalized. We model the operational planning level as a Stochastic Dynamic Program (SDP). This SDP accurately models real-time recovery actions such as speed optimization, as well as propagation of delays from port call to port call. However, the buffer times are exogenous to this SDP: Different buffer time allocations yield SDPs that are structurally different. The optimal buffer allocation yields the SDP which has minimal long run average costs.

We contribute a theoretical analysis of the problem. E.g., we show that speed should increase as the delay with respect to the schedule increases, and provide a bound on the maximum speed increase that should result from additional delay. We then focus on optimizing the buffer time allocation. We develop theoretical results in order to optimally combine the buffer allocation decision on the tactical level, and speed optimization (as part of the SDP) on the operational level. We prove, under mild assumptions, that the minimum costs of operating the timetable are convex in the buffer time allocation. Additional theory is developed, leading to a simple and efficient approach for computing subgradients. Our algorithm for optimal buffer time allocation is based on these results. We then report on a case study based on Maersk data for a round tour consisting of 14 ports.

The remainder of this paper is organized as follows. Section 5.2 reviews the existing literature. Section 5.3 provides a detailed description of the model. The theoretical analysis of the model is presented in Section 5.4. In Section 5.5 we develop the algorithm, and provide further theoretical results underlying the algorithm. Section 5.6 describes computational experiments. We conclude in Section 5.7.

5.2 Literature review

Timetables are often used in air, railway and maritime transport. Multiple studies have already been performed on managing, recovering and preventing delays in these transport modes. Wu and Caves (2003) and Wu (2005) show the importance of buffer time allocation on punctuality in air transport using a simulation approach. Clausen et al. (2010) give an overview on disruption management studies in the airline industry. They distinguish the reviewed in two different groups: delay recovery and robust planning. This distinction also mainly holds for railway and maritime transport.

The goal of delay recovery is to find a recovery policy such that delays in the existing timetables as a result of small disruptions are recovered from in order to minimize a certain objective (e.g. Wang and Meng (2012a), Brouer et al. (2013), Li et al. (2015b), Li et al. (2015a) in liner shipping, Corman et al. (2010), Binder and Albrecht (2012) in public transport and Rosenberger et al. (2003), Petersen et al. (2012), Arikan et al. (2016), Aktürk et al. (2014), Maher (2016) in air transport). In all these studies, the goal is to optimize recovery strategies after the occurrence of disruptions. Recovery strategies include travel time (or speed) adjustments and rerouting decisions. However, the influence of the available buffer time in the existing timetables is not considered in these studies. Visentini et al. (2014) review recovery actions in general transportation, while Psaraftis and Kontovas (2013) overview speed models for energy efficient maritime transportation.

The objective of robust planning is to construct timetables which perform well under uncertainty. Two different approaches are used to construct robust timetables. First, the total available buffer time in an existing schedule can be rearranged in order to obtain more robust networks (e.g. Kroon et al. (2007), Kroon et al. (2008), Fischetti et al. (2009), Hassannayebi et al. (2014), Wu et al. (2015) in public transport, Lan et al. (2006), AhmadBeygi et al. (2010), Chiraphadhanakul and Barnhart (2013) in air transport). All these studies only consider the allocation of buffer times (also framed as time supplements or slack time) in the schedule, but do not consider recovery strategies when disruptions occur. Second, schedules satisfying certain robustness concepts can be constructed. Du et al. (2015) and Norlund et al. (2015) describe methods to design robust schedules that minimize the fuel consumption in shipping taking into account uncertain weather conditions. However, only the fuel consumption of the planned schedule without recovery strategies is taken into account. Cucala et al. (2012) and Duran et al. (2015) consider similar problems for respectively public and air transport. These papers also determine an optimal speed policy together with the constructed timetable, but the speed is independent on incurred delays.

Delay-resistant timetables and real-time recovery actions are interrelated, and in recent years there has been increasing interest in approaches that incorporate both. (The studies referenced above either study delay-resistant timetabling without recovery actions or real-time recovery for fixed timetables.) Various approaches to incorporate wait-depart decisions in timetabling exist: A genetic algorithm (Engelhardt-Funke and Kolonko 2004), a light robustness concept for timetabling combined with scenario-based wait-depart decisions (Liebchen et al. 2010), and a recoverable robustness concept that aims to find timetables that are recoverable when disruptions occur (Cicerone et al. 2009, 2012). Furthermore, Gong et al. (2014) develop a two-stage approach to solve the integrated problem. The first stage considers the timetable optimization and the second stage the speed optimization.

Two-stage stochastic programming (SP) applies naturally to robust timetabling under stochastic delays (Kroon et al. 2007, 2008, Fischetti et al. 2009), and Qi and Song (2012) and Wang and Meng (2012b) have extended this approach to take into account speed adjustments in liner shipping. However, both Qi and Song (2012) and Wang and Meng (2012b) assume that incurred delays are recovered during the first sea leg if the required speed does not exceed the maximum sailing speed. These authors all apply the sample average approximation (SAA) as part of their solution methodology, which gives rise to large mixed integer problems (MIPs). The approaches thus suffer from optimality loss due to the SAA, and typically from additional optimality loss resulting because the large MIPs cannot be solved to optimality.

5.2.1 Contribution

In this paper, we develop a new approach for timetabling under stochastic delays and optimal speed adjustments. Unlike models proposed in literature, we operate the timetable by selecting the *optimal* speed at the start of each leg, which may depend non-linearly on the delay with respect to the schedule (cf. Lemmas 6 and 7 and Table 5.6). This naturally gives rise to a stochastic dynamic problem: Each state of the SDP corresponds to the current location of the ship and its current delay with respect to the timetable, and actions correspond to speed adjustments that must be optimized by taking into account present (fuel) cost and future (delay) costs.

5.3 The model

Consider a round tour with a fixed sequence of port calls and a total planned duration of T time units. A ship sails a *route* consisting of R round tours for a planned duration of RT. Eventually, we let $R \to \infty$ and focus on the long run average costs, which can be obtained by averaging the total route costs over time. Route costs consist in the costs of delayed port arrivals and departures and the costs of (optimally) performing recovery actions such as speed adjustments. Since the total route duration is assumed to be constant, fixed costs can be ignored in the model. The goal is to construct an optimal schedule by dividing the T time units over the round tour in such a way that the long run average costs are minimized.

The tactical planning level

Denote the ports visited in the round tour by $P = \{1, \ldots, |P|\}$. Rounds start in port 1, visit ports $2, 3, \ldots, |P|-1, |P|$ and then return to port 1, after which a new round starts. The route consists of R round tours and $N = R|P|+1$ port calls (including the final port call in port 1). Let $n \in \{1, \ldots, N\}$ index the port calls. The nth port call is made at port $p[n]$. Thus, $p[n] := p$ for $n = p, |P|+p, 2|P|+p, \ldots$, with $p \in P$.

Let t_n^{arr} and t_n^{dep} respectively denote the *planned* arrival and departure time of port call n. The planned arrival time of port call $n+1$ equals the planned departure time of port call n plus the planned sailing time. This planned sailing time consists of the fixed minimum sailing time needed between ports $p[n]$ and $p[n+1]$ (denoted by $t_{p[n]}^s$) and the buffer time included in the sea leg, which is a decision variable that will be denoted by $B_{p[n]}$. Thus $t_{n+1}^{arr} = t_n^{dep} + t_{p[n]}^s + B_{p[n]}$. The requirement of a cyclic schedule means that buffer time and minimum sailing time for a specific sea leg must be the same for each round. (The notation $t_{p[n]}^s$ and $B_{p[n]}$ effectively enforces this requirement, see the definition of $p[n]$.) The planned departure time of the ship for port call n is simply the planned arrival time plus the fixed port time, which will be denoted by $t_{p[n]}^p$. Thus $t_n^{dep} = t_n^{arr} + t_{p[n]}^p$. The results in this paper can be extended to optimize buffers for the ports as well, but we do not include such variables for simplicity and because buffer times in ports are expensive and therefore uncommon.

We can set $t_1^{dep} := 0$ without loss of generality. Then, all planned arrival and departure times for the remaining $R|P|$ port calls follow from the above recursive relations once we fix $B := (B_1, \ldots, B_{|P|}) \in \mathbb{R}^{|P|}$. So finding a schedule consists in fixing B. The requirement that the total planned duration equals T implies that B should satisfy $\sum_{p \in P} B_p = \overline{B}$, where $\overline{B} := T - \sum_{p \in P} t_p^p - \sum_{p \in P} t_p^s$. (We assume $T \geq \sum_{p \in P} t_p^p + \sum_{p \in P} t_p^s$, such that $\overline{B} \geq 0$.)

The operational planning level

While the ship sails the route, unforeseen events cause the ship to be delayed with respect to the planned timetable, i.e. the *planned* arrival and departure times t_n^{arr} and t_n^{dep}. Discussions at a large liner carrier have revealed that both delays in the port and delays during the sea leg are important (cf. Wang and Meng 2012a, p. 616). Therefore, let $X_n^p \geq 0$ and $X_n^s \geq 0$ denote the random delay incurred during port call n, and in the sea leg after port call n, respectively. The random variables X_n^p and X_n^s are assumed to be independent of each other, and of all other random variables, in particular of $X_{n'}^p$ and $X_{n'}^s$ for $n \neq n'$. Distributions are arbitrary, but the random delay in a port in a specific position in the round trip is identically distributed in each round trip. Thus X_n^p and $X_{n'}^p$ are identically distributed if $p[n] = p[n']$. Similarly, X_n^s and $X_{n'}^s$ are identically distributed if $p[n] = p[n']$.

To reduce the delay with respect to the schedule, the liner company can perform two types of recovery actions. *Speed adjustments* during the sea leg are the preferred approach to deal with delays. But in case of excessive delays, *extreme (recovery) actions* in the port are sometimes taken in practice, such as cut-and-go. In cut-and-go, the vessel will stop (un)loading and will immediately leave the port. This is very costly in terms of customer reputation, and consequently, we will attach very high costs to this action. Let τ_n be the difference in the time used to sail from port $p[n]$ to port $p[n+1]$ (excluding unforeseen delays) and the minimum sailing time needed. We will refer to τ_n as the additional sailing time or the sailing time action. Let γ_n denote the time recovered by the extreme recovery action in the nth port, which is taken after the port delay is revealed. Note that τ_n and γ_n are online decision variables, these decisions are taken dynamically in each port and before each sea leg. In contrast, all buffer times B are decided upon before the ship starts sailing the route.

The following recursive relations govern the propagation of the delay during the trip:

$$d_{n+1}^{arr} = (d_n^{dep} + \tau_n - B_{p[n]} + X_n^s)^+, \tag{5.1}$$

$$d_n^{dep} = (d_n^{arr} + X_n^p - \gamma_n)^+. \tag{5.2}$$

where $x^+ = \max\{x, 0\}$. Since ships have to adhere to the berthing plans made by terminal operators, we assume that ships cannot arrive early in a port. And a ship is not allowed to depart earlier than the schedule, because export containers may arrive just in time to be loaded according to the schedule.

Note that only a single sailing time decision is made for each sea leg. Especially when delays at sea occur frequently, it could be better to adapt the sailing time during the leg. This could be achieved by dividing the sea leg into parts. At the end of each part, the planned sailing time for the next part would be determined. The methods and results obtained in this paper can be extended to this more sophisticated model.

Costs

For $p \in P$, let $\mathcal{D}_p^{arr}(d)$ and $\mathcal{D}_p^{dep}(d)$ be respectively the cost of arriving in and departing from port p of the round tour with a delay of d time units with respect to the schedule. We assume that both $\mathcal{D}_p^{arr}(d)$ and $\mathcal{D}_p^{dep}(d)$ are convex and increasing in d. Penalizing the average delay satisfies this assumption and is arguably the most intuitive approach for measuring delays. This latter approach is common (e.g. Kroon et al. 2008, Fischetti et al. 2009), but more general delay cost models have also been proposed (Wang and Meng 2012b).

Let $\mathcal{F}_p(\tau)$ denote the fuel cost incurred between port p and the next port when using a sailing time of $t_p^s + \tau$ time units. $\mathcal{F}_p(\tau)$ is decreasing and convex in τ. Indeed, for economic sailing speeds the bunker consumption rate can be accurately approximated by a constant times the third power of sailing speed (Notteboom and Vernimmen 2009, Brouer et al. 2014a), which implies that $\mathcal{F}_p(\tau)$ is proportional to $\frac{1}{(t_p^s+\tau)^2}$, which is decreasing and convex in τ. (For details see Section 5.6.1.) Furthermore, let $\tau_p^u \geq 0$ be the upper bound on the sailing time action obtained from the minimum sailing speed. Then, $\mathcal{F}_p(\tau)$ is well-defined for all $0 \leq \tau \leq \tau_p^u$. Denote the costs of using the extreme recovery action to reduce the delay by one unit of time by $c^e > 0$.

Remember that the sailed route consists of $N = R|P| + 1$ port calls. Let $\mathcal{C}_{n,N}^{arr}(d; B)$ denote the expected cost of completing the route when arriving for port call n with a delay of d time units. Let $\mathcal{C}_{n,N}^{dep}(d; B)$ denote these costs at the departure of port call n. The dynamics of the problem will be described by recursive relations. The following recursive relation holds for $1 \leq n < N$:

$$\mathcal{C}_{n,N}^{dep}(d_n^{dep}; B) = \mathcal{D}_{p[n]}^{dep}(d_n^{dep}) + \min_{0 \leq \tau \leq \tau_{p[n]}^u} \left\{ \mathcal{F}_{p[n]}(\tau) + \mathcal{K}_n(d_n^{dep} + \tau; B) \right\}, \tag{5.3}$$

$$\text{where} \quad \mathcal{K}_n(d_n^{dep} + \tau; B) := \mathbb{E}_{X_n^s} \left[\mathcal{C}_{n+1,N}^{arr} \left((d_n^{dep} + \tau - B_{p[n]} + X_n^s)^+; B \right) \right]. \tag{5.4}$$

And the following recursive relation holds for $1 < n < N$:

$$\mathcal{C}^{arr}_{n,N}(d^{arr}_n; B) = \mathcal{D}^{arr}_{p[n]}(d^{arr}_n) + \mathbb{E}_{X^p_n}\left[\min_{\gamma\geq 0}\left\{c^e\gamma + \mathcal{C}^{dep}_{n,N}((d^{arr}_n + X^p_n - \gamma)^+; B)\right\}\right]. \quad (5.5)$$

Note that delay propagates according to (5.1) and (5.2). Also, note that the extreme recovery action is taken after the port delay is incurred. For the final arrival in port 1, we have the following:

$$\mathcal{C}^{arr}_{N,N}(d^{arr}_N; B) = \mathcal{D}^{arr}_{p[N]}(d^{arr}_N)$$

So for the final arrival, we only pay the delay costs.

We introduce notation regarding the optimal sailing times and extreme recovery actions. Let $\mathcal{T}_n(d; B)$ denote the optimal sailing time after port call n (on the sea leg towards port call $n+1$) when the departure delay equals d:

$$\mathcal{T}_n(d; B) := \min\left\{\tau' \,|\, \tau' \in \arg\min_{0\leq\tau\leq\tau^u_{p[n]}} \left\{\mathcal{F}_{p[n]}(\tau) + \mathcal{K}_n(d+\tau; B)\right\}\right\}. \quad (5.6)$$

Let $\mathcal{Y}_n(d + X^p_n; B)$ denote the optimal extreme recovery action in port call n when the delay (including port delay) equals $d + X^p_n$:

$$\mathcal{Y}_n(d + X^p_n; B) := \max\left\{\gamma' \,|\, \gamma' \in \arg\min_{\gamma\geq 0} \left\{c^e\gamma + \mathcal{C}^{dep}_{n,N}((d + X^p_n - \gamma)^+; B)\right\}\right\}. \quad (5.7)$$

So as a tie-breaking rule, we use minimization of the delay in the next port.

The long run average costs

Because the buffers are transformed into a timetable, which is operated for many rounds, we adopt the long run average costs as performance criterion, which will be denoted by $\mathcal{C}^*(B)$. We have:

$$\mathcal{C}^*(B) := \lim_{R\to\infty} \frac{\mathcal{C}^{dep}_{1,R|P|+1}(d^{dep}_1; B)}{R}, \qquad \forall B \in \mathcal{B}, \quad (5.8)$$

where $\mathcal{B} = \left\{B \in \mathbb{R}^{|P|}_{\geq 0} \,\middle|\, \sum_{p\in P} B_p = \overline{B}\right\}$ is the set of feasible buffers. For now, we assume that the limit on the RHS exists, and that it is independent of d^{dep}_1. Later, in Corollary 1, we will formally prove the existence of the limit, and that it is independent of d^{dep}_1, under certain sufficient conditions.

We consider the following optimization problem:

$$\mathcal{C}^* = \min_{B \in \mathcal{B}} \mathcal{C}^*(B). \tag{5.9}$$

This problem is non-standard. Each buffer allocation $B \in \mathcal{B}$ yields a SDP whose optimal long term average costs equal $\mathcal{C}^*(B)$. But the buffer time variables themselves cannot be accommodated for in the SDP because they are one-time decisions that affect multiple states. Note that the sailing speed decisions are part of the SDP, so the problem jointly optimizes the buffer allocation and the sailing speed decisions.

5.3.1 Assumptions for computational purposes

In general, solving for the optimal costs $\mathcal{C}^*(B)$ of the SDP that arises for *fixed* buffers B is already computationally intractable. This is because the SDP has a continuous state space because delay is continuous. (Apart from the current delay, the state consists of the current port $p[n]$ and whether we are arriving or departing.) To deal with this computational issue, we will assume discrete delays and piecewise linear fuel costs (see also Wang and Meng 2012a, who use a similar approach in their model). Specifically, we assume that there exists an appropriate basic time unit $l > 0$. We let $\{zl | z \in \mathbb{Z}^+\}$ be the set of integer multiples of l, and make the following assumption.

Assumption 4 (Discrete model primitives)**.** *The delays X^s and X^p take on values in $\{zl | z \in \mathbb{Z}^+\}$. The total buffer $\overline{B}$ and the maximum additional sailing time τ_p^u are in $\{zl | z \in \mathbb{Z}^+\}$. The functions $\mathcal{D}_p^{arr}(\cdot)$, $\mathcal{D}_p^{dep}(\cdot)$ and $\mathcal{F}_p(\cdot)$ are piecewise linear functions, with breakpoints on $\{zl | z \in \mathbb{Z}^+\}$. The initial delay is in $\{zl | z \in \mathbb{Z}^+\}$.*

If we encounter large sea and port delays repeatedly, the delay with respect to the schedule may grow arbitrarily large. In practice, it seems reasonable to assume that when delay exceeds some (possibly large) threshold, it will be optimal to perform the extreme recovery action. We therefore make this assumption to simply and straightforwardly bound the maximum delay. For ease of exposition, we will also assume that the random sea and port delays are bounded by some arbitrary number. These assumptions will simplify the computation of the optimal costs associated with a buffer $B \in \mathcal{B}$.

Assumption 5 (Bounded delays)**.** *For each $p \in P$, there exists a delay $d_p^{max} < \infty$ such that $\mathcal{D}_p^{dep}(d) - c^e d$ is monotonically increasing $\forall d > d_p^{max}$. There exist $X_{p[n]}^{s,max}, X_{p[n]}^{p,max} \in \{zl | z \in \mathbb{Z}^+\}$ such that $\forall n : \mathbb{P}(X_n^s > X_{p[n]}^{s,max}) = 0$, $\mathbb{P}(X_n^p > X_{p[n]}^{p,max}) = 0$.*

These assumptions are not restrictive in practice as $d_p^{max}, X_p^{\mathrm{s,max}}$ and $X_p^{\mathrm{p,max}}$ can be taken to be large (e.g. one or more weeks when operating a weekly schedule).

5.4 Theoretical insights

In this section we will derive various theoretical insights into the problem. Results in this section hold for the general model presented in Section 5.3. Assumptions 4 and 5, which are made for computational purposes only, are not needed to obtain the results in this section.

Our first result verifies that more delay is worse than less delay. (All proofs can be found in the appendix.)

Lemma 3. *The functions* $\mathcal{C}^{dep}_{n,N}(d;B)$ *and* $\mathcal{C}^{arr}_{n+1,N}(d;B)$ *are nondecreasing in the amount of delay* d *for* $1 \leq n < N$ *and* $B \in \mathcal{B}$.

The following result is more surprising, since the costs are not separable because delays may propagate from port call to port call.

Lemma 4. *The functions* $\mathcal{C}^{dep}_{n,N}(d;B)$ *and* $\mathcal{C}^{arr}_{n+1,N}(d;B)$ *are joint convex in* d *and* B *for* $1 \leq n < N$.

A direct result of Lemma 4 is that the average cost per period $\mathcal{C}^*(B)$ is also convex in B.

Theorem 5. *The optimal long term average cost* $\mathcal{C}^*(B)$ *is joint convex in* $B \in \mathcal{B}$*, provided that* $\mathcal{C}^*(B)$ *exists in* $\mathbb{R}$ *for* $B \in \mathcal{B}$.

This result will be used later to find the optimal buffer B, and thus the optimal schedule. (As for the condition: Corollary 1 proves the existence of the optimal long term average costs $\mathcal{C}^*(B)$ in $\mathbb{R}$ under sufficient conditions, namely Assumptions 4 and 5.)

The following results give some more insight into how the sailing times and extreme actions should depend on the current delay. The following lemma shows that the larger the delay, the more action should be taken.

Lemma 6.

(a) The optimal sailing time action $\mathcal{T}_n(d^{dep}_n;B)$ *between two ports is nonincreasing in the departure delay* d^{dep}_n *for* $1 \leq n < N$ *and* $B \in \mathcal{B}$;

(b) the optimal extreme recovery action $\mathcal{Y}_n(d^{arr}_n + X^p_n;B)$ *in a port is nondecreasing in the amount of delay* $d^{arr}_n + X^p_n$ *before that action for* $1 < n < N$ *and* $B \in \mathcal{B}$.

Thus, a ship with larger departure delay should sail faster than a ship with smaller delay. We wonder whether it could even be optimal for the first ship to plan to "overtake" the latter ship. The following lemma answers this question, by proving that this can never be optimal.

Lemma 7.

(a) The optimal arrival delay $d_{n+1}^{arr} = (d_n^{dep} + \mathcal{T}_n(d_n^{dep}; B) - B_{p[n]} + X_n^s)^+$ *is stochastically nondecreasing in the departure delay* d_n^{dep}*, for* $1 \leq n < N$ *and* $B \in \mathcal{B}$*;*

(b) the optimal departure delay $d_n^{dep} = (d_n^{arr} + X_n^p - \mathcal{Y}_n(d_n^{arr} + X_n^p; B))^+$ *is nondecreasing in the delay* $d_n^{arr} + X_n^p$ *after incurring port delay, for* $1 < n < N$ *and* $B \in \mathcal{B}$*.*

This lemma thus effectively bounds the maximal decrease in sailing time (and thus the increase in speed) that should result from being more delayed.

5.5 Solution Approach

Our objective is finding a $B \in \mathcal{B}$ that minimizes $\mathcal{C}^*(B)$. Since $\mathcal{C}^*(B)$ is convex by Theorem 5, a range of optimal subgradient-based algorithms is at our disposal for this problem, provided we can compute *subgradients* of $\mathcal{C}^*(B)$ at arbitrary B. We discuss this in-depth in Section 5.5.1, and provide a simple algorithm that works well computationally for our problem. The novelty of our algorithm lies in developing an approach for computing subgradients of $\mathcal{C}^*(B)$, which is discussed in Section 5.5.2.

5.5.1 Subgradient-based algorithms

Theorem 5 implies that $\mathcal{C}^*(B)$ is convex. Thus for each $B \in \mathcal{B}$, there exists a $g = (g_1, \ldots, g_{|P|})$ that satisfies the subgradient inequality:

$$\forall B' \in \mathcal{B} : \mathcal{C}^*(B') \geq \mathcal{C}^*(B) + \sum_{p \in P} g_p(B'_p - B_p). \tag{5.10}$$

We now first sketch how subgradients can be used in an efficient optimization algorithm.

Our algorithm iteratively generates subgradients using the method described in Section 5.5.2. In the ith iteration, the subgradient at B^i is computed. Denote it by $g^i = (g_1^i, \ldots, g_{|P|}^i)$, and denote $g_0^i = \mathcal{C}^*(B^i) - \sum_{p \in P} g_p^i B_p^i$. After iteration I, we have the following problem:

$$\min \quad z \tag{5.11}$$

$$\text{s.t.} \qquad z \geq \sum_{p \in P} g_p^i B_p + g_0^i \qquad i \in \{1, \ldots, I\} \tag{5.12}$$

$$\sum_{p \in P} B_p = \overline{B} \tag{5.13}$$

$$B_p \geq 0 \qquad p \in P. \tag{5.14}$$

The inequalities (5.12) ensure that z satisfies the inequality imposed by the subgradients, see (5.10). Together, (5.13) and (5.14) ensure that $B = (B_1, \dots, B_{|P|}) \in \mathcal{B}$. For any subgradients $g_1, \dots, g_I$, the optimal z^* of (5.11-5.14) satisfies $z^* \leq \mathcal{C}^* = \mathcal{C}^*(B^*)$. Indeed, $\forall B$, z^* must become $\max_{i \in \{1,\dots,I\}} \left\{ \sum_{p \in P} g_p^i B_p + g_0^i \right\}$, which cannot exceed $\mathcal{C}^*(B^*)$ by (5.10).

Algorithm 10 explains how we use this formulation in our optimization approach:

Algorithm 10: Solution algorithm

1. Initialize $i = 1$, $B^1 = (B_1^1, \dots, B_{|P|}^1)$ with $B_p^1 = \frac{\overline{B}}{|P|}$, $UB = \infty$ and $LB = -\infty$.
2. Compute $\mathcal{C}^*(B^i)$ and the gradient g^i at B^i (see Section 5.5.2).
3. If $\mathcal{C}^*(B^i) < UB$, set $UB = \mathcal{C}^*(B^i)$ and $B^{UB} = B^i$.
4. Let (z^*, B') denote the optimal solution of (5.11-5.14) for $I = i$. Set $LB = z^*$, $B^{i+1} = B'$.
5. If $UB - LB \leq \epsilon$, designate B^{UB} as ϵ-optimal and terminate. Otherwise, set $i \leftarrow i + 1$ and go to Step 2.

In initial steps, the B^{i+1} from Step 4 may lie far away from the last search point B^i, adversely impacting performance. Therefore, we limit the distance between B^i and $B^{i+1} = B'$. Consider the constraints,

$$\sum_{p \in P} |B_p - B_p^i| \leq w^{max}, \qquad \forall p \in P : |B_p - B_p^i| \leq w_p^{max} \tag{5.15}$$

where w^{max} and $\forall p \in P : w_p^{max}$ are parameters. Then Step 4 is replaced by the following in the first 25 iterations of the algorithm:

4' Let (z^*, B') denote the optimal solution of (5.11-5.14)+(5.15) for $I = i$. Set $B^{i+1} = B'$. Let $\tilde{z}^*$ be the optimal solution value of (5.11-5.14). Set $LB = \tilde{z}^*$.

5.5.2 Subgradients

In general, computing subgradients involves analyzing the change of the objective function when the input changes. For our problem, changing B affects the *structure* of the SDP underlying $\mathcal{C}^*(B)$, which complicates the computation of the subgradient. In this section, we first analyze this structure. This analysis involves a number of complex ideas and quite some additional notation, but it yields a relatively simple algorithm for computing subgradients that we present later in this section.

Analysis of SDP structure

Throughout this section, we work with the specialized model that is obtained by imposing discrete model primitives (Assumption 4) and bounded delays (Assumption 5). For a detailed discussion of these assumptions we refer to Section 5.3.1. As we need subgradients for general B, we must also consider cases where B_p is *fractional*, where a quantity will be referred to as fractional if it cannot be written as zl with $z \in \mathbb{Z}$ and l the basic time unit in Assumption 4. A quantity that can be written as zl for some $z \in \mathbb{Z}$ will be referred to as *discrete*.

For notational convenience, we transform the buffers. Each $B \in \mathcal{B}$ corresponds to a cumulative buffer allocation $\tilde{B}$, by setting $\tilde{B}_p := \sum_{p'=1}^{p-1} B_{p'}$ and $\tilde{B} = (\tilde{B}_1, \dots, \tilde{B}_{|P|})$. (Thus $\tilde{B}_1 := 0$.) Let $\tilde{\mathcal{B}}$ contain every $\tilde{B}$ that can be obtained in this fashion from a $B \in \mathcal{B}$. Thus $\forall \tilde{B} \in \tilde{\mathcal{B}} : \tilde{B}_{|P|} \leq \overline{B}$ and $\forall \tilde{B} \in \tilde{\mathcal{B}}, \forall p \in P : \tilde{B}_{p+1} \geq \tilde{B}_p$. For $\tilde{B} \in \tilde{\mathcal{B}}$, define $\mathcal{C}^*(\tilde{B}) := \mathcal{C}^*(B)$, with B obtained by setting $B_p = \tilde{B}_{p+1} - \tilde{B}_p$ for $p \in \{1, \dots, |P|-1\}$, and $B_{|P|} = \overline{B} - \tilde{B}_{|P|}$.

For each port call $n \in \{1, \dots, N\}$, define:

$$Q_n(p) := Q_{p[n]}(p) := \left\{ zl + \tilde{B}_p - \tilde{B}_{p[n]} | z \in \mathbb{Z} \right\}, \tag{5.16}$$

and let $Q_n := \bigcup_{p \in P} Q_n(p)$. In the absence of fractional actions and/or waiting for departure or arrival, $Q_n(p)$ contains precisely the delays for port call n that result in a discrete delay in port calls to port p. Thus, Q_n contains delays that result in a discrete delay in some future port call. The following lemma shows that Q_n contains all delays that may occur for port call n. The lemma is a consequence of Assumption 4 and the recursive relations (5.3) and (5.5).

Lemma 8. *Fix $\tilde{B} \in \tilde{\mathcal{B}}$, and choose τ_n and γ_n optimally using* (5.6) *and* (5.7) *to break ties. Then for each port call n: $d_n^{dep} \in Q_n$, $d_n^{dep} + \tau_n \in Q_n$, $d_n^{arr} + X_n^p \in Q_n$, $d_n^{arr} + X_n^p - \gamma_n \in Q_n$.*

We next introduce notation for states and policies of the SDP based on Lemma 8. A generic state will be denoted by s. Let $s_{p,\text{dep}}[z, p'; \tilde{B}]$ correspond to departing from port p with delay $d^{dep} = zl + \tilde{B}_{p'} - \tilde{B}_p$. Let $s_{p,\text{port}}[z, p'; \tilde{B}]$ correspond to being in port p with a delay of $d^{arr} + X^p = zl + \tilde{B}_{p'} - \tilde{B}_p$, *after* incurring port delay. Let

$$S_{\tilde{B}} := \left\{ s_{p,u}[z, p'; \tilde{B}] | z \in \mathbb{Z}, p \in P, p' \in P, u \in \{\text{dep}, \text{port}\} \right\}.$$

By Lemma 8, all combinations of states and delays that can occur for $\tilde{B} \in \tilde{\mathcal{B}}$ are in $S_{\tilde{B}}$, though $S_{\tilde{B}}$ also contains states that cannot occur because their associated delay is negative. It is immediate from Lemma 8 that the optimal actions τ and $-\gamma$ in any state $s_{p,u}[z, p'; \tilde{B}]$ must take their values in $Q_{p'}(p'')$ for some p''. We will denote a generic action by a. In a state with $u=$ dep, the action $a_{p'}[z,p'';\tilde{B}]$ will denote $\tau = zl + \tilde{B}_{p''} - \tilde{B}_{p'}$, and in a state with $u=$ port, action $a_{p'}[z,p'';\tilde{B}]$ will denote $-\gamma = zl + \tilde{B}_{p''} - \tilde{B}_{p'}$. Let $A_{\tilde{B}} = \{a_{p'}[z,p'';\tilde{B}] | z \in \mathbb{Z}, p' \in P, p'' \in P\}$.

Since delay is non-negative, and bounded above by Assumption 5, for each $\tilde{B}$ a finite subset of $S_{\tilde{B}}$ and $A_{\tilde{B}}$ suffices for a complete description of the model. As a consequence, we have the following result.

Corollary 1. *For all $B \in \mathcal{B}$, the limit $\lim_{R\to\infty} \mathcal{C}^{dep}_{1,R|P|+1}(d_1^{dep};B)/R$ exists in $\mathbb{R}$ and is independent of d_1^{dep}. There exists a stationary deterministic policy that is average cost optimal.*

A stationary deterministic policy for $\tilde{B} \in \tilde{\mathcal{B}}$ will be represented by a function $\Pi_{\tilde{B}} : S_{\tilde{B}} \to A_{\tilde{B}}$ and we denote the optimal stationary deterministic policy for $\tilde{B}$ by $\Pi^*_{\tilde{B}}$.

We will now investigate the change of $\mathcal{C}^*(\tilde{B})$ when $\tilde{B}$ changes. First some preliminaries. A cumulative buffer $\tilde{B} \in \tilde{\mathcal{B}}$ is *completely fractional* if for every $p, p' \in P$ with $p \neq p'$, the number $\tilde{B}_{p'} - \tilde{B}_p$ cannot be written as zl for $z \in \mathbb{Z}$. Define for every completely fractional $\tilde{B} \in \tilde{\mathcal{B}}$ the following subset of $\tilde{\mathcal{B}}$:

$$\Delta(\tilde{B}) := \left\{ \tilde{B}' \in \tilde{\mathcal{B}} \middle| \forall p, p' \in P : l \left\lfloor (\tilde{B}_{p'} - \tilde{B}_p)/l \right\rfloor \leq \tilde{B}'_{p'} - \tilde{B}'_p \leq l \left\lfloor (\tilde{B}_{p'} - \tilde{B}_p)/l \right\rfloor + l \right\}. \tag{5.17}$$

Note that $\tilde{B} \in \Delta(\tilde{B})$. We are now ready to formulate the main result of this section.

Theorem 9. *Take any completely fractional $\tilde{B} \in \tilde{\mathcal{B}}$ and let $\Pi^*_{\tilde{B}}$ denote its average cost optimal policy. For all $\tilde{B}' \in \Delta(\tilde{B})$, define the policy $\hat{\Pi}_{\tilde{B}'}$ as follows:*

$$\hat{\Pi}_{\tilde{B}'}(s_{p,u}[z,p';\tilde{B}']) = a_{p'}[z',p'';\tilde{B}'] \quad \textit{iff} \quad \Pi^*_{\tilde{B}}(s_{p,u}[z,p';\tilde{B}]) = a_{p'}[z',p'';\tilde{B}].$$

Let $\hat{\mathcal{C}}(\tilde{B}')$ denote the long run average costs for $\tilde{B}'$ under $\hat{\Pi}_{\tilde{B}'}$. Then

$$\forall \tilde{B}' \in \Delta(\tilde{B}) : \hat{\mathcal{C}}(\tilde{B}') = \mathcal{C}^*(\tilde{B}') = \mathcal{C}^*(\tilde{B}) + \sum_{p\in P} g_p(\tilde{B}'_p - \tilde{B}_p),$$

where $g = (g_1, \ldots, g_{|P|})$ is a subgradient at $\tilde{B}$.

It is surprising that $\hat{\Pi}_{\tilde{B}'}$ is optimal for $\tilde{B}'$, because $\hat{\Pi}_{\tilde{B}'}$ is rather different from $\Pi^*_{\tilde{B}}$: $s_{p,u}[z,p';\tilde{B}]$ and $s_{p,u}[z,p';\tilde{B}']$ represent different delays, and $a_p[z,p';\tilde{B}]$ and $a_p[z,p';\tilde{B}']$ represent different actions. The proof of Theorem 9 shows that, when states are expressed as $s_{p,u}[z,p';\tilde{B}']$, the transitions are independent of $\tilde{B}'$, and the theorem follows from that result and convexity.

Theorem 9 implies the following.

Corollary 2. *The gradient g at $\tilde{B}$ from Theorem 9 is a subgradient for any $\tilde{B}' \in \Delta(\tilde{B})$.*

To arrive at a simple algorithm to compute subgradients, we investigate $\Delta(\tilde{B})$. For any $\tilde{B} \in \tilde{\mathcal{B}}$, it holds that $0 \leq \tilde{B}_p \leq \overline{B}$. By Assumption 4, define $z_{\overline{B}} = \overline{B}/l \in \mathbb{Z}$. Thus, we can always write $\tilde{B}_p = lz_p + lx_p$, with $z_p \in \{0, 1, \ldots, z_{\overline{B}} - 1\}$ and $0 \leq x_p \leq 1$. For completely fractional $\tilde{B} \in \tilde{\mathcal{B}}$, this decomposition into lz_p and lx_p is unique.

Lemma 10. *For any completely fractional $\tilde{B} \in \tilde{\mathcal{B}}$, write $\tilde{B}_p = l(z_p + x_p)$ with $z_p \in \{0, 1, \ldots, z_{\overline{B}} - 1\}$ and $0 \leq x_p < 1$. Let $f : P \to P$ be the unique permutation of P such that $p > p' \to x_{f(p)} > x_{f(p')}$. Then $\Delta(\tilde{B})$ contains precisely all $\tilde{B}' = (\tilde{B}'_1, \ldots, \tilde{B}'_{|P|}) \in \tilde{\mathcal{B}}$ such that $\tilde{B}'_p = lz'_p + lx'_p$, with $z'_p = z_p$ and $0 \leq x'_{f(1)} \leq x'_{f(2)} \leq \ldots \leq x'_{f(|P|)} \leq 1$.*

Computing subgradients

The above analysis yields Algorithm 11 to compute a subgradient for any $B \in \mathcal{B}$: We

Algorithm 11: Computing subgradients

1. Let $\tilde{B} \in \tilde{\mathcal{B}}$ be the cumulative buffer corresponding to B.
2. For all $p \in P$, write $\tilde{B}_p = lz_p + lx_p$, with $z_p \in \{0, 1, \ldots, z_{\overline{B}} - 1\}$ and $0 \leq x_p \leq 1$. Let $f : P \to P$ be any permutation of P such that $f(1) = 1$ and $\forall p, p' \in P$: $p > p' \to x_{f(p)} \geq x_{f(p')}$.
3. For each $i \in \{1, \ldots, |P|\}$ define $\tilde{B}^i = (\tilde{B}^i_1, \ldots, \tilde{B}^i_{|P|}) \in \tilde{\mathcal{B}}$,
$$\tilde{B}^i_p = \begin{cases} lz_p & \text{if } f^{-1}(p) \leq i \\ l(z_p + 1) & \text{if } f^{-1}(p) > i. \end{cases}$$
for $p \in \{1, \ldots, |P|\}$. Then $\exists \tilde{B}' \in \tilde{B}$ such that $\tilde{B} \in \Delta(\tilde{B}')$, and $\tilde{B}^i \in \Delta(\tilde{B}')$ for each $i \in \{1, \ldots, |P|\}$.
4. Compute $\mathcal{C}^*(\tilde{B}^i)$ for each $i \in \{1, \ldots, |P|\}$.
5. By Theorem 9 and Corollary 2, since $\tilde{B} \in \Delta(\tilde{B}')$, and $\tilde{B}^i \in \Delta(\tilde{B}')$ for each $i \in \{1, \ldots, |P|\}$, a subgradient g at $\tilde{B}$ satisfies the system of equations:
$$\mathcal{C}^*(\tilde{B}^i) = \mathcal{C}^*(\tilde{B}) + \sum_{p \in P} g_p(\tilde{B}^i_p - \tilde{B}_p), i \in \{1, \ldots, |P|\} \tag{5.18}$$
Solve the system to obtain $\mathcal{C}^*(\tilde{B})$ and a subgradient $g = (g_1, \ldots, g_{|P|})$ at $\tilde{B}$.
6. Use this subgradient to obtain a subgradient for $\mathcal{C}^*(B)$ at B.

now explain some details. For Step 2, note that $\tilde{B}_1 := 0$ such that $x_1 = 0$, implying that

$f(1) = 1$ never contradicts the other requirements on f. The $\tilde{B}'$ in Step 3 can be any completely fractional $\tilde{B}'$ with $\tilde{B}'_p = l(z'_p + x'_p)$, such that $\forall p \in P : z'_p = z_p$ and $\forall p, p' \in P : p > p' \to x'_{f(p)} > x'_{f(p')}$. (If $\tilde{B}$ is completely fractional, it suffices to set $\tilde{B}' = \tilde{B}$.) It can then be verified using Lemma 10 that $\tilde{B} \in \Delta(\tilde{B}')$, and $\forall i \in \{1, \ldots, |P|\} : \tilde{B}^i \in \Delta(\tilde{B}')$. (In fact, it can be shown that $\Delta(\tilde{B}')$ is the convex hull of $\{\tilde{B}^i | i \in \{1, \ldots, |P|\}$, see Lemma 12 in the appendix.) Hence, the optimal solution returned by our algorithm will contain an integer buffer time allocation.

For Step 4, note that $\mathcal{C}^*(\tilde{B}^i)$ is the long run average cost of a finite state SDP, which can be solved efficiently using linear programming. The specific choice of $\tilde{B}^i$ reduces the complexity of finding $\mathcal{C}^*(\tilde{B}^i)$, because only integral multiples of l occur for these buffers, see Lemma 8. For Step 5, note that g_1 is free in (5.18), since $\tilde{B}_1 := 0$ for all $\tilde{B} \in \tilde{\mathcal{B}}$. For the same reason, the value of g_1 is inconsequential, so set it to 0. Since $\tilde{B}^i$ are linearly independent by construction, (5.18) has $|P|$ linearly independent equations, leading to a unique solution for the variables $g_2, \ldots, g_{|P|}$ and $\mathcal{C}^*(\tilde{B})$. Step 6 is straightforward, since $\tilde{B}$ is obtained from B using a linear transformation.

5.6 Case Study

5.6.1 Data

To test our method, we use the ME1 route in September 2012 of the Maersk Line network. Time is discretized in units of four hours. Table 5.1 shows the order in which the ports are visited in the route, the distances and sailing times between ports and the time needed in the port. The second column of Table 5.1 denotes the total time planned in the port to load and unload the ship. In the third column the distances between the ports in nautical miles are presented. Distances are obtained from SeaRates (2015). The distance shown for each port is the distance that the ship has to cover to sail from that port to the next port. The fourth column shows the sailing time in hours according to the schedule. The planned sailing time for Antwerp is 32 hours, which means that a ship might take 32 hours to sail from Antwerp to Bremerhaven before it will encounter a delay during its trip. The last column shows the buffer time in the current schedule assuming that the route is sailed at maximum speed. The time needed to make one full round tour is 1,176 hours (7 weeks).

We assume that the route is sailed using a post panamax ship with capacity 8,400 TEU, using data from Brouer et al. (2014a). The minimum and maximum speed of this ship are 12 and 23 knots respectively. Bunker consumption per time unit can be accurately

Port	Port time (hr)	Distance (nmi)	Sailing time (hr)	Buffer time (hr)
Jebel Ali	31	1,329	72	12
Jawaharlal Nehru	33	443	24	4
Mundra	16	1,122	56	4
Salalah	14	1,553	68	0
Jeddah	11	778	36	0
Suez Canal	16	2,283	100	0
Algeciras	18	1,476	88	20
Felixstowe	24	156	16	8
Antwerp	16	366	32	16
Bremerhaven	24	283	24	8
Rotterdam	20	3,829	192	24
Suez Canal	22	395	20	0
Aqaba	20	656	40	8
Jeddah	19	2,648	124	8

Table 5.1: Characteristics of the route

approximated as a constant times the third power of speed. Thus, the fuel cost function becomes:

$$\mathcal{F}_p(\tau) = \tilde{f}e\left(\frac{v}{\tilde{v}}\right)^3 (t_p^s + \tau)\frac{24}{l} = \tilde{f}e(t_p^s + \tau)\frac{24}{l}\left(\frac{\delta_p}{(t^s + \tau)l\tilde{v}}\right)^3,$$

where v is the sailing speed in knots (nmi/hour) and δ_p is distance in nmi from port p to the next port. The ship has a design speed of $\tilde{v} = 16.5$ knots, and bunker consumption at design speed is $\tilde{f} = 82.2$ ton per day. Bunker cost is assumed to be $e = 600$ USD per ton (Brouer et al. 2014a).

5.6.2 Test instances

Given the fixed port times and the total duration of a round tour, 52 time units remain to allocate over the ports. By changing the additional delay distributions, different scenarios can be constructed. Since we do not know the actual delay distribution, we will gauge the outcomes under different delay distributions. We will assume that each $X_n^s \sim U\left(0, a + \left\lfloor \frac{\delta_{p[n]}}{b} \right\rfloor\right)$, where a and b are instance specific parameters and $\delta_{p[n]}$ is the distance between the current and the next port. For each test instance we can compute the minimum average time to complete one round tour of the route. This time is obtained by sailing at maximum speed and incurring the average delay in each port. The minimum average additional time to complete a round tour should not exceed the available time of 52 time units, since ships will not be able to recover from incurred delays in these scenarios. We will refer to the (positive) difference between the available time and the minimum average completion time as the expected buffer time. Ten instances are constructed by varying the expected buffer time between 2.5 and 25 time units in steps of 2.5 time units. This is done using $a = \{4, 4, 3, 3, 3, 2, 2, 2, 1, 1\}$ and $b = \{300, 400, 200, 325, 500, 225, 332, 600, 250, 350\}$. The extreme expedite cost is given by ten million USD per time unit and $\mathcal{D}_p(d) = 10,000d$ for $0 \leq d \leq d_p^{max} = 42$ time units (one week) for $p \in P$ and we assume that the unit costs are larger than ten million USD for $d > d_p^{max}$ for $p \in P$, such that delays are bounded by 42 time units. Finally, $w_p^{max} = 0.25$ for $p \in P$, $w^{max} = 2$, and $\epsilon = 10^{-8}$.

5.6.3 Results

For each test instance, we first calculate the cost of the schedule when we consider deterministic delays. That is, we assume that the delay incurred between each two ports is fixed and equal to the expected delay between those two ports. The optimal schedule is then found by allocating the available buffer time in such a way that a constant speed is used over the round tour. Furthermore, we calculate the costs of the initial schedule (initial schedule), the costs of the schedule in which the buffer time is uniformly distributed over the ports (uniform schedule) and the costs of the optimal schedule (optimal schedule). All linear programming models are solved using CPLEX 12.6.

Available buffer (time units)	Expected delay (time units)	Expected buffer (time units)
52.0	49.5	2.5
52.0	47.0	5.0
52.0	44.5	7.5
52.0	42.0	10.0
52.0	39.5	12.5
52.0	37.0	15.0
52.0	34.5	17.5
52.0	32.0	20.0
52.0	29.5	22.5
52.0	27.0	25.0

Table 5.2: Characteristics of the ten instances

Table 5.2 shows the expected delay and expected buffer times in time units for the ten instances. The expected buffer times vary between 2.5 and 25 time units.

Expected buffer (time units)	Deterministic schedule (million USD)	Cost of uncertainty Initial (million USD)	Uniform (million USD)	Optimal (million USD)
2.5	3.831	0.904 (100%)	0.709 (78%)	0.702 (78%)
5.0	3.735	0.537 (100%)	0.381 (71%)	0.377 (70%)
7.5	3.644	0.446 (100%)	0.315 (71%)	0.292 (66%)
10.0	3.557	0.369 (100%)	0.257 (70%)	0.234 (63%)
12.5	3.471	0.317 (100%)	0.216 (68%)	0.193 (61%)
15.0	3.387	0.312 (100%)	0.225 (72%)	0.172 (55%)
17.5	3.306	0.272 (100%)	0.190 (70%)	0.137 (50%)
20.0	3.229	0.242 (100%)	0.173 (72%)	0.099 (41%)
22.5	3.153	0.256 (100%)	0.182 (71%)	0.068 (26%)
25.0	3.080	0.222 (100%)	0.158 (71%)	0.040 (18%)

Table 5.3: Total average round tour costs for the ten test instances

Table 5.3 shows the average expected round tour costs for the ten instances. Clearly, the costs of sailing a round tour decreases when the available buffer time increases. The deterministic schedule provides a lower bound on the optimal cost schedule. The difference between the cost of the deterministic and stochastic schedules is the effect of uncertainty on the cost, which is shown in the last three columns of the table. In these columns first the absolute cost of uncertainty is given and in between brackets the relative difference compared to the initial schedule is given. We observe that for high expected buffer times, a large part of the cost is already incurred in the deterministic case. Furthermore, the absolute difference in cost between the initial and the uniform schedule decreases when more buffer time is available, while the absolute difference in cost between the uniform and the optimal schedule increases when more time is available. From the relative costs, we can conclude that the uniform schedule always performs about 30% better than the initial schedule, while the optimal schedule has costs that are $22 - 82\%$ lower than the initial schedule. The relative performance of the optimal schedule increases when more buffer time is available. When only 2.5 time units of buffer time are available, the largest absolute cost reduction between the initial and the optimal schedule can be obtained, while the lowest absolute reduction is obtained for 12.5 time units of expected buffer. The lowest and largest reductions are respectively 123 and 202 thousand USD per round tour can be obtained. Since liner companies usually provide weekly services, this would result in cost reductions of 6-10 million USD per year.

Table 5.4 shows the solution times of the subgradient algorithm. Furthermore, the number of generated subgradients are shown. All instances can be solved to optimality

Expected buffer (time units)	Time (seconds)	Number subgradients
2.5	21	19
5.0	65	26
7.5	77	30
10.0	56	20
12.5	63	23
15.0	64	38
17.5	62	37
20.0	57	34
22.5	64	34
25.0	65	34

Table 5.4: Solution times for the ten test instances

within 80 seconds. In total, 19-38 subgradients have to be determined in the solution algorithm.

Port	On time prob	Avg arr delay (time units)	Distance (nmi)	Buffer time (time units)
Jebel Ali	0.49	0.78	1,329	2
Jawaharlal Nehru	0.49	0.68	443	1
Mundra	0.41	0.68	1,122	1
Salalah	0.30	1.09	1,553	3
Jeddah	0.43	0.80	778	1
Suez Canal	0.38	0.73	2,283	3
Algeciras	0.35	1.21	1,476	3
Felixstowe	0.62	0.44	156	0
Antwerp	0.31	0.94	366	1
Bremerhaven	0.39	0.74	283	1
Rotterdam	0.63	0.44	3,829	6
Suez Canal	0.50	1.00	395	1
Aqaba	0.33	1.00	656	1
Jeddah	0.37	0.84	2,648	4

Table 5.5: Optimal buffer time on the next sea leg

Table 5.5 shows for each port the probability of arriving on time, the average arrival delay in time units and the optimal buffer allocation in time units for the instance with an expected buffer of 15 time units. In general, more buffer time is added to sea legs with larger distances, because on these legs larger additional delays are expected to be incurred.

Port	Delay in time units 0	1	2	3	4	5	≥ 6	Feasible range
Jebel Ali	16	16	15	15	15	15	15	[15, 27]
Jawaharlal Nehru	6	5	5	5	5	5	5	[5, 9]
Mundra	14	13	13	13	13	13	13	[13, 23]
Salalah	19	19	18	17	17	17	17	[17, 32]
Jeddah	10	9	9	9	9	9	9	[9, 16]
Suez Canal	27	27	26	25	25	25	25	[25, 47]
Algeciras	19	18	17	17	17	17	17	[17, 30]
Felixstowe	2	2	2	2	2	2	2	[2, 3]
Antwerp	5	4	4	4	4	4	4	[4, 7]
Bremerhaven	4	4	4	4	4	4	4	[4, 5]
Rotterdam	46	45	44	44	43	42	42	[42, 79]
Suez Canal	6	5	5	5	5	5	5	[5, 8]
Aqaba	9	8	8	8	8	8	8	[8, 13]
Jeddah	32	31	30	30	29	29	29	[29, 55]

Table 5.6: Sailing time action in time units to be used on the next sea leg

Table 5.6 shows the sailing times in time units that will be used on the next sea leg given a certain amount of delay for the instance with ten time units of expected buffer. The last column shows the range of feasible speeds for the given sea leg. The table shows that ships will not always speed up when a larger delay is incurred even when the maximum sailing speed limit is not reached yet (see for example a departure from Jebel Ali with 0 and 1 time units of delay). This confirms that the optimal sailing speed policy is not always to try to recover from all delays during the coming sea leg, as is usually assumed in the literature (e.g. Wang and Meng 2012b). Furthermore, the table shows that Lemmas 6 and 7 are indeed satisfied: ships will never slow down when they incur higher delays, but will also always arrive with at least the same amount of delay in the next port as when they would have incurred a lower delay. Finally, when we consider the amount of extreme actions in the solutions, we observe that more extreme actions are taken when less buffer time is available. In the instances with 15 or more units of buffer, no extreme actions are taken in the optimal solutions. Furthermore, in the instance with 2.5 units of buffer most extreme actions are taken, namely in expectation 0.00038 time units per round tour, which corresponds to once every 353 years.

5.7 Conclusion and Future Research

We developed a new approach for allocating buffers in timetables. Our model jointly optimizes decisions over two stages: buffer times during timetable development and speed optimization during timetable execution. We model the execution of the timetable as a stochastic dynamic program (SDP), allowing for accurate modelling of real-time recovery actions using the latest information, random events causing delays, and propagation of delays from port call to port call. Our theoretical analysis revealed that as the delay with respect to the timetable increases, so should our travelling speed.

Optimizing the buffer allocation decisions presented a challenge, because they must be exogenous to the SDP since they affect transitions in multiple states. In general, only enumeration techniques can optimize over variables exogenous to an SDP. But we were able to show, under relatively mild assumptions, that $\mathcal{C}^*(B)$ is convex in the buffer time variables. A detailed investigation of the cost function $\mathcal{C}^*(B)$ yielded a simple method to compute subgradients. Based on these results, we proposed a relatively simple algorithm.

In our experiments, the algorithm computes the optimal buffer time allocation in under 80 seconds. We compared the optimal schedule with the cost of the initial schedule as executed by Maersk Line and with the cost of a schedule in which buffer times are uniformly distributed over the ports. We observe that the uniform schedule provides very good solutions for schedules with low buffers, but that the optimal schedule generates costs that are six to ten million USD per year lower compared to the initial schedule. For schedules with high buffers, the optimal schedule also results in much lower costs than the uniform schedule.

Our experiments thus revealed that the proposed algorithm is very efficient. Its efficiency stems from the use of convexity of $\mathcal{C}^*(B)$, allowing us to take into account on-line speed optimization without severely reducing performance. Moreover, we directly extract subgradients from the SDP formulation, so we can take into account the stochasticity without sampling. These properties make the algorithm a good candidate for further research in timetable optimization, also in contexts other than container shipping. However, challenges need to be overcome to use the algorithm in settings where the timetable involves multiple trains/ships/metros that interact. Further research is needed to reveal whether the algorithm may be valuable in those settings as well.

Appendix

5.A Proofs of theoretical results

To simplify notation, define

$$\mathcal{L}_n(d;B) := \min_{0\le\tau\le\tau^u_{p[n]}} \left\{\mathcal{F}_{p[n]}(\tau) + \mathcal{K}_n(d+\tau;B)\right\}.$$

Then $\mathcal{C}^{dep}_{n,N}(d;B)$ can be written as

$$\mathcal{C}^{dep}_{n,N}(d;B) = \mathcal{D}^{dep}_{p[n]}(d) + \mathcal{L}_n(d;B). \tag{5.19}$$

Proof of Lemma 3. By backward induction in n, starting at N. Let $B \in \mathcal{B}$ be arbitrary. For $n = N$, $\mathcal{C}^{arr}_{N,N}(d;B) = \mathcal{D}^{arr}_{p[N]}(d)$, which is a nondecreasing function in d by assumption.

Assume now that $\mathcal{C}^{arr}_{n+1,N}(d;B)$ is nondecreasing in d for some $1 < n < N$. We will prove that $\mathcal{C}^{arr}_{n,N}(d;B)$ is also nondecreasing in d. Let $d, d' \in \mathbb{R}_{\ge 0}$ be arbitrary such that $d' \ge d$. Then,

$$\begin{aligned}
\mathcal{K}_n(d;B) &= \mathbb{E}_{X^s_n}\left[\mathcal{C}^{arr}_{n+1,N}\left(\left(d - B_{p[n]} + X^s_n\right)^+; B\right)\right] \\
&\le \mathbb{E}_{X^s_n}\left[\mathcal{C}^{arr}_{n+1,N}\left(\left(d' - B_{p[n]} + X^s_n\right)^+; B\right)\right] \\
&= \mathcal{K}_n(d';B),
\end{aligned}$$

where the inequality follows from the induction hypothesis and because X^s_n does not depend on the current delay, since by assumption X^s_n is independent of all other random variables. This proves that $\mathcal{K}_n(d;B)$ is nondecreasing in d. Then,

$$\begin{aligned}
\mathcal{L}_n(d;B) &= \min_{0\le\tau\le\tau^u_{p[n]}} \left\{\mathcal{F}_{p[n]}(\tau) + \mathcal{K}_n\left(d+\tau;B\right)\right\} \\
&\le \mathcal{F}_{p[n]}\left(\mathcal{T}_n(d';B)\right) + \mathcal{K}_n(d + \mathcal{T}_n\left(d';B\right);B) \\
&\le \mathcal{F}_{p[n]}\left(\mathcal{T}_n(d';B)\right) + \mathcal{K}_n\left(d' + \mathcal{T}_n(d';B);B\right) \\
&= \mathcal{L}_n(d';B),
\end{aligned}$$

where the first inequality holds because $0 \le \mathcal{T}_n(d';B) \le \tau^u_{p[n]}$ and the second because $\mathcal{K}_n(d;B)$ is nondecreasing in d. Hence, $\mathcal{L}_n(d;B)$ is nondecreasing in d. By (5.19) we know that $\mathcal{C}^{dep}_{n,N}(d;B)$ is the sum of two nondecreasing functions, namely $\mathcal{D}_{p[n]}(d)$ and $\mathcal{L}_n(d;B)$, which proves that $\mathcal{C}^{dep}_{n,N}(d;B)$ is also nondecreasing in d.

Further, $\mathcal{C}^{arr}_{n,N}(d;B)$ is the sum of $\mathcal{D}^{arr}_{p[n]}(d)$, which is nondecreasing in d by assumption, and $\mathbb{E}_{X^p_n}\left[\min_{\gamma\geq 0}\left\{c^e\gamma + \mathcal{C}^{dep}_{n,N}((d+X^p_n-\gamma)^+;B)\right\}\right]$, for which we find:

$$\begin{aligned}
&\mathbb{E}_{X^p_n}\left[\min_{\gamma\geq 0}\left\{c^e\gamma + \mathcal{C}^{dep}_{n,N}\left((d+X^p_n-\gamma)^+;B\right)\right\}\right]\\
&\leq \mathbb{E}_{X^p_n}\left[c^e\mathcal{Y}_n(d'+X^p_n;B) + \mathcal{C}^{dep}_{n,N}\Big((d+X^p_n-\mathcal{Y}_n(d'+X^p_n;B))^+;B\Big)\right]\\
&\leq \mathbb{E}_{X^p_n}\left[c^e\mathcal{Y}_n(d'+X^p_n;B) + \mathcal{C}^{dep}_{n,N}\Big((d'+X^p_n-\mathcal{Y}_n(d'+X^p_n;B))^+;B\Big)\right]\\
&= \mathbb{E}_{X^p_n}\left[\min_{\gamma\geq 0}\left\{c^e\gamma + \mathcal{C}^{dep}_{n,N}\Big((d'+X^p_n-\gamma)^+;B\Big)\right\}\right]
\end{aligned}$$

where the second inequality holds because $\mathcal{C}^{dep}_{n,N}(d;B)$ is nondecreasing in d. The last equality holds because the additional delay incurred is independent of the current delay, because by assumptions it is independent of all other random variables. Hence, $\mathcal{C}^{arr}_{n,N}(d;B)$ is nondecreasing in d, which completes the induction argument. □

Proof of Lemma 4. By backward induction in n, starting at N. $\mathcal{C}^{arr}_{N,N}(d;B) = \mathcal{D}^{arr}_{p[N]}(d)$ is joint convex in d and B by assumption. Now suppose $\mathcal{C}^{arr}_{n+1,N}(d;B)$ is joint convex in d and B for some $1 < n < N$. Let $d, d' \in \mathbb{R}_{\geq 0}$ be arbitrary nonnegative real numbers and let $B, B' \in \mathcal{B}$ and $\lambda \in [0,1]$ be arbitrary. Then,

$$\begin{aligned}
&\lambda\mathcal{K}_n(d;B) + (1-\lambda)\mathcal{K}_n(d';B')\\
&\quad= \lambda\mathbb{E}_{X^s_n}\left[\mathcal{C}^{arr}_{n+1,N}\Big(\left(d-B_{p[n]}+X^s_n\right)^+;B\Big)\right]\\
&\qquad+(1-\lambda)\mathbb{E}_{X^s_n}\left[\mathcal{C}^{arr}_{n+1,N}\Big(\left(d'-B'_{p[n]}+X^s_n\right)^+;B'\Big)\right]\\
&\quad\geq \mathbb{E}_{X^s_n}\left[\mathcal{C}^{arr}_{n+1,N}\Big(\lambda\left(d-B_{p[n]}+X^s_n\right)^+ + (1-\lambda)\left(d'-B'_{p[n]}+X^s_n\right)^+;\lambda B+(1-\lambda)B'\Big)\right]\\
&\quad\geq \mathbb{E}_{X^s_n}\left[\mathcal{C}^{arr}_{n+1,N}\Big(\left(\lambda(d-B_{p[n]}+X^s_n) + (1-\lambda)(d'-B'_{p[n]}+X^s_n)\right)^+;\lambda B+(1-\lambda)B'\Big)\right]\\
&\quad= \mathcal{K}_n(\lambda d+(1-\lambda)d';\lambda B+(1-\lambda)B'),
\end{aligned}$$

where the first inequality holds by the induction hypothesis and because X^s_n is independent of the current delay, since by assumption it is independent of all other random variables. The second inequality follows because $\mathcal{C}^{arr}_{n+1,N}(d,B)$ nondecreasing in d. It follows that

$\mathcal{K}_n(d;B)$ is also joint convex in d and B. Next,

$$\begin{aligned}
&\lambda\mathcal{L}_n(d;B)+(1-\lambda)\mathcal{L}_n(d';B')\\
&\quad=\lambda\left(\mathcal{F}_{p[n]}\left(\mathcal{T}_n(d;B)\right)+\mathcal{K}_n\left(d+\mathcal{T}_n(d;B);B\right)\right)+\\
&\quad\quad(1-\lambda)\left(\mathcal{F}_{p[n]}\left(\mathcal{T}_n(d';B')\right)+\mathcal{K}_n\left(d'+\mathcal{T}_n(d';B');B'\right)\right)\\
&\quad=\lambda\mathcal{F}_{p[n]}\left(\mathcal{T}_n(d;B)\right)+(1-\lambda)\mathcal{F}_{p[n]}\left(\mathcal{T}_n(d';B')\right)+\\
&\quad\quad\lambda\mathcal{K}_n\left(d+\mathcal{T}_n(d;B);B\right)+(1-\lambda)\mathcal{K}_n\left(d'+\mathcal{T}_n(d';B');B'\right)\\
&\quad\geq\mathcal{F}_{p[n]}\left(\lambda\mathcal{T}_n(d;B)+(1-\lambda)\mathcal{T}_n(d';B')\right)+\\
&\quad\quad\mathcal{K}_n\left(\lambda\left(d+\mathcal{T}_n(d;B)\right)+(1-\lambda)\left(d'+\mathcal{T}_n(d';B')\right);\lambda B+(1-\lambda)B'\right)\\
&\quad\geq\min_{0\leq\tau\leq\tau^u_{p[n]}}\left\{\mathcal{F}_{p[n]}(\tau)+\mathcal{K}_n\left(\tau+\lambda d+(1-\lambda)d';\lambda B+(1-\lambda)B'\right)\right\}\\
&\quad=\mathcal{L}_n\left(\lambda d+(1-\lambda)d';\lambda B+(1-\lambda)B'\right)
\end{aligned}$$

where the first inequality follows because $\mathcal{F}_{p[n]}$ is convex and $\mathcal{K}_n$ is joint convex. The second inequality holds because $0\leq\mathcal{T}_n(d;B),\mathcal{T}_n(d';B')\leq\tau^u_{p[n]}$. Hence, $\mathcal{L}_n(d;B)$ is joint convex in d and B. Then, $\mathcal{C}^{dep}_{n,N}(d;B)$ is the sum of two (joint) convex functions, so $\mathcal{C}^{dep}_{n,N}(d;B)$ is joint convex in d and B.

Further, $\mathcal{C}^{arr}_{n,N}(d;B)$ is the sum of $\mathcal{D}^{arr}_{p[n]}(d)$, which is convex in d and hence joint convex in (d,B) by assumption, and $\mathbb{E}_{X^p_n}\left[\min_{\gamma\geq0}\left\{c^e\gamma+\mathcal{C}^{dep}_{n,N}((d^{arr}_n+X^p_n-\gamma)^+;B)\right\}\right]$, for which we find:

$$\begin{aligned}
&\lambda\mathbb{E}_{X^p_n}\left[\left(c^e\mathcal{Y}_n(d+X^p_n;B)+\mathcal{C}^{dep}_{n,N}\left((d+X^p_n-\mathcal{Y}_n(d+X^p_n;B))^+;B\right)\right)\right]+\\
&(1-\lambda)\mathbb{E}_{X^p_n}\left[\left(c^e\mathcal{Y}_n(d'+X^p_n;B')+\mathcal{C}^{dep}_{n,N}\left((d'+X^p_n-\mathcal{Y}_n(d'+X^p_n;B'))^+;B'\right)\right)\right]\\
\geq{}&\mathbb{E}_{X^p_n}\left[c^e\left(\lambda\mathcal{Y}_n(d+X^p_n;B)+(1-\lambda)\mathcal{Y}_n(d'+X^p_n;B')\right)\right]+\\
&\mathbb{E}_{X^p_n}\Big[\mathcal{C}^{dep}_{n,N}\Big(\lambda\left(d+X^p_n-\mathcal{Y}_n(d+X^p_n;B)\right)^++(1-\lambda)\left(d'+X^p_n-\mathcal{Y}_n(d'+X^p_n;B')\right)^+;\\
&\lambda B+(1-\lambda)B'\Big)\Big]\\
\geq{}&\mathbb{E}_{X^p_n}\left[c^e\left(\lambda\mathcal{Y}_n(d+X^p_n;B)+(1-\lambda)\mathcal{Y}_n(d'+X^p_n;B')\right)\right]+\\
&\mathbb{E}_{X^p_n}\Big[\mathcal{C}^{dep}_{n,N}\Big(\left(\lambda d+(1-\lambda)d'+X^p_n-\left(\lambda\mathcal{Y}_n(d+X^p_n;B)+(1-\lambda)\mathcal{Y}_n(d'+X^p_n;B')\right)\right)^+;\\
&\lambda B+(1-\lambda)B'\Big)\Big]\\
\geq{}&\mathbb{E}_{X^p_n}\left[\min_{\gamma\geq0}\left\{c^e\gamma+\mathcal{C}^{dep}_{n,N}\left((\lambda d+(1-\lambda)d'+X^p_n-\gamma)^+;\lambda B+(1-\lambda)B'\right)\right\}\right]
\end{aligned}$$

where the first inequality holds by the induction hypothesis and the second inequality holds because $\mathcal{C}^{dep}_{n,N}(d;B)$ is nondecreasing in d. Hence, $\mathcal{C}^{arr}_{n,N}(d;B)$ is joint convex in d and B, which proves the lemma. □

Proof of Theorem 5. The $\lim_{R\to\infty} \frac{\mathcal{C}^{dep}_{1,R|P|+1}(d_1^{dep};B)}{R}$ exists by assumption, and since convexity is preserved when taking limits, this limit is joint convex in B and d_1^{dep} by Lemma 4. By assumption, the limit is independent of d_1^{dep}, which implies the desired result. □

Proof of Lemma 6. Take any $n \in [1, N-1]$ and $B \in \mathcal{B}$ and let $d, d' \in \mathbb{R}_{\geq 0}$ such that $d' \geq d$. We will prove that $\mathcal{T}_n(d;B) \geq \mathcal{T}_n(d';B)$ by contradiction. Suppose $\mathcal{T}_n(d;B) < \mathcal{T}_n(d';B)$. By (5.6) it follows that

$$\begin{aligned}\mathcal{F}_{p[n]}\left(\mathcal{T}_n(d;B)\right) + \mathcal{K}_n\left(d + \mathcal{T}_n(d;B);B\right) &\leq \mathcal{F}_{p[n]}\left(\mathcal{T}_n(d';B)\right) + \mathcal{K}_n\left(d + \mathcal{T}_n(d';B);B\right)\\ \mathcal{F}_{p[n]}\left(\mathcal{T}_n(d;B)\right) - \mathcal{F}_{p[n]}\left(\mathcal{T}_n(d';B)\right) &\leq \mathcal{K}_n\left(d + \mathcal{T}_n(d';B);B\right) - \mathcal{K}_n\left(d + \mathcal{T}_n(d;B);B\right).\end{aligned}$$

Furthermore, by (5.6) it follows that:

$$\begin{aligned}\mathcal{F}_{p[n]}\left(\mathcal{T}_n(d';B)\right) + \mathcal{K}_n\left(d' + \mathcal{T}_n(d';B);B\right) &< \mathcal{F}_{p[n]}\left(\mathcal{T}_n(d;B)\right) + \mathcal{K}_n\left(d' + \mathcal{T}_n(d;B);B\right)\\ \mathcal{F}_{p[n]}\left(\mathcal{T}_n(d;B)\right) - \mathcal{F}_{p[n]}\left(\mathcal{T}_n(d';B)\right) &> \mathcal{K}_n\left(d' + \mathcal{T}_n(d';B);B\right) - \mathcal{K}_n\left(d' + \mathcal{T}_n(d;B);B\right).\end{aligned}$$

Hence,

$$\begin{aligned}\mathcal{K}_n\left(d' + \mathcal{T}_n(d';B);B\right) - \mathcal{K}_n\left(d' + \mathcal{T}_n(d;B);B\right) &< \mathcal{F}_{p[n]}\left(\mathcal{T}_n(d;B)\right) - \mathcal{F}_{p[n]}\left(\mathcal{T}_n(d';B)\right)\\ &\leq \mathcal{K}_n\left(d + \mathcal{T}_n(d';B);B\right) - \mathcal{K}_n\left(d + \mathcal{T}_n(d;B);B\right),\end{aligned}$$

thus $\mathcal{K}_n(d;B)$ has decreasing increments, which contradicts the convexity of $\mathcal{K}_n(d;B)$. Hence, $\mathcal{T}_n(d;B) \geq \mathcal{T}_n(d';B)$. The second part of the lemma can be proven analogously. □

Proof of Lemma 7. Let $n \in [1, N-1]$ be arbitrary and let $d, d' \in \mathbb{R}_{\geq 0}$ be arbitrary such that $d' \geq d$. We need to prove that $d + \mathcal{T}_n(d;B) \leq d' + \mathcal{T}_n(d';B)$, because, since $B_{p[n]}$ is fixed and X_n^s is independent of d_n^{dep} by assumption, this implies the desired result.

Define $\tau' := d + \mathcal{T}_n(d;B) - d'$ and $\tau := d' + \mathcal{T}_n(d';B) - d$. Assume now (by contradiction) that $d + \mathcal{T}_n(d;B) > d' + \mathcal{T}_n(d';B)$. Then $\tau' > \mathcal{T}_n(d';B)$ and $\tau < \mathcal{T}_n(d;B)$. By (5.6)

$$\mathcal{F}_{p[n]}\left(\mathcal{T}_n(d;B)\right) + \mathcal{K}_n\left(d + \mathcal{T}_n(d;B);B\right) < \mathcal{F}_{p[n]}(\tau) + \mathcal{K}_n(d + \tau;B).$$

The inequality is strict because $\mathcal{T}_n(d;B)$ is by definition the smallest minimizer, see (5.6). By rearranging terms, we obtain

$$\begin{aligned}\mathcal{F}_{p[n]}\left(\mathcal{T}_n(d;B)\right) - \mathcal{F}_{p[n]}(\tau) &< \mathcal{K}_n(d+\tau;B) - \mathcal{K}_n\left(d+\mathcal{T}_n(d;B);B\right)\\ &= \mathcal{K}_n\left(d'+\mathcal{T}_n(d';B);B\right) - \mathcal{K}_n\left(d+\mathcal{T}_n(d;B);B\right).\end{aligned}$$

where the equality is due to the definition of τ. For $\mathcal{T}_n(d';B)$, the definition (5.6) implies

$$\mathcal{F}_{p[n]}\left(\mathcal{T}_n(d';B)\right) + \mathcal{K}_n\left(d'+\mathcal{T}_n(d';B);B\right) \leq \mathcal{F}_{p[n]}(\tau') + \mathcal{K}_n(d'+\tau';B).$$

which implies

$$\begin{aligned}\mathcal{F}_{p[n]}(\tau') - \mathcal{F}_{p[n]}\left(\mathcal{T}_n(d';B)\right) &\geq \mathcal{K}_n\left(d'+\mathcal{T}_n(d';B);B\right) - \mathcal{K}_n(d'+\tau';B)\\ &= \mathcal{K}_n\left(d'+\mathcal{T}_n(d';B);B\right) - \mathcal{K}_n\left(d+\mathcal{T}_n(d;B);B\right).\end{aligned}$$

Combining these, we obtain:

$$\begin{aligned}\mathcal{F}_{p[n]}\left(\mathcal{T}_n(d;B)\right) - \mathcal{F}_{p[n]}(\tau) &< \mathcal{K}_n\left(d'+\mathcal{T}_n(d';B);B\right) - \mathcal{K}_n\left(d+\mathcal{T}_n(d;B);B\right)\\ &\leq \mathcal{F}_{p[n]}(\tau') - \mathcal{F}_{p[n]}\left(\mathcal{T}_n(d';B)\right)\\ &= \mathcal{F}_{p[n]}\left(\mathcal{T}_n(d;B) - (d'-d)\right) - \mathcal{F}_{p[n]}(\tau - (d'-d)).\end{aligned}$$

Thus, $\mathcal{F}_{p[n]}(d)$ has decreasing increments which contradicts convexity. Thus, $d+\mathcal{T}_n(d;B) \leq d' + \mathcal{T}_n(d';B)$. The second part of the lemma can be proven analogously. □

To prove Lemma 8, we first prove that the value functions are continuous and piecewise linear with specific breakpoints. Define $\Psi(Q_n)$ as the set of functions that are piecewise linear, with breakpoints only on Q_n. More precisely: for any $f(\cdot) \in \Psi(Q_n)$ and any open interval $(\underline{d}, \bar{d})$ that does not intersect Q_n (thus $(\underline{d}, \bar{d}) \subseteq \mathbb{R} \setminus Q_n$), there exist a slope $a \in \mathbb{R}$ and an offset $b \in \mathbb{R}$ such that $\forall d \in (\underline{d}, \bar{d}) : f(d) = ad + b$.

Lemma 11 (Auxiliary towards Lemma 8)**.** *For every n with $1 \leq n < N$: $\mathcal{K}_n(\,\cdot\,;B) \in \Psi(Q_n)$, $\mathcal{C}^{dep}_{n,N}(\,\cdot\,;B) \in \Psi(Q_n)$, $\mathcal{C}^{arr}_{n+1,N}(\,\cdot\,;B) \in \Psi(Q_{n+1})$. This yields additional results for the actions:*

1. *Optimal sailing time: for every n with $1 \leq n < N$ and every $d \geq 0$ that $\mathcal{T}_n(d;B) \in \{zl | z \in \mathbb{Z}\}$ and/or $d + \mathcal{T}_n(d;B) \in Q_n$.*

2. *Extreme actions: for every n with $1 < n < N$ and every $d \geq 0$ that $\mathcal{Y}_n(d;B) = 0$ and/or $d - \mathcal{Y}_n(d;B) \in Q_n$.*

Proof of Lemma 11. We will prove the lemma by induction. For the base case, note that $\mathcal{C}^{arr}_{N,N}(d^{arr}_N; B) = \mathcal{D}^{arr}_{p[N]}(d^{arr}_N) \in \Psi(Q_N)$, since $\mathcal{D}^{arr}_{p[N]}(d^{arr}_N)$ is piecewise linear with breakpoints on $\{zl | z \in \mathbb{Z}\}$ by Assumption 4, and $\{zl | z \in \mathbb{Z}\} \subseteq Q_N$ because $\{zl | z \in \mathbb{Z}\} = Q_N(p[N])$.

Thus, for some n with $1 \leq n < N$, the induction hypothesis is $\mathcal{C}^{arr}_{n+1,N}(d; B) \in \Psi(Q_{n+1})$, and we will show that $\mathcal{K}_n(d; B) \in \Psi(Q_n)$, $\mathcal{C}^{dep}_{n,N}(d; B) \in \Psi(Q_n)$ and $\mathcal{C}^{arr}_{n,N}(d; B) \in \Psi(Q_n)$.

We first show that $\mathcal{K}_n(d; B) = \mathbb{E}_{X^s_n}\left[\mathcal{C}^{arr}_{n+1,N}\left((d - B_{p[n]} + X^s_n)^+; B\right)\right]$ is in $\Psi(Q_n)$. Conditioned on X^s_n, by Assumption 4 there exists $z^s \in \mathbb{Z}$ such that $X^s_n = z^s l$. By definition of the cumulative buffers $\tilde{B}'$ it holds that $\tilde{B}'_{p[n+1]} - B'_{p[n]} = \tilde{B}'_{p[n]} + z_n l$, with $z_n = \overline{B}/l \in \mathbb{Z}$ if $p[n+1] = 1$ and $z_n = 0$ otherwise. Fix any open interval $(\underline{d}, \bar{d}) \subseteq \mathbb{R} \setminus Q_n$. For any $d \in (\underline{d}, \bar{d})$ suppose $d - B_{p[n]} + z^s l \in Q_{n+1}$. That would imply $\exists z \in \mathbb{Z}, p \in P$ such that $d - B_{p[n]} + z^s l = zl + \tilde{B}_p - \tilde{B}_{p[n+1]}$, and thus $d = (z - z_s)l + \tilde{B}_p - (\tilde{B}_{p[n+1]} - B_{p[n]}) = (z - z^s - z_n)l + \tilde{B}_p - \tilde{B}_{p[n]} \in Q_n$, a contradiction with $(\underline{d}, \bar{d}) \subseteq \mathbb{R} \setminus Q_n$. Hence, it holds that $d - B_{p[n]} + z^s l \notin Q_{n+1}$. Thus $(\underline{d} - B_{p[n]} + z^s l, \bar{d} - B_{p[n]} + z^s l) \subseteq \mathbb{R} \setminus Q_{n+1}$. Since $0 \in Q_{n+1}$, the following two cases are exhaustive: 1) $\forall d \in (\underline{d}, \bar{d}) : d - B_{p[n]} + z^s l \geq 0$ and 2) $\forall d \in (\underline{d}, \bar{d}) : d - B_{p[n]} + z^s l \leq 0$. For the first case, by induction hypothesis and since $(\underline{d} - B_{p[n]} + z^s l, \bar{d} - B_{p[n]} + z^s l) \subseteq \mathbb{R} \setminus Q_{n+1}$, we know that $\exists a, b \in \mathbb{R}$ such that $\forall d \in (\underline{d}, \bar{d})$:

$$\mathcal{C}^{arr}_{n+1,N}\left((d - B_{p[n]} + z^s l)^+; B\right) = \mathcal{C}^{arr}_{n+1,N}\left(d - B_{p[n]} + z^s l; B\right) = a(d - B_{p[n]} + z^s l) + b.$$

Note that the RHS is affine in d. For the second case, we find

$$\forall d \in (\underline{d}, \bar{d}) : \mathcal{C}^{arr}_{n+1,N}\left((d - B_{p[n]} + z^s l)^+; B\right) = \mathcal{C}^{arr}_{n+1,N}(0; B) = a'd + b',$$

with $a' = 0$ and $b' = \mathcal{C}^{arr}_{n+1,N}(0; B)$. Now, since

$$\mathcal{K}_n(d; B) = \sum_{z^s \in \mathbb{Z}} \mathbb{P}(X^s_n = z^s l)\mathcal{C}^{arr}_{n+1,N}\left((d - B_{p[n]} + z^s l)^+; B\right),$$

and since each of the functions on the RHS is affine in d for all $d \in (\underline{d}, \bar{d})$, $\mathcal{K}_n(d; B)$ is affine in d for $d \in (\underline{d}, \bar{d})$. This proves $\mathcal{K}_n(d; B) \in \Psi(Q_n)$.

We next show that $\mathcal{C}^{dep}_{n,N}(d; B) = \mathcal{D}^{dep}_{p[n]}(d) + \mathcal{L}_n(d; B) \in \Psi(Q_n)$, where $\mathcal{L}_n(d; B) = \min_{0 \leq \tau \leq \tau^u_{p[n]}} \left\{\mathcal{F}_{p[n]}(\tau) + \mathcal{K}_n(d + \tau; B)\right\}$. Since $\mathcal{D}^{dep}_{p[n]}(d)$ is piecewise linear with breakpoints on $\{zl | z \in \mathbb{Z}\} \subseteq Q_n$ by Assumption 4, it remains to show that $\mathcal{L}_n(d; B) \in \Psi(Q_n)$. Fix an interval $(\underline{d}, \bar{d}) \subseteq \mathbb{R} \setminus Q_n$, and let $d \in (\underline{d}, \bar{d})$. For brevity, let $\tau^* = \mathcal{T}(d; B)$ denote the optimal sailing time for d. By (5.6), τ^* is the smallest minimizer of $\mathcal{F}_{p[n]}(\tau) + \mathcal{K}_n(d + \tau; B)$, and therefore τ^* must be one of the breakpoints of $\mathcal{F}_{p[n]}(\,\cdot\,)$ (which occur at $\{zl | z \in \mathbb{Z}\}$) and/or

$d + \tau^*$ must be one of the breakpoints of $\mathcal{K}_n(\,\cdot\,; B)$ (which occur at Q_n by $\mathcal{K}_n(\,\cdot\,; B) \in \Psi(Q_n)$). It thus suffices to consider the following two cases: 1) $\tau^* \in \{zl | z \in \mathbb{Z}\}$ and 2) $\tau^* + d \in Q_n$. (This is additional result 1.)

For Case 1, note that for every $d' \in (\underline{d}, \bar{d})$ we have $d' \notin Q_n$ and $\tau^* \in \{zl | z \in \mathbb{Z}\}$ and thus $d' + \tau^* \notin Q_n$. This implies $(\underline{d} + \tau^*, \bar{d} + \tau^*) \subseteq \mathbb{R} \setminus Q_n$. Thus, by $\mathcal{K}_n(d; B) \in \Psi(Q_n)$ there exist $a, b \in \mathbb{R}$ such that for every $d' \in (\underline{d}, \bar{d})$:

$$\mathcal{L}_n(d'; B) \leq \mathcal{F}_{p[n]}(\tau^*) + \mathcal{K}_n(d' + \tau^*; B) = \mathcal{F}_{p[n]}(\tau^*) + a(d' + \tau^*) + b = a'd' + b' \quad (5.20)$$

with $a' = a$ and $b' = b + a\tau^* + \mathcal{F}_{p[n]}(\tau^*)$. Write $d' = d + x$. We now show that $\mathcal{L}_n(d + x; B) = a'(d + x) + b'$. This is immediate for $x = 0$, so suppose $x \neq 0$. Let $\epsilon > 0$ be such that $d - \epsilon x \in (\underline{d}, \bar{d})$. The proof of Lemma 4 shows that $\mathcal{L}_n(d; B)$ is joint convex in (d, B), and therefore convex in d, which implies $\lambda\mathcal{L}_n(d + x; B) + (1 - \lambda)\mathcal{L}_n(d - \epsilon x; B) \geq \mathcal{L}_n\big(\lambda(d + x) + (1 - \lambda)(d - \epsilon x); B\big)$ for any $\lambda \in [0, 1]$. Setting $\lambda = \epsilon/(1 + \epsilon)$ and multiplying by $(1 + \epsilon)$ yields:

$$\begin{aligned}
\epsilon\mathcal{L}_n(d + x; B) &\geq (1 + \epsilon)\mathcal{L}_n(d; B) - \mathcal{L}_n(d - \epsilon x; B) \\
&\geq (1 + \epsilon)[a'd + b'] - [a'(d - \epsilon x) + b'] \\
&= \epsilon[a'(d + x) + b'] \\
&= \epsilon[\mathcal{F}_{p[n]}(\tau^*) + \mathcal{K}_n(d + x + \tau^*; B)] \geq \epsilon\mathcal{L}_n(d + x; B)
\end{aligned}$$

where the second inequality results from (5.20) and optimality of τ^* for d, the equality at the third line rearranges terms, and the final (in)equalities result from (5.20). This shows $\mathcal{L}_n(d'; B) = a'd' + b'$ (which implies that τ^* is optimal for every $d' \in (\underline{d}, \bar{d})$). Thus for Case 1 we have established that $\mathcal{L}_n(d'; B)$ is affine in d' for all $d' \in (\underline{d}, \bar{d})$.

Now Case 2: $\tau^* + d \in Q_n$. For any $d' \in (\underline{d}, \bar{d})$, we will show that the action $\tau' = \tau^* - d' + d$ is optimal. Because $\tau' + d' = \tau^* + d \in Q_n$ and $d' \notin Q_n$, we know that $\tau' \notin \{zl | z \in \mathbb{Z}\}$. This implies $(\tau^* - \bar{d} + d, \tau^* - \underline{d} + d) \subseteq \mathbb{R} \setminus \{zl | z \in \mathbb{Z}\}$. Because $\mathcal{F}_{p[n]}(\tau)$ is piecewise linear with breakpoints on $\{zl | z \in \mathbb{Z}\}$ by Assumption 4, we now know that there exist $a, b \in \mathbb{R}$ such that for every $\tau' \in (\tau^* - \bar{d} + d, \tau^* - \underline{d} + d)$ it holds that $\mathcal{F}_{p[n]}(\tau') = a\tau' + b$. This yields:

$$\mathcal{L}_n(d'; B) \leq a\tau' + b + \mathcal{K}_n(d' + \tau'; B) = a(\tau^* - d' + d) + \mathcal{K}_n(d + \tau^*; B) = a'd' + b' \quad (5.21)$$

with $a' = -a$ and $b' = a\tau^* + ad + \mathcal{K}_n(d + \tau^*; B)$. This allows us to show that $\mathcal{L}_n(d'; B) = a'd' + b'$, exactly in the same fashion as for Case 1, using (5.21) and convexity of $\mathcal{L}_n(d'; B)$. Thus also for Case 2, we have established that $\mathcal{L}_n(d'; B)$ is affine in d' for all $d' \in (\underline{d}, \bar{d})$.

Since the two cases are exhaustive, we have shown that $\mathcal{L}_n(d'; B)$ is affine in d' for all $d' \in (\underline{d}, \bar{d})$. This shows that $\mathcal{L}_n(d'; B) \in \Psi(Q_n)$, and thus $\mathcal{C}^{dep}_{n,N}(d; B) = \mathcal{D}^{dep}_{p[n]}(d) + \mathcal{L}_n(d; B) \in \Psi(Q_n)$.

Finally, we show that $\mathcal{C}^{arr}_{n,N}(d; B) = \mathcal{D}^{arr}_{p[n]}(d) + \mathbb{E}_{X^p_n}\left[\min_{\gamma \geq 0}\left\{c^e\gamma + \mathcal{C}^{dep}_{n,N}((d + X^p_n - \gamma)^+; B)\right\}\right]$ is in $\Psi(Q_n)$. We condition on X^p_n, and write $X^p_n = z^p l$, with $z^p \in \mathbb{Z}$ by Assumption 4. We first show that $\min_{\gamma \geq 0}\left\{c^e\gamma + \mathcal{C}^{dep}_{n,N}((d + z^p l - \gamma)^+; B)\right\} \in \Psi(Q_n)$. Fix an interval $(\underline{d}, \bar{d}) \subseteq \mathbb{R} \setminus Q_n$, and let $d \in (\underline{d}, \bar{d})$. Denote $\gamma^* = \mathcal{Y}(d + z^p l, B)$ for brevity. Since $c^e > 0$ and $d + z^p l \geq 0$, optimality of γ^* implies that $\gamma^* \leq d + z^p l$, and thus $(d + z^p l - \gamma^*)^+ = d + z^p l - \gamma^*$. Also, γ^* is the largest minimizer of $c^e\gamma + \mathcal{C}^{dep}_{n,N}((d + z^p l - \gamma)^+; B)$, and the following to cases are thus exhaustive: Case 1) $\gamma^* = 0$ and Case 2) $(d + z^p l - \gamma^*)^+ = d + z^p l - \gamma^*$ is a breakpoint of $\mathcal{C}^{dep}_{n,N}(\,\cdot\,; B)$, and thus $d + z^p l - \gamma^* \in Q_n$. (This is additional result 2.)

For Case 1, since $\forall d' \in (\underline{d}, \bar{d}) : d' \notin Q_n$, we know that $d' + z^p l - \gamma^* \notin Q_n$, and thus $(\underline{d} + z^p l - \gamma^*, \bar{d} + z^p l - \gamma^*) \subseteq \mathbb{R} \setminus Q_n$. Therefore, by $\mathcal{C}^{dep}_{n,N}(\cdot; B) \in \Psi(Q_n)$, there exist $a, b \in \mathbb{R}$ such that $\forall d' \in (\underline{d}, \bar{d})$:

$$\min_{\gamma \geq 0}\left\{c^e\gamma + \mathcal{C}^{dep}_{n,N}((d' + z^p l - \gamma)^+; B)\right\} \leq c^e\gamma^* + \mathcal{C}^{dep}_{n,N}(d' + z^p l - \gamma^*; B) = ad' + b \quad (5.22)$$

Because $\gamma^* = 0$ is optimal for $d' = d$ by definition, we can proceed in the same way as before to show that the inequality in (5.22) can be strengthened to an equality. For Case 2, we note that $\gamma' = \gamma^* + d' - d > 0$ for $d' \in (\underline{d}, \bar{d})$, because $\gamma' = 0$ would contradict $d' + z^p l - \gamma' = d + z^p l - \gamma^* \in Q_n$, since $d' + z^p l \notin Q_n$ by construction. We obtain:

$$\min_{\gamma \geq 0}\left\{c^e\gamma + \mathcal{C}^{dep}_{n,N}((d' + z^p l - \gamma)^+; B)\right\} \leq c^e(\gamma^* + d' - d) + \mathcal{C}^{dep}_{n,N}(d + z^p l - \gamma^*; B) = ad' + b, \quad (5.23)$$

with $a = c^e$ and $b = c^e(\gamma^* - d) + \mathcal{C}^{dep}_{n,N}(d + z^p l - \gamma^*; B)$. Since γ^* is optimal for d by assumption, we can proceed in the same way as before to show that the inequality in (5.23) can be strengthened to equality. This yields $\min_{\gamma \geq 0}\left\{c^e\gamma + \mathcal{C}^{dep}_{n,N}((d' + z^p l - \gamma)^+; B)\right\} \in \Psi(Q_n)$ and since $\mathcal{D}^{arr}_{p[n]}(d)$ is piecewise linear with breakpoints on $\{zl | z \in \mathbb{Z}\}$, we find that $\mathcal{C}^{arr}_{n,N}(d; B) \in \Psi(Q_n)$, which completes the proof. □

With this lemma, we are now ready to prove Lemma 8.

Proof of Lemma 8. The proof is by induction, starting at $n = 1$. Note that $d_1^{dep} \in \{zl|z \in \mathbb{Z}\} \subseteq Q_1$ by Assumption 4 and by definition of Q_1. We will now assume that $d_n^{dep} \in Q_n$ holds for some n with $1 \leq n < N$.

By additional result 1 of Lemma 11, we must either have $\mathcal{T}_n(d_n^{dep}; B) \in \{zl|z \in \mathbb{Z}\}$ or $d_n^{dep} + \mathcal{T}_n(d_n^{dep}; B) \in Q_n$. Because by assumption $d_n^{dep} \in Q_n$, in both cases we obtain $d_n^{dep} + \mathcal{T}_n(d_n^{dep}; B) \in Q_n$, and thus $\exists z \in \mathbb{Z}, p \in P$ such that $d_n^{dep} + \mathcal{T}_n(d_n^{dep}; B) = zl + \tilde{B}_p - \tilde{B}_{p[n]}$. Since X_n^s takes on integer multiples of l, write $X_n^s = z^s l$. By definition of the cumulative buffers $\tilde{B}'$ it holds that $\tilde{B}'_{p[n]} + B'_{p[n]} = \tilde{B}'_{p[n+1]} + z_n l$, with $z_n = \overline{B}/l \in \mathbb{Z}$ if $p[n+1] = 1$ and $z_n = 0$ otherwise. Thus $d_{n+1}^{arr} = (d_n^{dep} + \mathcal{T}_n(d_n^{dep}; B) + X_n^s - B_{p[n]})^+ = \big((z + z^s)l + \tilde{B}_p - \tilde{B}_{p[n]} - B_{p[n]}\big)^+ = \big((z + z^s - z_n)l + \tilde{B}_p - \tilde{B}_{p[n+1]}\big)^+$. Now consider the cases $d_{n+1}^{arr} = 0$ and $d_{n+1}^{arr} > 0$. In the former case, $d_{n+1}^{arr} \in Q_{p[n+1]}(p[n+1]) \subseteq Q_{p[n+1]}$, and in the latter case we find $d_{n+1}^{arr} = (z + z^s - z_n)l + \tilde{B}_p - \tilde{B}_{p[n+1]} \in Q_{p[n+1]}(p) \subseteq Q_{p[n+1]}$. Thus $d_{n+1}^{arr} \in Q_{n+1}$. Since X_{n+1}^p takes on values in $\{zl|z \in \mathbb{Z}\}$, $d_{n+1}^{arr} + X_{n+1}^p \in Q_{n+1}$ follows immediately.

Now, by additional result 2 of Lemma 11, we must either have $\mathcal{Y}_{n+1}(d_{n+1}^{arr} + X_{n+1}^p; B) = 0$ or $d_{n+1}^{arr} + X_{n+1}^p - \mathcal{Y}_{n+1}(d_{n+1}^{arr} + X_{n+1}^p; B) \in Q_n$. Since $d_{n+1}^{arr} + X_{n+1}^p \in Q_{n+1}$, in both cases we obtain $d_{n+1}^{arr} + X_{n+1}^p - \mathcal{Y}_{n+1}(d_{n+1}^{arr} + X_{n+1}^p; B) \in Q_n$. Thus $d_{n+1}^{dep} = (d_{n+1}^{arr} + X_{n+1}^p - \mathcal{Y}_{n+1}(d_{n+1}^{arr} + X_{n+1}^p; B))^+ \in Q_n$. This completes the proof by induction. □

Proof of Corollary 1. We first show that the limit in the corollary corresponds to the long-term average costs of a finite-state, finite-action SDP. We distinguish between departure states and port states: Departure states are identified by the delay d_n^{dep} and the port $p[n]$ and correspond to the moment of departure. Port states are identified by the delay $d_n^{arr} + X_n^p$ after incurring port delay and the port $p[n]$. (States depend only on $p[n]$, and not on n.)

For the extreme action, we impose the additional restriction that $\gamma_n \geq d_n^{arr} + X_n^p - d_{p[n]}^{max}$. This does not affect $\mathcal{C}_{1,R|P|+1}^{dep}(d_1^{dep}; B)$, because $\gamma_n < d_n^{arr} + X_n^p - d_{p[n]}^{max}$ cannot be optimal since $\mathcal{D}_{p[n]}(d) - c^e d$ is monotonically increasing for all $d > d_{p[n]}^{max}$ by Assumption 5. We thus have $0 \leq d_n^{dep} \leq d_{p[n]}^{max} < \infty$, since early departure is not allowed. Also, $d_n^{arr} + X_n^p \geq d_n^{arr} \geq 0$ since early arrival is not allowed and $d_{n+1}^{arr} + X_{n+1}^p \leq (d_n^{dep} + \tau_n - B_{p[n]} + X_n^s)^+ + X_{n+1}^p \leq d_{p[n]}^{max} + \tau_{p[n]}^u + X_{p[n]}^{\mathrm{s,max}} + X_{p[n+1]}^{\mathrm{p,max}} < \infty$. Thus, delays are bounded below and above. In addition, only delays in $Q_{p[n]}$ occur by Lemma 8, and we will thus restrict the delays to this set without affecting $\mathcal{C}_{1,R|P|+1}^{dep}(d_1^{dep}; B)$.

For the actions, we have $0 \leq \tau_n \leq \tau_{p[n]}^u$ by assumption. Since $c^e > 0$, it can never be optimal for γ_n to exceed $d_n^{arr} + X_n^p$ (for which we already found an upper bound), and $\gamma_n \geq 0$ by assumption. Thus actions can be bounded above and below. As a consequence of Lemma 8, we may impose that actions are in Q_p for some $p \in P$ without affecting

$\mathcal{C}^{dep}_{1,R|P|+1}(d^{dep}_1; B)$. This, together with boundedness of the actions, implies that only a finite number of actions need to be considered for each state.

Furthermore, $\mathcal{C}^{dep}_{1,R|P|+1}(d^{dep}_1; B)$ corresponds to the optimal expected costs incurred over R rounds, when starting with departure in port 1 and ending with arrival in port 1. During these R rounds, a total of $2R|P|$ states are visited, and thus $\mathcal{C}^{dep}_{1,R|P|+1}(d^{dep}_1; B)/(2R|P|)$ corresponds to the optimal average costs per state over the next $2R|P|$ states in a Markov Decision Problem (MDP), when starting with departure in port 1. This MDP has finitely many states and actions, by the above discussion. This implies that there exists a stationary deterministic policy that is average cost optimal (Bertsekas 2007, Prop. 4.1.3, Prop 4.1.7), proving the second claim of the corollary. Moreover, this implies that $\lim_{R\to\infty} \mathcal{C}^{dep}_{1,R|P|+1}(d^{dep}_1; B)\ /(2R|P|)$ exists in $\mathbb{R}$ for all $d^{dep}_1 \in \{zl|z \in \mathbb{Z}\}$ (Bertsekas 2007, Prop. 4.1.2, Prop. 4.1.3). Thus, the limit $\lim_{R\to\infty} \mathcal{C}^{dep}_{1,R|P|+1}(d^{dep}_1; B)/R$ also exists in $\mathbb{R}$ for all $d^{dep}_1 \in \{zl|z \in \mathbb{Z}\}$. (Note that $d^{dep}_1 \in \{zl|z \in \mathbb{Z}\}$ by Assumption 4.)

We next prove that $\lim_{R\to\infty} \mathcal{C}^{dep}_{1,R|P|+1}(d^{dep}_1; B)/R$ is independent of $d^{dep}_1 \in \{zl|z \in \mathbb{Z}\}$. Thereto, we will prove that the *weak accessibility* condition holds, which states that the set of states can be partitioned into two subsets S_1 and S_2 such that the following holds: 1) States $s \in S_1$ are transient under every stationary policy. 2) For every two states $s, s' \in S_2$, state s' is *accessible* from state s (Bertsekas 2007, p199). A state s' is accessible from state s if there exists a stationary policy such that the probability of entering state s' in a finite number of transitions starting from state s is strictly positive (Bertsekas 2007, p199).

Note that if s' is accessible from s and s'' is accessible from s', then s'' is accessible from s. Indeed, there must exist a sequence of states starting at s, going to s' and finally to s'', and if we take the right actions in all these states, there is a positive probability that this sequence occurs when we start at s. Should this sequence contain multiple visits to the same state (with different prescribed actions), then removing the loops yields a sequence from s to s'' that visits all states only once. For this latter sequence, a policy exists such that the sequence happens with positive probability when starting at s, showing that s'' is accessible from s.

Fix some port $p = p[n]$. Note that the state s_0 representing $d^{dep}_{p[n]} = 0$ is accessible from all states, by any policy that sets $\gamma_{p[n]} = d^{arr}_{p[n]} + X^p_n$. Hence, it remains to show that if a state s is recurrent under some policy, then s is accessible from s_0.

Assume $X^{\mathrm{s,max}}_{p[n]}$ and $X^{\mathrm{p,max}}_{p[n]}$ are chosen such that $\mathbb{P}(X^s_n = X^{\mathrm{s,max}}_{p[n]}) > 0$ and $\mathbb{P}(X^p_n = X^{\mathrm{p,max}}_{p[n]}) > 0$. Moreover, assume that $\overline{B} < \sum_{p=1}^{|P|}(\tau^u_p + X^{\mathrm{p,max}}_p + X^{\mathrm{s,max}}_p)$. (The *degenerate* alternative $\overline{B} \geq \sum_{p=1}^{|P|}(\tau^u_p + X^{\mathrm{p,max}}_p + X^{\mathrm{s,max}}_p)$ is ignored because trivially optimal solutions B with $B_p \geq X^{\mathrm{p,max}}_{p-1} + \tau^u_p + X^{\mathrm{s,max}}_p$, for $p \in P$ are feasible for this case.)

We show that there exists a port p for which the arrival delay $d_p^{arr} + X_n^p > d_p^{max}$ is accessible from s_0. Take a stationary policy Π which takes minimal action, i.e. it sets $\forall n : \tau_n = \tau_n^u$ and $\gamma_n = 0$ while $d_n^{arr} + X_n^p \leq d_{p[n]}^{max}$. While $\gamma = 0$, there is a strictly positive probability that the additional delay incurred in a round tour equals $\sum_{p=1}^{|P|}(\tau_p^u + X_p^{\text{p,max}} + X_p^{\text{s,max}}) - \overline{B} > 0$ (when we incur the maximum possible delay in each port and sea leg). Thus, possibly after multiple rounds, with positive probability we reach a port call n' for which $d_{p[n']}^{arr} + X_{n'}^p \geq d_{p[n']}^{max}$. From this state, *any* delay state for $d_{p[n']}^{dep}$ is accessible, by choosing $\gamma_{n'}$ appropriately.

Take a state $s_2 \in S_2$ that is recurrent under a certain policy Π'. A state corresponding to $d_{p[n']}^{dep} = d$ for some d is visited every round, and if s_2 is not accessible from such a state, then s_2 cannot be recurrent. Thus s_2 is accessible from a state $d_{p[n']}^{dep} = d$ for some d, say state s'. But we just showed that s' is accessible from s_0, and thus s_2 is also accessible from s_0. So, states that are recurrent under a policy communicate with s_0. This proves Weak Accessibility for our model, and thus that the long run average costs are independent of the starting state d_0^{dep} (Bertsekas 2007, p199, Prop 4.2.3). □

Proof of Theorem 9. In a finite horizon, the sequence of random variables

$$X = (X_1, \ldots, X_{2N-2}) = (X_1^s, X_2^p, X_2^s, \ldots, X_{N-1}^p, X_{N-1}^s, X_N^p)$$

yields a sequence of states and a sequence of actions:

$$(s_1, \ldots, s_{2N-2}) := (d_1^{dep}, d_2^{arr} + X_2^p, d_2^{dep}, \ldots, d_{N-1}^{arr} + X_{N-1}^p, d_{N-1}^{dep}, d_N^{arr} + X_{N-1}^p),$$

$$(a_1, \ldots, a_{2N-2}) := (\tau_1, \gamma_2, \tau_2, \ldots, \gamma_{N-1}, \tau_{N-1}, \gamma_N).$$

These latter sequences may depend on $\tilde{B}'$, the policy $\hat{\Pi}_{\tilde{B}'}$, and the random sequence X.

By Lemma 8 and the notation following that lemma, $\forall i \in \{1, \ldots, 2N-2\}$, the state in period i can be expressed as $s_i = s_{p[i],u[i]}[z_i, p'[i]; \tilde{B}']$. Since the port sequence is fixed, and arrivals and departures alternate, $p[i]$ and $u[i]$ are independent of $\tilde{B}'$ and the policy. We will show that under $\hat{\Pi}_{\tilde{B}'}$, the variables $z_i \in \mathbb{Z}$ and $p'[i] \in P$ are *independent* of $\tilde{B}'$, as long as $\tilde{B}' \in \Delta(\tilde{B})$. That means that for a fixed random sequence X, the delay in state i can be expressed as $d = z_i l + \tilde{B}_{p'[i]} - \tilde{B}_{p[i]}$, and that $z_i, p'[i]$ and $p[i]$ are independent of $\tilde{B}'$, as long as $\tilde{B}' \in \Delta(\tilde{B})$ and as long as we use the policy $\hat{\Pi}_{\tilde{B}'}$.

The proof is by induction in i. For $i = 1$, $d_1^{dep} = zl$ by Assumption 4. Setting $s_1 = s_{p[1],\text{dep}}[z, p[1]; \tilde{B}']$ yields this delay for all $\tilde{B}' \in \Delta(\tilde{B})$, which implies independence for $i = 1$. The induction step will be proved separately for odd, and for even i.

First assume the statement holds for some odd i (which corresponds to a departure delay for some port call n). Let $s_i = s_{p[n],\mathrm{dep}}[z,p;\tilde{B}']$, which corresponds to departing from port $p[n]$ with delay $d_n^{dep} = zl + \tilde{B}'_p - \tilde{B}'_{p[n]}$. Assume $\Pi^*_{\tilde{B}}(s_{p[n],\mathrm{dep}}[z,p;\tilde{B}]) = a_p[z',p';\tilde{B}]$. Note that an action of this form must be optimal since $d_n^{dep} + \mathcal{T}_n(d_n^{dep};B) \in Q_n$ by Lemma 8. Thus $a_i = \hat{\Pi}_{\tilde{B}'}(s_i) = a_p[z',p';\tilde{B}']$ is the action taken under $\hat{\Pi}_{\tilde{B}'}$. Note that z' and p' are independent of $\tilde{B}'$ by induction hypothesis, and by construction of $\hat{\Pi}_{\tilde{B}'}$. Write $X_n^s = lz^s$ and $X_{n+1}^p = lz^p$, where $z^s, z^p \in \mathbb{Z}$ by Assumption 4. Note that a_i denotes $\tau_n = z'l + \tilde{B}'_{p'} - \tilde{B}'_p$. Thus action a_i in state s_i yields $d_n^{dep} + \tau_n = (z+z')l + \tilde{B}'_{p'} - \tilde{B}'_{p[n]}$. By definition of the cumulative buffers $\tilde{B}'$ it holds that $\tilde{B}'_{p[n]} + B'_{p[n]} = \tilde{B}'_{p[n+1]} + z_n l$, with $z_n := \overline{B}/l \in \mathbb{Z}$ if $p[n+1] = 1$ and $z_n = 0$ otherwise. Thus, $d_n^{dep} + \tau_n - B'_{p[n]} + X_n^s = (z+z'+z^s)l + \tilde{B}'_{p'} - (\tilde{B}'_{p[n]} + B'_{p[n]}) = (z+z'+z^s+z_n)l + \tilde{B}'_{p'} - \tilde{B}'_{p[n+1]} = \tilde{z}l + \tilde{B}'_{p'} - \tilde{B}'_{p[n+1]}$, for some $\tilde{z} \in \mathbb{Z}$. Thus $d_{n+1}^{arr} = (\tilde{z}l + \tilde{B}'_{p'} - \tilde{B}'_{p[n+1]})^+$. We now distinguish two cases: 1) $\tilde{z}l + \tilde{B}_{p'} - \tilde{B}_{p[n+1]} \leq 0$ and 2) $\tilde{z}l + \tilde{B}_{p'} - \tilde{B}_{p[n+1]} > 0$. For Case 1, we have $\tilde{B}_{p[n+1]} - \tilde{B}_{p'} \geq \tilde{z}l$, which implies that $\left\lfloor (\tilde{B}_{p[n+1]} - \tilde{B}_{p'})/l \right\rfloor l \geq \tilde{z}l$, and thus, by $\tilde{B}' \in \Delta(\tilde{B})$, that $\tilde{B}'_{p[n+1]} - \tilde{B}'_{p'} \geq \tilde{z}l$. Hence, $d_{n+1}^{arr} + X_{n+1}^p = (\tilde{z}l + \tilde{B}'_{p'} - \tilde{B}'_{p[n+1]})^+ + lz^p = lz^p = lz^p + \tilde{B}'_{p[n+1]} - \tilde{B}'_{p[n+1]}$ which implies $s_{i+1} = s_{p[n+1],\mathrm{port}}[z'',p'';\tilde{B}']$ with $z'' = z^p$ and $p'' = p[n+1]$. For Case 2, we have $\tilde{B}_{p[n+1]} - \tilde{B}_{p'} < \tilde{z}l$, and thus $\left\lceil (\tilde{B}_{p[n+1]} - \tilde{B}_{p'})/l \right\rceil l \leq \tilde{z}l$ which implies (since $\tilde{B}' \in \Delta(\tilde{B})$) that $\tilde{B}'_{p[n+1]} - \tilde{B}'_{p'} \leq \tilde{z}l$. Thus $d_{n+1}^{arr} + X_{n+1}^p = \tilde{z}l + \tilde{B}'_{p'} - \tilde{B}'_{p[n+1]} + z^p l = (\tilde{z} + z^p)l + \tilde{B}'_{p'} - \tilde{B}'_{p[n+1]}$, implying $s_{i+1} = s_{p[n+1],\mathrm{port}}[z'',p'';\tilde{B}']$ with $z'' = \tilde{z} + z^p$ and $p'' = p'$. Note that $\tilde{z} = z + z' + z^s + z_n$ is independent of $\tilde{B}'$ by induction hypothesis, and thus case checking is independent of $\tilde{B}'$. (It is thus essential that $\tilde{B}$ can be used for case checking.) Hence, z'' and p'' are independent of $\tilde{B}'$, which proves the result for $i+1$.

Now assume the result holds for some even i' (which corresponds to a port delay for some port call n). We will reuse some notation. Let $s_{i'} = s_{p[n],\mathrm{port}}[z,p;\tilde{B}']$, which corresponds to being in port $p[n]$ with delay $d_n^{arr} + X_n^p = zl + \tilde{B}'_p - \tilde{B}'_{p[n]}$. Assume $\Pi^*_{\tilde{B}}(s_{p[n],\mathrm{port}}[z,p;\tilde{B}]) = a_p[z',p';\tilde{B}]$. Note that an action of this form must be optimal by Lemma 8. Let $a_{i'} = \hat{\Pi}_{\tilde{B}'}(s_i) = a_p[z',p';\tilde{B}']$, and note that z' and p' are independent of $\tilde{B}'$ by induction hypothesis. We find $d_n^{dep} = (d_n^{arr} + X_n^p + \gamma_n)^+$. We have $d_n^{arr} + X_n^p + \gamma_n = (zl + \tilde{B}'_p - \tilde{B}'_{p[n]}) + (z'l + \tilde{B}'_{p'} - \tilde{B}'_p) = \tilde{z}l + \tilde{B}'_{p'} - \tilde{B}'_{p[n]}$, where $\tilde{z} = z + z' \in \mathbb{Z}$. Distinguish between two cases: 1) $\tilde{z}l + \tilde{B}_{p'} - \tilde{B}_{p[n]} \leq 0$ and 2) $\tilde{z}l + \tilde{B}_{p'} - \tilde{B}_{p[n]} > 0$. For the first case, we obtain in a similar fashion as before that $\tilde{z}l + \tilde{B}'_{p'} - \tilde{B}'_{p[n]} \leq 0$. Thus $d_n^{dep} = 0$, which implies $s_{i'+1} = s_{p[n],\mathrm{dep}}[z'',p'';\tilde{B}']$ with $p'' = p[n]$ and $z'' = 0$. For the second case, we obtain $\tilde{z}l + \tilde{B}'_{p'} - \tilde{B}'_{p[n]} > 0$ and thus $d_n^{dep} = \tilde{z}l + \tilde{B}'_{p'} - \tilde{B}'_{p[n]}$ implying $s_{i'+1} = s_{p[n],\mathrm{dep}}[z'',p'';\tilde{B}']$ with $p'' = p'$ and $z'' = \tilde{z}$. Note that $\tilde{z}$ is independent of $\tilde{B}'$, thus so is the case checking. Thus p'' and z'' are independent of $\tilde{B}'$, which proves the statement for $i'+1$. This completes the proof by induction.

Thus for all $\tilde{B}' \in \Delta(\tilde{B})$ under $\hat{\Pi}_{\tilde{B}'}$, for each sequence X, there are $z, z' \in \mathbb{Z}, p, p' \in P$ such that:

$$d_n^{dep} = zl + \tilde{B}'_p - \tilde{B}'_{p[n]}, \quad \tau_n = z'l + \tilde{B}'_{p'} - \tilde{B}'_p$$

Here $z, z' \in \mathbb{Z}, p, p' \in P$ are independent of $\tilde{B}'$, provided policy $\hat{\Pi}_{\tilde{B}'}$ is used. It follows that $\mathcal{D}_{p[n]}^{dep}(d_n^{dep})$ for the sequence X under $\hat{\Pi}_{\tilde{B}'}$ is affine in $\tilde{B}'$. Indeed, when $p \neq p[n]$ since $l\left\lfloor(\tilde{B}_p - \tilde{B}_{p[n]})/l\right\rfloor \leq (\tilde{B}'_p - \tilde{B}'_{p[n]}) \leq l\left\lceil(\tilde{B}_p - \tilde{B}_{p[n]})/l\right\rceil$ and since the delay costs $\mathcal{D}_{p[n]}^{dep}(d_n^{dep})$ are piecewise linear with breakpoints at zl for $z \in \mathbb{Z}$ by Assumption 4, there exist constants c_1 and c_2 such that $\mathcal{D}_{p[n]}^{dep}(zl + \tilde{B}'_p - \tilde{B}'_{p[n]}) = c_1 + c_2(zl + \tilde{B}'_p - \tilde{B}'_{p[n]})$ for all $\tilde{B}' \in \Delta(\tilde{B})$, which is affine in $\tilde{B}'$. When $p = p[n]$, we have $\mathcal{D}_{p[n]}^{dep}(d_n^{dep}) = \mathcal{D}_{p[n]}^{dep}(zl)$, which is independent of $\tilde{B}'$, and hence, affine. For the sailing costs $\mathcal{F}_{p[n]}(\tau_n)$, if $p \neq p'$, we have $l\left\lfloor(\tilde{B}_{p'} - \tilde{B}_p)/l\right\rfloor \leq (\tilde{B}'_{p'} - \tilde{B}'_p) \leq l\left\lceil(\tilde{B}_{p'} - \tilde{B}_p)/l\right\rceil$. Thus, since $\tau_n = z'l + \tilde{B}_{p'} - \tilde{B}_p$ is feasible for $\tilde{B}$, $\tau'_n = z'l + \tilde{B}'_{p'} - \tilde{B}'_p$ is also feasible for $\tilde{B}'$. Moreover, since $\mathcal{F}_{p[n]}(\tau_n)$ is piecewise linear function with breakpoints at zl for $z \in \mathbb{Z}$ by Assumption 4, $\mathcal{F}_{p[n]}(z'l + \tilde{B}'_{p'} - \tilde{B}'_p)$ is affine in $\tilde{B}'$ for all $\tilde{B}' \in \Delta(\tilde{B})$. In a very similar fashion it can be shown that, for the sequence X under $\hat{\Pi}_{\tilde{B}'}$, the arrival delay $\mathcal{D}_{p[n]}^{arr}(d_n^{arr})$, and the costs of the extreme action $c^e\gamma_n$, are affine in $\tilde{B}'$. Because this holds for all n, the total costs incurred over the periods $\{1, \ldots, N = R|P| + 1\}$ for the sequence X under $\hat{\Pi}_{\tilde{B}'}$ are affine in $\tilde{B}' \in \Delta(\tilde{B})$.

The total *expected* costs under $\hat{\Pi}_{\tilde{B}'}$ over the periods $\{1, \ldots, N = R|P| + 1\}$ are the expectation of the costs for each sequence over all sequences, and they are affine in $\tilde{B}'$ because taking a linear combination over affine functions yields an affine function. The average expected costs $\hat{\mathcal{C}}(\tilde{B}')$ under $\hat{\Pi}_{\tilde{B}'}$ are obtained by dividing the total costs incurred over $\{1, \ldots, N = R|P| + 1\}$ by R, taking the limit $R \to \infty$. This limit exists in $\mathbb{R}$ for all $\tilde{B}'$ since it corresponds to the average costs of a stationary policy in a finite-state Markov Process (see the proof of Corollary 1). Moreover, $\hat{\Pi}_{\tilde{B}'}$ is affine in $\tilde{B}'$, since the limit of functions that are affine in $\tilde{B}'$ is affine in $\tilde{B}'$, provided the limit exists for each $\tilde{B}'$. This establishes the existence of g_0 and $g = (g_1, \ldots, g_{|P|})$ such that $\forall \tilde{B}' \in \Delta(\tilde{B}) : \hat{\mathcal{C}}(\tilde{B}') = g_0 + \sum_{p \in P} g_p \tilde{B}'_p$.

We now show that g must be a subgradient at $\tilde{B}$. For some arbitrary $\tilde{B}'' \in \tilde{\mathcal{B}}$, let $\tilde{B}(x) = \tilde{B} + x(\tilde{B}'' - \tilde{B})$. We have $\mathcal{C}^*(\tilde{B}(0)) = \mathcal{C}^*(\tilde{B})$, and $\mathcal{C}^*(\tilde{B}) = \hat{\mathcal{C}}(\tilde{B}) = g_0 + \sum_{p \in P} g_p \tilde{B}_p$ because $\hat{\Pi}_{\tilde{B}}$ is optimal for $\tilde{B}$ by construction. Because $\tilde{B}$ is completely fractional, there is some $\epsilon > 0$ such that $\tilde{B}(-\epsilon) = \tilde{B} - \epsilon(\tilde{B}'' - \tilde{B}) \in \Delta(\tilde{B})$. Thus $\hat{\Pi}_{\tilde{B}(-\epsilon)}$ is feasible for $\tilde{B}(-\epsilon)$, and we obtain $\mathcal{C}^*(\tilde{B}(-\epsilon)) \leq \hat{\mathcal{C}}(\tilde{B}(-\epsilon)) = g_0 + \sum_{p \in P} g_p \tilde{B}(-\epsilon)_p = g_0 + \sum_{p \in P} g_p[\tilde{B}_p - \epsilon(\tilde{B}''_p - \tilde{B}_p)] = \mathcal{C}^*(\tilde{B}) - \epsilon \sum_{p \in P} g_p(\tilde{B}''_p - \tilde{B}_p)$. Now, by Theorem 5 and Corollary 1, $\mathcal{C}^*(B)$ is convex in $B \in \mathcal{B}$, and since $\tilde{B}$ is obtained by an affine transformation of B, $\mathcal{C}^*(\tilde{B})$ is convex in $\tilde{B} \in \tilde{\mathcal{B}}$. Thus $\mathcal{C}^*(\tilde{B}(x))$ is convex in x, which implies that $(1 + \epsilon)\mathcal{C}^*(\tilde{B}(0)) \leq$

$\mathcal{C}^*(\tilde{B}(-\epsilon)) + \epsilon\mathcal{C}^*(\tilde{B}(1))$. Thus $\epsilon\mathcal{C}^*(\tilde{B}'') = \epsilon\mathcal{C}^*(\tilde{B}(1)) \geq (1+\epsilon)\mathcal{C}^*(\tilde{B}(0)) - \mathcal{C}^*(\tilde{B}(-\epsilon)) \geq (1+\epsilon)\mathcal{C}^*(\tilde{B}) - \left[\mathcal{C}^*(\tilde{B}) - \epsilon\sum_{p\in P} g_p(\tilde{B}''_p - \tilde{B}_p)\right] = \epsilon\left[\mathcal{C}^*(\tilde{B}) + \sum_{p\in P} g_p(\tilde{B}''_p - \tilde{B}_p)\right]$. Thus we find $\forall \tilde{B}'' \in \tilde{\mathcal{B}} : \mathcal{C}^*(\tilde{B}'') \geq \mathcal{C}^*(\tilde{B}) + \sum_{p\in P} g_p(\tilde{B}''_p - \tilde{B}_p)$, which is precisely the subgradient inequality. Thus g is a subgradient at $\tilde{B}$.

Additionally, for any $\tilde{B}' \in \Delta(\tilde{B})$, we know that $\mathcal{C}^*(\tilde{B}') \leq \hat{\mathcal{C}}(\tilde{B}')$ since $\hat{\Pi}_{\tilde{B}'}$ is a feasible policy for $\tilde{B}'$. But $\hat{\mathcal{C}}(\tilde{B}') = g_0 + \sum_{p\in P} g_p\tilde{B}'_p = \mathcal{C}^*(\tilde{B}) + \sum_{p\in P} g_p(\tilde{B}'_p - \tilde{B}_p) \leq \mathcal{C}^*(\tilde{B}')$, where the second equality follows because $\mathcal{C}^*(\tilde{B}) = g_0 + \sum_{p\in P} g_p\tilde{B}_p$, and the inequality is the subgradient inequality at $\tilde{B}$ that we just proved. Combining these inequalities yields $\forall \tilde{B}' \in \Delta(\tilde{B}) : \mathcal{C}^*(\tilde{B}') = \hat{\mathcal{C}}(\tilde{B}')$, completing the proof. □

Proof of Corollary 2. Let $\tilde{B}$ be completely fractional and let g denote the subgradient from Theorem 9. For arbitrary $\tilde{B}'' \in \tilde{\mathcal{B}}$, we obtain the subgradient inequality at $\tilde{B}' \in \Delta(\tilde{B})$:

$$\mathcal{C}^*(\tilde{B}'') \geq \mathcal{C}^*(\tilde{B}) + \sum_{p\in P} g_p(\tilde{B}''_p - \tilde{B}_p) = \mathcal{C}^*(\tilde{B}') + \sum_{p\in P} g_p(\tilde{B}''_p - \tilde{B}'_p)$$

Here, the inequality holds because g is a subgradient at $\tilde{B}$, and the equality because $\mathcal{C}^*(\tilde{B}') = \hat{\mathcal{C}}(\tilde{B}') = \mathcal{C}^*(\tilde{B}) + \sum_{p\in P} g_p(\tilde{B}'_p - \tilde{B}_p)$ for $\tilde{B}' \in \Delta(\tilde{B})$ by Theorem 9. But the subgradient inequality for g at $\tilde{B}'$ shows that g is a subgradient at $\tilde{B}'$. □

Proof of Lemma 10. Let $\tilde{B} = (\tilde{B}_1, \ldots, \tilde{B}_{|P|})$ be completely fractional, and $\tilde{B}' \in \Delta(\tilde{B})$. Write $\tilde{B}_p = l(z_p + x_p)$, with $z_p \in \{0, 1, \ldots, z_{\overline{B}} - 1\}$ and $0 \leq x_p < 1$. (Setting $x_p = 1$ is never required because $0 \leq \tilde{B}_p < \overline{B}$ since $\tilde{B}$ is completely fractional.) Note that by definition $\tilde{B}_1 = \tilde{B}'_1 = 0$, and that $\lfloor \tilde{B}_p/l \rfloor = z_p$. Hence, the constraints in (5.17) concerning $p = 1$ simplify to: $\forall p' \in P : lz_{p'} \leq \tilde{B}'_{p'} \leq lz_{p'} + l$. Thus for $p \in P$ there $\exists x'_p \in [0,1]$ such that $\tilde{B}'_p = l(z_p + x'_p)$.

Let $f : P \to P$ be the unique permutation of P such that $p' > p \to x_{f(p')} > x_{f(p)}$, and let f^{-1} denote its inverse. Uniqueness follows because $\forall p, p' \in P$ with $p \neq p'$ it holds that $x_p \neq x_{p'}$, since $x_p = x_{p'}$ would contradict that $\tilde{B}$ is completely fractional. Thus $f^{-1}(p') > f^{-1}(p)$ if and only if $x_{p'} > x_p$. We find:

$$\begin{aligned} l\left\lfloor \left(\tilde{B}_{p'} - \tilde{B}_p\right)/l \right\rfloor &= l\left\lfloor z_{p'} + x_{p'} - (z_p + x_p) \right\rfloor \\ &= l\left\lfloor z_{p'} - z_p + x_{p'} - x_p \right\rfloor \\ &= \begin{cases} l(z_{p'} - z_p) & \text{if } f^{-1}(p') > f^{-1}(p) \\ l(z_{p'} - z_p) - l & \text{if } f^{-1}(p') < f^{-1}(p). \end{cases} \end{aligned}$$

Note that $\tilde{B}'_{p'} - \tilde{B}'_p = l(z_{p'} - z_p + x'_{p'} - x'_p)$. Thus the condition $l\lfloor(\tilde{B}_{p'} - \tilde{B}_p)/l\rfloor \leq \tilde{B}'_{p'} - \tilde{B}'_p \leq l\lfloor(\tilde{B}_{p'} - \tilde{B}_p)/l\rfloor + l$ for $p, p' \in P$ that is part of the definition of $\Delta(\tilde{B})$ in (5.17) is equivalent to the following condition on $x'_p, x'_{p'}$:

$$\begin{aligned} 0 \leq x'_{p'} - x'_p \leq 1 \qquad & \text{if } f^{-1}(p') > f^{-1}(p) \\ -1 \leq x'_{p'} - x'_p \leq 0 \qquad & \text{if } f^{-1}(p') < f^{-1}(p). \end{aligned} \tag{5.24}$$

Thus, for any $i, i' \in \{1, \ldots, |P|\}$ with $i' > i$, substituting $p' = f(i')$ and $p = f(i)$ in (5.24) yields $x'_{f(i')} - x'_{f(i)} \geq 0$ since $f^{-1}(f(i')) > f^{-1}(f(i))$. Thus $x'_{f(1)} \leq x'_{f(2)} \leq \ldots \leq x'_{f(|P|)}$.

Conversely, assume $0 \leq x'_{f(1)} \leq x'_{f(2)} \leq \ldots \leq x'_{f(|P|)} \leq 1$. Then (5.24) is satisfied, since $\forall p \in P : 0 \leq x_p \leq 1$ implies that $-1 \leq x_{p'} - x_p \leq 1$, and the other inequalities of (5.24) follow by reversing the above argument. □

Lemma 12. *For any completely fractional $\tilde{B}'$, $\Delta(\tilde{B}')$ is the convex hull of $\{\tilde{B}^i | i \in \{1, \ldots, |P|\}\}$, with $\tilde{B}^i$ as defined in Section 5.5.2.*

Proof of Lemma 12. Lemma 10 provides an alternative formulation of $\Delta(\tilde{B})$. Hence, the extreme points induced by $0 \leq x_{f(1)} \leq x_{f(2)} \leq \ldots \leq x_{f(|P|)} \leq 1$ are the extreme points of $\Delta(\tilde{B})$. It is easy to verify that the extreme points are exactly given by $\tilde{B}^i$ for $i \in \{1, \ldots, |P|\}$. Namely, each extreme point will have $x_p = 0$ or $x_p = 1$ for $1 \leq p \leq |P|$. Furthermore, the ordering induced by the bijection f ensures that $x_p = 1$ can only be valid if $x_{p'} = 1$ for all p' such that $f(p') > f(p)$. Since $0 \leq x_{f(1)} \leq x_{f(2)} \leq \ldots \leq x_{f(|P|)} \leq 1$ only contains linear inequalities, the feasible region is a polyhedron. This polyhedron is clearly bounded. Combining this with the fact that the extreme points of the polyhedron are $\tilde{B}^i$, $i \in \{1, \ldots, |P|\}$, the convex hull of $\{\tilde{B}^i | i \in \{1, \ldots, |P|\}$ must equal $\Delta(\tilde{B})$. □

Chapter 6

Summary and conclusions

This dissertation provides methods for designing and improving liner shipping networks. Liner shipping routes are cyclic, meaning that the first and last port of a route are the same. Furthermore, routes are usually serviced once a week, such that the duration of a route determines the number of ships needed on the route. Different vessel types exist, but not every vessel may be able to enter each port because of draft restrictions. Each vessel type has a feasible speed range, hence each vessel should sail with a speed in between a minimum and a maximum speed. The executed speed has a large influence on the route costs, since bunker consumption is usually convex in the speed. Some of these aspects are very typical for liner shipping and all of them should be taken into account when designing the route network.

In Chapters 2 and 3, methods are proposed to design networks satisfying the above properties. Since this problem is very complex, the ports are first clustered to obtain smaller problem instances. The route network is split into two parts: a main network on which the most important ports are visited and a feeder network used to transport containers from the most important ports to the other ports in the cluster. Chapter 2 makes use of a genetic algorithm based method to generate the routes on the main network, while Chapter 3 uses an iterative MIP based approach. Both chapters show that our methods provide networks that yield higher profits than existing networks. In Chapter 3 we use benchmark datasets from the literature and compare our results with the current best results known. Although, our networks are constrained to satisfy a hub-and-feeder structure, we are still able to generate networks with more than 25% higher profits than the current best networks for the Europe-Asia instance of the benchmark data.

Chapters 4 and 5 consider a problem in which the routes and frequency are already determined. Hence, the order and duration of the route is already fixed. In these chapters we study the effect of small disturbances incurred during the execution of the route. In general, disruptions can be handled in two ways: additional time (buffer) can be added to the schedule during the construction of the timetable in order to prevent delays and recovery actions can be performed during the execution of the timetable in order to recover from the incurred delay. However, during the timetable construction phase, the exact delays to be incurred during the execution of the route are unknown, while the timetable cannot be changed anymore in order to prevent delays as soon as it is published and the ship starts sailing the route. Hence, simultaneously determining the buffer time to be allocated to each sea leg and the recovery action to be taken when delays are incurred during execution of the route is a complex problem.

In Chapter 4 we provide a mixed integer programming formulation for this problem. Because this MIP model will take a long time to solve in many instances, we also provide several heuristics to solve the problem. The heuristics all use the observation that the problem to determine optimal recovery actions given a certain buffer time allocation can be formulated as a Markov decision process and hence be determined efficiently. The heuristics either start without any buffer allocated and iteratively add one unit of buffer in a greedy way, or start with an initial feasible buffer allocation and iteratively exchange units of buffer between two ports. Chapter 4 shows that the heuristics provide results that are close to (and often even equal to or better than) the best solution found solving the MIP model with a time limit of one hour. In most instances, increasing the sailing speed is the only recovery action available, but we also find results for instances where some ports might be skipped in order to gain time. Because the state space of the Markov decision processes is much larger than the state space without this possibility, the MDP can become quite difficult to solve. This results in long computational times for some of the heuristics, since they rely on repeatedly solving multiple of these MDPs. However, heuristics using the value function of the MDP to estimate the gain of a change in the buffer allocation are fast and provide good results.

Finally, in Chapter 5 we consider the same problem but under the assumption that all cost functions are convex. This eliminates the action to skip a port, leaving only the option to change the sailing speed. In this chapter, we prove some important theoretical properties of the problem. The most important result is the convexity of the cost function in the buffer allocation. Although it follows directly from the convexity result that subgradients can be used to find an optimal solution, it is not directly apparent for this problem how subgradients can be obtained. We provide an extensive analysis of the problem and explain a procedure to find the subgradients. Using these subgradients, we are able to provide an efficient exact solution algorithm.

References

Agarwal, R., Ö. Ergun. 2008. Ship Scheduling and Network Design for Cargo Routing in Liner Shipping. *Transportation Science* **42**(2) 175–196.

AhmadBeygi, S., A. Cohn, M. Lapp. 2010. Decreasing airline delay propagation by re-allocating scheduled slack. *IIE Transactions* **42**(7) 478–489.

Aktürk, M.S., A. Atamtürk, S. Gürel. 2014. Aircraft Rescheduling with Cruise Speed Control **62**(4) 829–845.

Álvarez, J.F. 2009. Joint routing and deployment of a fleet of container vessels. *Maritime Economics & Logistics* **11**(2) 186–208.

Arikan, U., S. Gürel, M. Selim Aktürk. 2016. Integrated aircraft and passenger recovery with cruise time controllability. *Annals of Operations Research* **236**(2) 295–317.

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

Bertsekas, D.P. 2007. *Dynamic Programming and Optimal Control - Volume II*. 3rd ed. Athena Scientific, Belmont, Massachusetts.

Binder, A., T. Albrecht. 2012. Predictive energy-efficient running time control for metro lines. *CASPT12 - Conference on Advanced Systems for Public Transport*.

Brouer, B.D. 2015. Correction sheet results Brouer, Desaulniers and Pisinger. `https://github.com/blof/LINERLIB/tree/master/results/BrouerDesaulniersPisinger2014`. Accessed: 11 January 2016.

Brouer, B.D., J.F. Álvarez, C.E.M. Plum, D. Pisinger, M.M. Sigurd. 2014a. A Base Integer Programming Model and Benchmark Suite for Liner-Shipping Network Design. *Transportation Science* **48**(2) 281–312.

Brouer, B.D., G. Desaulniers, D. Pisinger. 2014b. A matheuristic for the liner shipping network design problem. *Transportation Research Part E* **72** 42–59.

Brouer, B.D., J. Dirksen, D. Pisinger, C.E.M. Plum, B. Vaaben. 2013. The Vessel Schedule Recovery Problem (VSRP) - a MIP model for handling disruptions in liner shipping. *European Journal of Operational Research* **224**(2) 362–374.

Cariou, P. 2011. Is slow steaming a sustainable means of reducing CO_2 emissions from container shipping? *Transportation Research Part D* **16**(3) 260–264.

Chiraphadhanakul, V., C. Barnhart. 2013. Robust flight schedules through slack re-allocation. *EURO Journal on Transportation and Logistics* **2** 277–306.

Christiansen, M., K. Fagerholt, B. Nygreen, D. Ronen. 2007. Maritime Transportation. C. Barnhart, G. Laporte, eds., *Handbook in OR & MS*, vol. 14. Elsevier B.V., 189–284.

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

Christiansen, M., K. Fagerholt, D. Ronen. 2004. Ship Routing and Scheduling: Status and Perspectives. *Transportation Science* **38**(1) 1–18.

Cicerone, S., G. D'Angelo, G. Di Stefano, D. Frigiono, A. Navarra, M. Schachtebeck, A. Schöbel. 2009. Recoverable Robustness in Shunting and Timetabling. R.K. Ahuja, R.H. Möhring, C.D. Zaroliagis, eds., *Robust and Online Large-Scale Optimization*, *Lecture Notes in Computer Science*, vol. 5868. Springer, 28–60.

Cicerone, S., G. Di Stefano, M. Schachtebeck, A. Schöbel. 2012. Multi-stage recovery robustness for optimization problems: A new concept for planning under disturbances. *Information Sciences* **190** 107–126.

Clausen, J., A. Larsen, J. Larsen, N.J. Rezanova. 2010. Disruption management in the airline industry - Concepts, models and methods. *Computers & Operations Research* **37**(5) 809–821.

Corbett, J.J., H. Wang, J.J. Winebrake. 2009. The effectiveness and costs of speed reductions on emissions from international shipping. *Transportation Research Part D* **14**(8) 593–598.

Corman, F., A. D'Ariano, D. Pacciarelli, M. Pranzo. 2010. A tabu search algorithm for rerouting trains during rail operations. *Transportation Research Part B* **44**(1) 175–192.

Cucala, A.P., A. Fernández, C. Sicre, M. Domínguez. 2012. Fuzzy optimal schedule of high speed train operation to minimize energy consumption with uncertain delays and driver's behavioral response. *Engineering Applications of Artificial Intelligence* **25** 1548–1557.

Cullinane, K., M. Khanna. 1999. Economies of Scale in Large Container Ships. *Journal of Transport Economics and Policy* **33**(2) 185–207.

Di Francesco, M., M. Lai, P. Zuddas. 2014. A multi-scenario model for empty container repositioning with uncertain demand. *International Journal of Services and Operations Management* **19**(2) 212–228.

Du, Y., Q. Meng, Y. Wang. 2015. Budgeting the fuel consumption of a container ship over a round voyage via robust optimization. *Transportation Research Record: Journal of the Transportation Research Board* **2477** 68–75.

Duran, A.S., S. Gürel, M.S. Aktürk. 2015. Robust Airline Scheduling with Controllable Cruise Times and Chance Constraints. *IIE Transactions* **47**(1) 64–83.

Engelhardt-Funke, O., M. Kolonko. 2004. Analysing stability and investments in railway networks using advanced evolutionary algorithms. *International Transactions in Operational Research* **11**(4) 381–394.

Fagerholt, K. 1999. Optimal fleet design in a ship routing problem. *International Transactions in Operational Research* **6**(5) 453–464.

Fagerholt, K. 2004. Designing optimal routes in a liner shipping problem. *Maritime Policy & Management* **31**(4) 259–268.

Fagerholt, K., G. Laporte, I. Norstad. 2010. Reducing fuel emissions by optimizing speed on shipping routes. *Journal of the Operational Research Society* **61**(3) 523–529.

Fagerholt, K., H. Lindstad. 2007. Turborouter: An Interactive Optimisation-Based Decision Support System for Ship Routing and Scheduling. *Maritime Economics & Logistics* **9** 214–233.

Fischetti, M., D. Salvagnin, A. Zanette. 2009. Fast Approaches to Improve the Robustness of a Railway Timetable. *Transportation Science* **43**(3) 321–335.

Francesetti, D.C., A.D. Foschi. 2002. The Impact of Hub and Spokes Networks in the Mediterranean Peculiarity. *IAME Annual Conference Panama*.

Gelareh, S., N. Maculan, P.Mahey, P.N. Monemi. 2013. Hub-and-spoke network design and fleet deployment for string planning of liner shipping. *Applied Mathematical Modelling* **37**(5) 3307–3321.

Gelareh, S., D. Pisinger. 2011. Fleet deployment, network design and hub location of liner shipping companies. *Transportation Research Part E* **47**(6) 947–964.

Gong, C., S. Zhang, F. Zhang, J. Jiang, X. Wang. 2014. An Integrated Energy-Efficient Operation Methodology for Metro Systems Based on a Real Case of Shanghai Metro Line One. *Energies* **7** 7305–7329.

Hassannayebi, E., A. Sajedinejad, S. Mardani. 2014. Urban rail transit planning using a two-stage simulation-based optimization approach. *Simulation Modelling Practice and Theory* **49** 151–166.

Hillier, F.S., G.J. Lieberman. 2001. *Introduction to Operations Research*. 7th ed. McGraw-Hill Education, New York, 1053–1078.

Hsu, C.I., Y.P. Hsieh. 2007. Routing, ship size, and sailing frequency decision-making for a maritime hub-and-spoke container network. *Mathematical and Computer Modelling* **45**(7-8) 899–916.

Huang, Y.F. J.K. Hu, B. Yang. 2015. Liner service network design and fleet deployment with empty container repositioning. *Computers & Industrial Engineering* **89** 116–124.

Imai, A., K. Shintani, S. Papadimitriou. 2009. Multi-port vs. Hub-and-Spoke port calls by containerships. *Transportation Research Part E* **45**(5) 740–757.

Karsten, C.V., D. Pisinger, S. Ropke, B.D. Brouer. 2015. The time constrained multi-commodity network flow problem and its application to liner shipping network design. *Transportation Research Part E* **76** 122–138.

Kontovas, C.A. 2014. The Green Ship Routing and Scheduling Problem (GSRSP): A conceptual approach. *Transportation Research Part D* **31** 61–69.

Kroon, L.G., R. Dekker, M.C.J.M. Vromans. 2007. Cyclic Railway Timetabling: A Stochastic Optimization Approach. F. Geraets, L.G. Kroon, A. Schöbel, D. Wagner, C.D. Zaroliagis, eds., *Algorithmic Methods for Railway Optimization*, *Lecture Notes in Computer Science*, vol. 4359. Springer, 41–66.

Kroon, L.G., G. Maróti, M.J. Retel Helmrich, M.J.C.M. Vromans, R. Dekker. 2008. Stochastic improvement of cyclic railway timetables. *Transportation Research Part B* **42**(6) 553–570.

Lachner, S.A., V.A.G. Boskamp. 2011. Routing and scheduling in liner shipping with multi-start local search heuristics. Master's thesis, Erasmus University Rotterdam.

Lan, S., J.P. Clarke, C. Barnhart. 2006. Planning for Robust Airline Operations: Optimizing Aircraft Routings and Flight Departure Times to Minimize Passenger Disruptions. *Transportation Science* **40**(1) 15–28.

Lawrence, S.A. 1972. International Sea Transport: The Years Ahead.

Li, C., X. Qi, C.Y. Lee. 2015a. Disruption Recovery for a Vessel in Liner Shipping. *Transportation Science* **49**(4) 900–921.

Li, C., X. Qi, D. Song. 2015b. Real-time schedule recovery in liner shipping service with regular uncertainties and disruption events. *Transportation Research Part B* doi:10.1016/j.trb.2015.10.004.

Liebchen, C., M. Schachtebeck, A. Schöbel, S. Stiller, A. Prigge. 2010. Computing delay resistant railway timetables. *Computers & Operations Research* **37**(5) 857–868.

LINERLIB. 2014. Liner shipping network desing benchmark instances. `https://github.com/blof/LINERLIB/tree/master/data`. Version 1.1.

Liu, Z., Q. Meng, S. Wang, Z. Sun. 2014. Global intermodal liner shipping network design. *Transportation Research Part E* **61** 28–39.

Long, Y., E.P. Chew, L.H. Lee. 2015. Sample average approximation under non-i.i.d. sampling for stochastic empty container repositioning problem. *OR Spectrum* **37** 389–405.

Maher, S.J. 2016. Solving the Integrated Airline Recovery Problem Using Column-and-Row Generation. *Transportation Science* **50**(1) 216–239.

Meng, Q., S. Wang, H. Andersson, K. Thun. 2014. Containership Routing and Scheduling in Liner Shipping: Overview and Future Research Directions. *Transportation Science* **48**(2) 265–280.

Mulder, J., R. Dekker. 2014. Methods for strategic liner shipping network design. *European Journal of Operational Research* **235**(2) 367–377.

Nagy, G., S. Salhi. 2005. Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *European Journal of Operational Research* **162**(1) 126–141.

Norlund, E.K., I. Gribkovskaia, G. Laporte. 2015. Supply vessel planning under cost, environment and robustness considerations **57** 271–281.

Notteboom, T.E. 2004. A carrier's perspective on container network configuration at sea and on land. *Journal of International Logistics and Trade* **1**(2) 65–87.

Notteboom, T.E. 2006. The Time Factor in Liner Shipping Services. *Maritime Economics & Logistics* **8**(1) 19–39.

Notteboom, T.E., B. Vernimmen. 2009. The effect of high fuel costs on liner service configuration in container shipping. *Journal of Transport Geography* **17**(5) 325–337.

Pantuso, G., K. Fagerholt, L.M. Hvattum. 2014. A survey on maritime fleet size and mix problems. *European Journal of Operational Research* **235**(2) 341–349.

Pedrielli, G., L.H. Lee, S.H. Ng. 2015. Optimal bunkering contract in a buyer-seller supply chain under price and consumption uncertainty. *Transportation Research Part E* **77** 77–94.

Petersen, J.D., G. Sölveling, J.P. Clarke, E.L. Johnson, S. Shebalov. 2012. An Optimization Approach to Airline Integrated Recovery. *Transportation Science* **46**(4) 482–500.

Plum, C.E.M., P.N. Jensen, D. Pisinger. 2014a. Bunker purchasing with contracts. *Maritime Economics & Logistics* **16**(4) 418–435.

Plum, C.E.M., D. Pisinger, J.J. Salazar-González, M.M. Sigurd. 2014b. Single liner shipping service design. *Computers & Operations Research* **45** 1–6.

Plum, C.E.M., D. Pisinger, M.M. Sigurd. 2014c. A service flow model for the liner shipping network design problem. *European Journal of Operational Research* **235**(2) 378–386.

Polat, O., H.O. Günther, O. Kulak. 2014. The feeder network design problem: Application to container services in the Black Sea region. *Maritime Economics & Logistics* **16**(3) 343–369.

Powell, B.J., A.N. Perkins. 1997. Fleet deployment optimization for liner shipping: an integer programming model. *Maritime Policy & Management* **24**(2) 183–192.

Psaraftis, H.N., C.A. Kontovas. 2013. Speed models for energy-efficient maritime transportation: A taxonomy and survey. *Transportation Research Part C* **26** 331–351.

Psaraftis, H.N., C.A. Kontovas. 2014. Ship speed optimization: Concepts, models and combined speed-routing scenarios. *Transportation Research Part C* **44** 52–69.

Qi, X., D.P. Song. 2012. Minimizing fuel emissions by optimizing vessel schedules in liner shipping with uncertain port times. *Transportation Research Part E* **48**(4) 863–880.

Rana, K., R.G. Vickson. 1991. Routing Container Ships Using Lagrangean Relaxation and Decomposition. *Transportation Science* **25**(3) 201–214.

Rodrigue, J.P., C. Claude, B.Slack. 2013. *The geography of transport systems*. 3rd ed. Routledge.

Ronen, D. 1983. Cargo ships routing and scheduling: Survey of models and problems. *European Journal of Operational Research* **12**(2) 119–126.

Ronen, D. 1993. Ship scheduling: The last decade. *European Journal of Operational Research* **71**(3) 325–333.

Rosenberger, J.M., E.L. Johnson, G.L. Nemhauser. 2003. Rerouting Aircraft for Airline Recovery. *Transportation Science* **37**(4) 408–421.

Sanchez, R.J., G. Wilmsmeier. 2011. Liner Shipping Networks and Market Concentration. K. Cullinane, ed., *International Handbook of Maritime Economics*. Edward Elgar Publishing Inc., 162–206.

SeaRates. 2015. Reference quide, port to port distance. `https://www.searates.com/reference/portdistance`. Accessed: 19 October 2015.

Sheng, X., E.P. Chew, L.H. Lee. 2015. (s,S) policy model for liner shipping refueling and sailing speed optimization problem. *Transportation Research Part E* **76** 76–92.

Song, D., J. Zhang, J. Carter, T. Field, J. Marshall, J. Polak, K. Schumacher, P. Sinha-Ray, J. Woods. 2007. On cost-efficiency of the global container shipping network. *Maritime Policy & Management* **32**(1) 15–30.

Stopford, M. 2009. *Maritime Economics*. 3rd ed. Routledge.

Tierney, K., D. Pacino, R.M. Jensen. 2014. On the complexity of container stowage planning problems. *Discrete Applied Mathematics* **169** 225–230.

UNCTAD. 2010. Review of Maritime Transport 2010. United Nations Conference on Trade And Development.

UNCTAD. 2014. Review of Maritime Transport 2014. United Nations Conference on Trade And Development.

Vernimmen, B., W. Dullaert, S. Engelen. 2007. Schedule Unreliability in Liner Shipping: Origins and Consequences for the Hinterland Supply Chain. *Maritime Economics & Logistics* **9**(3) 193–213.

Visentini, M.S., D. Borenstein, J.Q. Li, P.B. Mirchandani. 2014. Review of real-time vehicle schedule recovery methods in transportation services. *Journal of Scheduling* **17**(6) 541–567.

Wang, S., Z. Liu, Q. Meng. 2015. Segment-based alteration for container liner shipping network design. *Transportation Research Part B* **72** 128–145.

Wang, S., Q. Meng. 2011. *International Maritime-Port Technology and Development Conference (MTEC 2011)*, chap. Liner Ship Routing and Scheduling: State of the Art. 69–74.

Wang, S., Q. Meng. 2012a. Liner ship route schedule design with sea contingency time and port time uncertainty. *Transportation Research Part B* **46**(5) 615–633.

Wang, S., Q. Meng. 2012b. Robust schedule design for liner shipping services. *Transportation Research Part E* **48**(6) 1093–1106.

Wang, S., Q. Meng. 2014. Liner shipping network design with deadlines. *Computers & Operations Research* **41** 140–149.

Wang, S., Q. Meng. 2015. Robust bunker management for liner shipping services. *European Journal of Operational Research* **243**(3) 789–797.

Wang, S., Q. Meng, Z. Liu. 2013. Bunker consumption optimization methods in shipping: A critical review and extensions. *Transportation Research Part E* **53** 49–62.

Wang, Y., G.T. Yeo, A.K.Y. Ng. 2014. Choosing optimal bunkering ports for liner shipping companies: A hybrid Fuzzy-Delphi-TOPSIS approach. *Transport Policy* **35** 358–365.

Wu, C.L. 2005. Inherent delays and operational reliability of airline schedules. *Journal of Air Transport Management* **11**(4) 273–282.

Wu, C.L., R.E. Caves. 2003. The punctuality performance of aircraft rotations in a network of airports. *Transportation Planning and Technology* **26**(5) 417–436.

Wu, Y., J. Tang, Y. Yu, Z. Pan. 2015. A stochastic optimization model for transit network timetable design to mitigate the randomness of traveling time by adding slack time. *Transportation Research Part C* **52** 15–31.

Xia, J., K.X. Li, H. Ma, Z. Xu. 2015. Joint Planning of Fleet Deployment, Speed Optimization, and Cargo Allocation for Liner Shipping. *Transportation Science* **49**(4) 922–938.

Yao, Z., S.H. Ng, L.H. Lee. 2012. A study on bunker fuel management for the shipping liner services. *Computers & Operations Research* **39**(5) 1160–1172.

Zhang, Y., C. Facanha. 2014. Strategic planning of empty container repositioning in the transpacific market: a case study. *International Journal of Logistics Research and Applications* **17**(5) 420–439.

Zheng, J., Q. Meng, Z. Sun. 2015. Liner hub-and-spoke shipping network design. *Transportation Research Part E* **75** 32–48.

Nederlandse Samenvatting (Summary in Dutch)

In dit proefschrift bestuderen we twee optimalisatieproblemen voor containerlijnschepen. Het eerste optimalisatieprobleem dat we bestuderen is het ontwerpen van een routenetwerk voor lijnschepen. Lijnscheepsnetwerken moeten voldoen aan een aantal specifieke aspecten, die we nu zullen bespreken. Lijnschepen varen cyclische routes; de eerste haven van een route gelijk is aan de laatste haven van die route. Over het algemeen wordt iedere haven op de route wekelijks op dezelfde dag bezocht, waardoor het aantal benodigde schepen op een route gelijk is aan de duur van de route in weken. Verder zijn er verschillende typen lijnschepen beschikbaar, maar niet ieder type schip is geschikt voor iedere route. De diepgang van een schip kan bijvoorbeeld groter zijn dan de diepgang in de haven, waardoor het schip de haven niet binnen kan varen. De schepen hebben ook allemaal een minimale en maximale snelheid waarmee gevaren kan worden. De snelheid heeft een groot effect op het brandstofverbruik, aangezien dit verbruik over het algemeen convex is in de snelheid. Onze resultaten laten een stijging in de winst van meer dan 20% zien in vergelijking met de netwerken in de huidige literatuur.

In het tweede optimalisatieprobleem in dit proefschrift bekijken we een gegeven route uit een scheepsnetwerk. De volgorde waarin de havens bezocht worden op deze route en de duur van de route staan al vast, maar de exacte aankomst- en vertrektijden in iedere haven mogen nog aangepast worden. Als schepen daadwerkelijk gaan varen op deze route, kunnen ze vertraging oplopen door allerlei onverwachte gebeurtenissen. Rederijen kunnen deze vertragingen proberen te beperken door extra tijd in te plannen om van de ene haven naar de andere te varen of door het schip harder te laten varen. In het tweede deel van dit proefschrift bepalen we zowel de aankomst- en vertrektijden voor iedere haven als de snelheid waarmee een schip moet gaan varen bij een gegeven vertraging in een bepaalde haven op zo'n manier dat de kosten van de rederij zo laag mogelijk zijn. We introduceren een exact en efficient oplossingsalgoritme voor convexe kostenstructuren.

Curriculum Vitae

Judith Mulder was born on July 22, 1988 in Delft. She received her bachelor's and master's degree from Erasmus University Rotterdam in Econometrics and Management Science, with a specialization in Quantitative Logistics and Operational Research. She also holds a bachelor's degree in Applied Mathematics from Delft University of Technology. In 2011, Judith started her PhD research at the Erasmus University Rotterdam. Her main research interest is operations research, in particular problems related to maritime transportation.

Judith has presented her research at various international conferences, such as EURO, IFORS, LOGMS, TSL Workshop, OR and ODYSSEUS. A paper on which the second chapter of her dissertation is based, has been published in the *European Journal of Operational Research*. The first chapter of this dissertation has been accepted for publication in a book on Ports and Networks.

Currently, Judith is a postdoctoral researcher at the Econometric Institute of the Erasmus University Rotterdam. She will among others work on the NWO-funded project 'ISOLA'.

Portfolio

Publications

Publications in journals:

Mulder, J. & Dekker, R. (2014). Methods for strategic liner shipping network design. *European Journal of Operational Research, 235* (2), 367-377.

Book chapters:

Mulder, J. & Dekker, R. (2016). Optimization in Container Lining Shipping. *Accepted.*

Working papers and reports:

Mulder, J., Dekker, R. & Sharifyazdi, M. (2012). Designing robust liner shipping schedules: Optimizing recovery actions and buffer times. *Econometric Institute Report EI 2012-30.*

Mulder, J. & Dekker, R. (2016). Will liner ships make fewer port calls per route? *Econometric Institute Report EI 2016-04.*

Mulder, J., van Jaarsveld, W.L. & Dekker, R. Joint optimization of speed and buffer times in transportation systems.

Teaching

Tutorial lecturer:

Combinatorisch Optimaliseren (Combinatorial Optimization), Erasmus School of Economics, Econometrics and Operations Research, 2012-2014.

Lineair Programmeren (Linear Programming), Erasmus School of Economics, Econometrics and Operations Research, 2013 & 2014.

Lecturer:

Management Science 3, Erasmus University Rotterdam, Maritime Economics and Logistics, 2014.

Programming, Erasmus Research Institute of Management, PhD course, 2015.

PhD courses

Networks and Polyhedra
Networks and Semidefinite Programming
Combinatorial Optimization 2a
Combinatorial Optimization 2b
Randomized Algorithms
Markov Decision Processes
Noncooperative Games
Convex Analysis for Optimization
Robust Optimization
Stochastic Dynamic Optimization
Statistical Methods
Publishing Strategy
English (CPE certificate)
EURO Summer Institute on “Maritime Logistics”

Conferences attended

ESE-Technion workshop 2011, Ein-Gedi, Israel
LNMB conference 2012, Lunteren, the Netherlands
ODYSSEUS 2012, Mykonos, Greece
LOGMS 2012, Bremen, Germany
Smart Port Poster session 2012, Rotterdam, the Netherlands
LNMB conference 2013, Lunteren, the Netherlands
TSL Workshop 2013, Asilomar, Californië, USA
Smart Port Poster session 2013, Rotterdam, the Netherlands
EURO 2013, Rome, Italy
OR 2013, Rotterdam, the Netherlands
LNMB conference 2014, Lunteren, the Netherlands
EIPC 2014, Rotterdam, the Netherlands
TSL Workshop 2014, Chicago, Illinois, USA
IFORS 2014, Barcelona, Spain
LOGMS 2014, Rotterdam, the Netherlands
Smart Port Poster session 2014, Rotterdam, the Netherlands
LNMB conference 2015, Lunteren, the Netherlands
ODYSSEUS 2015, Ajaccio, France
EURO 2015, Glasgow, Scotland
LNMB conference 2016, Lunteren, the Netherlands

ERASMUS RESEARCH INSTITUTE OF MANAGEMENT (ERIM)

ERIM PH.D. SERIES RESEARCH IN MANAGEMENT

The ERIM PhD Series contains PhD dissertations in the field of Research in Management defended at Erasmus University Rotterdam and supervised by senior researchers affiliated to the Erasmus Research Institute of Management (ERIM). All dissertations in the ERIM PhD Series are available in full text through the ERIM Electronic Series Portal: http://repub.eur.nl/pub. ERIM is the joint research institute of the Rotterdam School of Management (RSM) and the Erasmus School of Economics at the Erasmus University Rotterdam (EUR).

DISSERTATIONS LAST FIVE YEARS

Abbink, E.J., *Crew Management in Passenger Rail Transport*, Promotor(s): Prof.dr. L.G. Kroon & Prof.dr. A.P.M. Wagelmans, EPS-2014-325-LIS, http://repub.eur.nl/pub/76927

Acar, O.A., *Crowdsourcing for Innovation: Unpacking Motivational, Knowledge and Relational Mechanisms of Innovative Behavior in Crowdsourcing Platforms*, Promotor(s): Prof.dr.ir. J.C.M. van den Ende, EPS-2014-321-LIS, http://repub.eur.nl/pub/76076

Akin Ates, M., *Purchasing and Supply Management at the Purchase Category Level: strategy, structure and performance*, Promotor(s): Prof.dr. J.Y.F. Wynstra & Dr. E.M. van Raaij, EPS-2014-300-LIS, http://repub.eur.nl/pub/50283

Akpinar, E., *Consumer Information Sharing*, Promotor(s): Prof.dr.ir. A. Smidts, EPS-2013-297-MKT, http://repub.eur.nl/pub/50140

Alexander, L., *People, Politics, and Innovation: A Process Perspective*, Promotor(s): Prof.dr. H.G. Barkema & Prof.dr. D.L. van Knippenberg, EPS-2014-331-S&E, http://repub.eur.nl/pub/77209

Almeida e Santos Nogueira, R.J. de, *Conditional Density Models Integrating Fuzzy and Probabilistic Representations of Uncertainty*, Promotor(s): Prof.dr.ir. U. Kaymak & Prof.dr. J.M.C. Sousa, EPS-2014-310-LIS, http://repub.eur.nl/pub/51560

Bannouh, K., *Measuring and Forecasting Financial Market Volatility using High-frequency Data*, Promotor(s): Prof.dr. D.J.C. van Dijk, EPS-2013-273-F&A, http://repub.eur.nl/pub/38240

Ben-Menahem, S.M., *Strategic Timing and Proactiveness of Organizations*, Promotor(s): Prof.dr. H.W. Volberda & Prof.dr.ing. F.A.J. van den Bosch, EPS-2013-278-S&E, http://repub.eur.nl/pub/39128

Benning, T.M., *A Consumer Perspective on Flexibility in Health Care: Priority Access Pricing and Customized Care*, Promotor(s): Prof.dr.ir. B.G.C. Dellaert, EPS-2011-241-MKT, http://repub.eur.nl/pub/23670

Benschop, N, *Biases in Project Escalation: Names, frames & construal levels*, Promotors: Prof.dr. K.I.M. Rhode, Prof.dr. H.R. Commandeur, Prof.dr. M.Keil & Dr. A.L.P. Nuijten, EPS-2015-375-S&E, hdl.handle.net/1765/79408

Berg, W.E. van den, *Understanding Salesforce Behavior using Genetic Association Studies*, Promotor(s): Prof.dr. W.J.M.I. Verbeke, EPS-2014-311-MKT, http://repub.eur.nl/pub/51440

Betancourt, N.E., *Typical Atypicality: Formal and Informal Institutional Conformity, Deviance, and Dynamics*, Promotor(s): Prof.dr. B. Krug, EPS-2012-262-ORG, http://repub.eur.nl/pub/32345

Bliek, R. de, *Empirical Studies on the Economic Impact of Trust*, Promotor(s): Prof.dr. J. Veenman & Prof.dr. Ph.H.B.F. Franses, EPS-2015-324-ORG, http://repub.eur.nl/pub/78159

Blitz, D.C., *Benchmarking Benchmarks*, Promotor(s): Prof.dr. A.G.Z. Kemna & Prof.dr. W.F.C. Verschoor, EPS-2011-225-F&A, http://repub.eur.nl/pub/22624

Boons, M., *Working Together Alone in the Online Crowd: The Effects of Social Motivationsand Individual Knowledge Backgrounds on the Participation and Performance of Members of Online Crowdsourcing Platforms*, Promotor(s): Prof.dr. H.G. Barkema & Dr. D.A. Stam, EPS-2014-306-S&E, http://repub.eur.nl/pub/50711

Brazys, J., *Aggregated Marcoeconomic News and Price Discovery*, Promotor(s): Prof.dr. W.F.C. Verschoor, EPS-2015-351-F&A, http://repub.eur.nl/pub/78243

Burger, M.J., *Structure and Cooptition in Urban Networks*, Promotor(s): Prof.dr. G.A. van der Knaap & Prof.dr. H.R. Commandeur, EPS-2011-243-ORG, http://repub.eur.nl/pub/26178

Byington, E., *Exploring Coworker Relationships: Antecedents and Dimensions of Interpersonal Fit,Coworker Satisfaction, and Relational Models*, Promotor(s): Prof.dr. D.L. van Knippenberg, EPS-2013-292-ORG, http://repub.eur.nl/pub/41508

Camacho, N.M., *Health and Marketing: Essays on Physician and Patient Decision-Making*, Promotor(s): Prof.dr. S. Stremersch, EPS-2011-237-MKT, http://repub.eur.nl/pub/23604

Cancurtaran, P., *Essays on Accelerated Product Development*, Promotor(s): Prof.dr. F. Langerak & Prof.dr.ir. G.H. van Bruggen, EPS-2014-317-MKT, http://repub.eur.nl/pub/76074

Caron, E.A.M., *Explanation of Exceptional Values in Multi-dimensional Business Databases*, Promotor(s): Prof.dr.ir. H.A.M. Daniels & Prof.dr. G.W.J. Hendrikse, EPS-2013-296-LIS, http://repub.eur.nl/pub/50005

Carvalho, L. de, *Knowledge Locations in Cities: Emergence and Development Dynamics*, Promotor(s): Prof.dr. L. Berg, EPS-2013-274-S&E, http://repub.eur.nl/pub/38449

Consiglio, I., *Others: Essays on Interpersonal and Consumer Behavior*, Promotor: Prof.dr. S.M.J. van Osselaer, EPS-2016-366-MKT, http://repub.eur.nl/pub/79820

Cox, R.H.G.M., *To Own, To Finance, and To Insure - Residential Real Estate Revealed*, Promotor(s): Prof.dr. D. Brounen, EPS-2013-290-F&A, http://repub.eur.nl/pub/40964

Deichmann, D., *Idea Management: Perspectives from Leadership, Learning, and Network Theory*, Promotor(s): Prof.dr.ir. J.C.M. van den Ende, EPS-2012-255-ORG, http://repub.eur.nl/pub/31174

Deng, W., *Social Capital and Diversification of Cooperatives*, Promotor(s): Prof.dr. G.W.J. Hendrikse, EPS-2015-341-ORG, http://repub.eur.nl/pub/77449

Desmet, P.T.M., *In Money we Trust? Trust Repair and the Psychology of Financial Compensations*, Promotor(s): Prof.dr. D. de Cremer, EPS-2011-232-ORG, http://repub.eur.nl/pub/23268

Dollevoet, T.A.B., *Delay Management and Dispatching in Railways*, Promotor(s): Prof.dr. A.P.M. Wagelmans, EPS-2013-272-LIS, http://repub.eur.nl/pub/38241

Doorn, S. van, *Managing Entrepreneurial Orientation*, Promotor(s): Prof.dr. J.J.P. Jansen, Prof.dr.ing. F.A.J. van den Bosch, & Prof.dr. H.W. Volberda, EPS-2012-258-STR, http://repub.eur.nl/pub/32166

Douwens-Zonneveld, M.G., *Animal Spirits and Extreme Confidence: No Guts, No Glory?* Promotor(s): Prof.dr. W.F.C. Verschoor, EPS-2012-257-F&A, http://repub.eur.nl/pub/31914

Duca, E., *The Impact of Investor Demand on Security Offerings*, Promotor(s): Prof.dr. A. de Jong, EPS-2011-240-F&A, http://repub.eur.nl/pub/26041

Duyvesteyn, J.G. *Empirical Studies on Sovereign Fixed Income Markets,* Promotor(s): Prof.dr P.Verwijmeren & Prof.dr. M.P.E. Martens, EPS-2015-361-F&A, hdl.handle.net/1765/79033

Duursema, H., *Strategic Leadership: Moving Beyond the Leader-Follower Dyad*, Promotor(s): Prof.dr. R.J.M. van Tulder, EPS-2013-279-ORG, http://repub.eur.nl/pub/39129

Eck, N.J. van, *Methodological Advances in Bibliometric Mapping of Science*, Promotor(s): Prof.dr.ir. R. Dekker, EPS-2011-247-LIS, http://repub.eur.nl/pub/26509

Elemes, A, *Studies on Determinants and Consequences of Financial Reporting Quality,* Promotor: Prof.dr. E.Peek, EPS-2015-354-F&A, http://hdl.handle.net/1765/79037

Ellen, S. ter, *Measurement, Dynamics, and Implications of Heterogeneous Beliefs in Financial Markets*, Promotor(s): Prof.dr. W.F.C. Verschoor, EPS-2015-343-F&A, http://repub.eur.nl/pub/78191

Erlemann, C., *Gender and Leadership Aspiration: The Impact of the Organizational Environment*, Promotor: Prof.dr. D.L. van Knippenberg, EPS-2016-376-ORG, http://repub.eur.nl/pub/79409

Eskenazi, P.I., *The Accountable Animal*, Promotor(s): Prof.dr. F.G.H. Hartmann, EPS-2015-355-F&A, http://repub.eur.nl/pub/78300

Essen, M. van, *An Institution-Based View of Ownership*, Promotor(s): Prof.dr. J. van Oosterhout & Prof.dr. G.M.H. Mertens, EPS-2011-226-ORG, http://repub.eur.nl/pub/22643

Evangelidis, I., *Preference Construction under Prominence*, Promotor(s): Prof.dr. S.M.J. van Osselaer, EPS-2015-340-MKT, http://repub.eur.nl/pub/78202

Faber, N., *Structuring Warehouse Management*, Promotor(s): Prof.dr. MB.M. de Koster, Prof.dr. Ale Smidts, EPS-2015-336-LIS, http://repub.eur.nl/pub/78603

Fernald, K., *The Waves of Biotechnological Innovation in Medicine: Interfirm Cooperation Effects and a Venture Capital Perspective,* Promotor(s): Prof.dr. E.Claassen, Prof.dr. H.P.G.Pennings & Prof.dr. H.R. Commandeur, EPS-2015-371-S&E, http://hdl.handle.net/1765/79120

Fourne, S.P., *Managing Organizational Tensions: A Multi-Level Perspective on Exploration, Exploitation and Ambidexterity*, Promotor(s): Prof.dr. J.J.P. Jansen & Prof.dr. S.J. Magala, EPS-2014-318-S&E, http://repub.eur.nl/pub/76075

Gharehgozli, A.H., *Developing New Methods for Efficient Container Stacking Operations*, Promotor(s): Prof.dr.ir. M.B.M. de Koster, EPS-2012-269-LIS, http://repub.eur.nl/pub/37779

Gils, S. van, *Morality in Interactions: On the Display of Moral Behavior by Leaders and Employees*, Promotor(s): Prof.dr. D.L. van Knippenberg, EPS-2012-270-ORG, http://repub.eur.nl/pub/38027

Ginkel-Bieshaar, M.N.G. van, *The Impact of Abstract versus Concrete Product Communications on Consumer Decision-making Processes*, Promotor(s): Prof.dr.ir. B.G.C. Dellaert, EPS-2012-256-MKT, http://repub.eur.nl/pub/31913

Gkougkousi, X., *Empirical Studies in Financial Accounting*, Promotor(s): Prof.dr. G.M.H. Mertens & Prof.dr. E. Peek, EPS-2012-264-F&A, http://repub.eur.nl/pub/37170

Glorie, K.M., *Clearing Barter Exchange Markets: Kidney Exchange and Beyond*, Promotor(s): Prof.dr. A.P.M. Wagelmans & Prof.dr. J.J. van de Klundert, EPS-2014-329-LIS, http://repub.eur.nl/pub/77183

Hekimoglu, M., *Spare Parts Management of Aging Capital Products,* Promotor: Prof.dr.ir. R. Dekker, EPS-2015-368-LIS, http://hdl.handle.net/1765/79092

Heij, C.V., *Innovating beyond Technology. Studies on how management innovation, co-creation and business model innovation contribute to firm's (innovation) performance*, Promotor(s): Prof.dr.ing. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2012-370-STR, http://repub.eur.nl/pub/78651

Heyde Fernandes, D. von der, *The Functions and Dysfunctions of Reminders*, Promotor(s): Prof.dr. S.M.J. van Osselaer, EPS-2013-295-MKT, http://repub.eur.nl/pub/41514

Heyden, M.L.M., *Essays on Upper Echelons & Strategic Renewal: A Multilevel Contingency Approach*, Promotor(s): Prof.dr.ing. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2012-259-STR, http://repub.eur.nl/pub/32167

Hoever, I.J., *Diversity and Creativity*, Promotor(s): Prof.dr. D.L. van Knippenberg, EPS-2012-267-ORG, http://repub.eur.nl/pub/37392

Hogenboom, A.C., *Sentiment Analysis of Text Guided by Semantics and Structure,* Promotor(s):Prof.dr.ir. U.Kaymak & Prof.dr. F.M.G. de Jong, EPS-2015-369-LIS, http://hdl.handle.net/1765/79034

Hogenboom, F.P., *Automated Detection of Financial Events in News Text*, Promotor(s): Prof.dr.ir. U. Kaymak & Prof.dr. F.M.G. de Jong, EPS-2014-326-LIS, http://repub.eur.nl/pub/77237

Hollen, R.M.A.*, Exploratory Studies into Strategies to Enhance Innovation-Driven International Competitiveness in a Port Context: Toward Ambidextrous Ports,* Promotor(s) Prof.dr.ing. F.A.J. Van Den Bosch & Prof.dr. H.W.Volberda, EPS-2015-372-S&E, hdl.handle.net/1765/78881

Hoogendoorn, B., *Social Entrepreneurship in the Modern Economy: Warm Glow, Cold Feet*, Promotor(s): Prof.dr. H.P.G. Pennings & Prof.dr. A.R. Thurik, EPS-2011-246-STR, http://repub.eur.nl/pub/26447

Hoogervorst, N., *On The Psychology of Displaying Ethical Leadership: A Behavioral Ethics Approach*, Promotor(s): Prof.dr. D. de Cremer & Dr. M. van Dijke, EPS-2011-244-ORG, http://repub.eur.nl/pub/26228

Hout, D.H. van, *Measuring Meaningful Differences: Sensory Testing Based Decision Making in an Industrial Context; Applications of Signal Detection Theory and Thurstonian Modelling*, Promotor(s): Prof.dr. P.J.F. Groenen & Prof.dr. G.B. Dijksterhuis, EPS-2014-304-MKT, http://repub.eur.nl/pub/50387

Houwelingen, G.G. van, *Something To Rely On*, Promotor(s): Prof.dr. D. de Cremer & Prof.dr. M.H. van Dijke, EPS-2014-335-ORG, http://repub.eur.nl/pub/77320

Hurk, E. van der, *Passengers, Information, and Disruptions*, Promotor(s): Prof.dr. L.G. Kroon & Prof.mr.dr. P.H.M. Vervest, EPS-2015-345-LIS, http://repub.eur.nl/pub/78275

Hytonen, K.A., *Context Effects in Valuation, Judgment and Choice: A Neuroscientific Approach*, Promotor(s): Prof.dr.ir. A. Smidts, EPS-2011-252-MKT, http://repub.eur.nl/pub/30668

Iseger, P. den, *Fourier and Laplace Transform Inversion with Applications in Finance*, Promotor(s): Prof.dr.ir. R. Dekker, EPS-2014-322-LIS, http://repub.eur.nl/pub/76954

Jaarsveld, W.L. van, *Maintenance Centered Service Parts Inventory Control*, Promotor(s): Prof.dr.ir. R. Dekker, EPS-2013-288-LIS, http://repub.eur.nl/pub/39933

Jalil, M.N., *Customer Information Driven After Sales Service Management: Lessons from Spare Parts Logistics*, Promotor(s): Prof.dr. L.G. Kroon, EPS-2011-222-LIS, http://repub.eur.nl/pub/22156

Kappe, E.R., *The Effectiveness of Pharmaceutical Marketing*, Promotor(s): Prof.dr. S. Stremersch, EPS-2011-239-MKT, http://repub.eur.nl/pub/23610

Karreman, B., *Financial Services and Emerging Markets*, Promotor(s): Prof.dr. G.A. van der Knaap & Prof.dr. H.P.G. Pennings, EPS-2011-223-ORG, http://repub.eur.nl/pub/22280

Khanagha, S., *Dynamic Capabilities for Managing Emerging Technologies*, Promotor(s): Prof.dr. H.W. Volberda, EPS-2014-339-S&E, http://repub.eur.nl/pub/77319

Kil, J., *Acquisitions Through a Behavioral and Real Options Lens*, Promotor(s): Prof.dr. H.T.J. Smit, EPS-2013-298-F&A, http://repub.eur.nl/pub/50142

Klooster, E. van 't, *Travel to Learn: the Influence of Cultural Distance on Competence Development in Educational Travel*, Promotor(s): Prof.dr. F.M. Go & Prof.dr. P.J. van Baalen, EPS-2014-312-MKT, http://repub.eur.nl/pub/51462

Koendjbiharie, S.R., *The Information-Based View on Business Network Performance: Revealing the Performance of Interorganizational Networks*, Promotor(s): Prof.dr.ir. H.W.G.M. van Heck & Prof.mr.dr. P.H.M. Vervest, EPS-2014-315-LIS, http://repub.eur.nl/pub/51751

Koning, M., *The Financial Reporting Environment: The Role of the Media, Regulators and Auditors*, Promotor(s): Prof.dr. G.M.H. Mertens & Prof.dr. P.G.J. Roosenboom, EPS-2014-330-F&A, http://repub.eur.nl/pub/77154

Konter, D.J., *Crossing Borders with HRM: An Inquiry of the Influence of Contextual Differences in the Adoption and Effectiveness of HRM*, Promotor(s): Prof.dr. J. Paauwe & Dr. L.H. Hoeksema, EPS-2014-305-ORG, http://repub.eur.nl/pub/50388

Korkmaz, E., *Bridging Models and Business: Understanding Heterogeneity in Hidden Drivers of Customer Purchase Behavior*, Promotor(s): Prof.dr. S.L. van de Velde & Prof.dr. D. Fok, EPS-2014-316-LIS, http://repub.eur.nl/pub/76008

Kroezen, J.J., *The Renewal of Mature Industries: An Examination of the Revival of the Dutch Beer Brewing Industry*, Promotor(s): Prof.dr. P.P.M.A.R. Heugens, EPS-2014-333-S&E, http://repub.eur.nl/pub/77042

Kysucky, V., *Access to Finance in a Cros-Country Context*, Promotor(s): Prof.dr. L. Norden, EPS-2015-350-F&A, http://repub.eur.nl/pub/78225

Lam, K.Y., *Reliability and Rankings*, Promotor(s): Prof.dr. Ph.H.B.F. Franses, EPS-2011-230-MKT, http://repub.eur.nl/pub/22977

Lander, M.W., *Profits or Professionalism? On Designing Professional Service Firms*, Promotor(s): Prof.dr. J. van Oosterhout & Prof.dr. P.P.M.A.R. Heugens, EPS-2012-253-ORG, http://repub.eur.nl/pub/30682

Langhe, B. de, *Contingencies: Learning Numerical and Emotional Associations in an Uncertain World*, Promotor(s): Prof.dr.ir. B. Wierenga & Prof.dr. S.M.J. van Osselaer, EPS-2011-236-MKT, http://repub.eur.nl/pub/23504

Lee, C.I.S.G, *Big Data in Management Research: Exploring New Avenues*, Promotor(s): Prof.dr. S.J. Magala & Dr W.A. Felps, EPS-2016-365-ORG, http://repub.eur.nl/pub/79818

Legault-Tremblay, P.O., Corporate Governance During Market Transition: Heterogeneous responses to Institution Tensions in China, Promotor: Prof.dr. B. Krug, EPS-2015-362-ORG, http://repub.eur.nl/pub/78649

Lenoir, A.S. *Are You Talking to Me? Addressing Consumers in a Globalised World,* Promotor(s) Prof.dr. S. Puntoni & Prof.dr. S.M.J. van Osselaer, EPS-2015-363-MKT, http://hdl.handle.net/1765/79036

Leunissen, J.M., *All Apologies: On the Willingness of Perpetrators to Apologize*, Promotor(s): Prof.dr. D. de Cremer & Dr. M. van Dijke, EPS-2014-301-ORG, http://repub.eur.nl/pub/50318

Li, D., *Supply Chain Contracting for After-sales Service and Product Support*, Promotor(s): Prof.dr.ir. M.B.M. de Koster, EPS-2015-347-LIS, http://repub.eur.nl/pub/78526

Li, Z., *Irrationality: What, Why and How*, Promotor(s): Prof.dr. H. Bleichrodt, Prof.dr. P.P. Wakker, & Prof.dr. K.I.M. Rohde, EPS-2014-338-MKT, http://repub.eur.nl/pub/77205

Liang, Q.X., *Governance, CEO Identity, and Quality Provision of Farmer Cooperatives*, Promotor(s): Prof.dr. G.W.J. Hendrikse, EPS-2013-281-ORG, http://repub.eur.nl/pub/39253

Liket, K., *Why 'Doing Good' is not Good Enough: Essays on Social Impact Measurement*, Promotor(s): Prof.dr. H.R. Commandeur & Dr. K.E.H. Maas, EPS-2014-307-STR, http://repub.eur.nl/pub/51130

Loos, M.J.H.M. van der, *Molecular Genetics and Hormones: New Frontiers in Entrepreneurship Research*, Promotor(s): Prof.dr. A.R. Thurik, Prof.dr. P.J.F. Groenen, & Prof.dr. A. Hofman, EPS-2013-287-S&E, http://repub.eur.nl/pub/40081

Lovric, M., *Behavioral Finance and Agent-Based Artificial Markets*, Promotor(s): Prof.dr. J. Spronk & Prof.dr.ir. U. Kaymak, EPS-2011-229-F&A, http://repub.eur.nl/pub/22814

Lu, Y., *Data-Driven Decision Making in Auction Markets*, Promotor(s): Prof.dr.ir. H.W.G.M. van Heck & Prof.dr. W. Ketter, EPS-2014-314-LIS, http://repub.eur.nl/pub/51543

Manders, B., *Implementation and Impact of ISO 9001*, Promotor(s): Prof.dr. K. Blind, EPS-2014-337-LIS, http://repub.eur.nl/pub/77412

Markwat, T.D., *Extreme Dependence in Asset Markets Around the Globe*, Promotor(s): Prof.dr. D.J.C. van Dijk, EPS-2011-227-F&A, http://repub.eur.nl/pub/22744

Mees, H., *Changing Fortunes: How China's Boom Caused the Financial Crisis*, Promotor(s): Prof.dr. Ph.H.B.F. Franses, EPS-2012-266-MKT, http://repub.eur.nl/pub/34930

Mell, J.N., *Connecting Minds: On The Role of Metaknowledge in Knowledge Coordination*, Promotor: Prof.dr.D.L. van Knippenberg, EPS-2015-359-ORG, http://hdl.handle.net/1765/78951

Meuer, J., *Configurations of Inter-firm Relations in Management Innovation: A Study in*

China's Biopharmaceutical Industry, Promotor(s): Prof.dr. B. Krug, EPS-2011-228-ORG, http://repub.eur.nl/pub/22745

Micheli, M.R., *Business Model Innovation: A Journey across Managers' Attention and Inter-Organizational Networks*, Promotor(s): Prof.dr. J.J.P. Jansen, EPS-2015-344-S&E, http://repub.eur.nl/pub/78241

Mihalache, O.R., *Stimulating Firm Innovativeness: Probing the Interrelations between Managerial and Organizational Determinants*, Promotor(s): Prof.dr. J.J.P. Jansen, Prof.dr.ing. F.A.J. van den Bosch, & Prof.dr. H.W. Volberda, EPS-2012-260-S&E, http://repub.eur.nl/pub/32343

Milea, V., *News Analytics for Financial Decision Support*, Promotor(s): Prof.dr.ir. U. Kaymak, EPS-2013-275-LIS, http://repub.eur.nl/pub/38673

Naumovska, I., *Socially Situated Financial Markets: A Neo-Behavioral Perspective on Firms, Investors and Practices*, Promotor(s): Prof.dr. P.P.M.A.R. Heugens & Prof.dr. A. de Jong, EPS-2014-319-S&E, http://repub.eur.nl/pub/76084

Nielsen, L.K., *Rolling Stock Rescheduling in Passenger Railways: Applications in short term planning and in disruption management*, Promotor(s): Prof.dr. L.G. Kroon, EPS-2011-224-LIS, http://repub.eur.nl/pub/22444

Nuijten, A.L.P., *Deaf Effect for Risk Warnings: A Causal Examination applied to Information Systems Projects*, Promotor(s): Prof.dr. G.J. van der Pijl, Prof.dr. H.R. Commandeur & Prof.dr. M. Keil, EPS-2012-263-S&E, http://repub.eur.nl/pub/34928

Osadchiy, S.E., *The Dynamics of Formal Organization: Essays on bureaucracy and formal rules*, Promotor(s): Prof.dr. P.P.M.A.R. Heugens, EPS-2011-231-ORG, http://repub.eur.nl/pub/23250

Ozdemir, M.N., *Project-level Governance, Monetary Incentives, and Performance in Strategic R&D Alliances*, Promotor(s): Prof.dr.ir. J.C.M. van den Ende, EPS-2011-235-LIS, http://repub.eur.nl/pub/23550

Peers, Y., *Econometric Advances in Diffusion Models*, Promotor(s): Prof.dr. Ph.H.B.F. Franses, EPS-2011-251-MKT, http://repub.eur.nl/pub/30586

Peters, M., *Machine Learning Algorithms for Smart Electricity Markets*, Promotor(s): Prof.dr. W. Ketter, EPS-2014-332-LIS, http://repub.eur.nl/pub/77413

Porck, J., *No Team is an Island: An Integrative View of Strategic Consensus between Groups*, Promotor(s): Prof.dr. P.J.F. Groenen & Prof.dr. D.L. van Knippenberg, EPS-2013-299-ORG, http://repub.eur.nl/pub/50141

Porras Prado, M., *The Long and Short Side of Real Estate, Real Estate Stocks, and*

Equity, Promotor(s): Prof.dr. M.J.C.M. Verbeek, EPS-2012-254-F&A, http://repub.eur.nl/pub/30848

Poruthiyil, P.V., *Steering Through: How organizations negotiate permanent uncertainty and unresolvable choices*, Promotor(s): Prof.dr. P.P.M.A.R. Heugens & Prof.dr. S.J. Magala, EPS-2011-245-ORG, http://repub.eur.nl/pub/26392

Pourakbar, M., *End-of-Life Inventory Decisions of Service Parts*, Promotor(s): Prof.dr.ir. R. Dekker, EPS-2011-249-LIS, http://repub.eur.nl/pub/30584

Pronker, E.S., *Innovation Paradox in Vaccine Target Selection*, Promotor(s): Prof.dr. H.J.H.M. Claassen & Prof.dr. H.R. Commandeur, EPS-2013-282-S&E, http://repub.eur.nl/pub/39654

Protzner, S. *Mind the gap between demand and supply: A behavioral perspective on demand forecasting,* Promotor(s): Prof.dr. S.L. van de Velde & Dr. L. Rook, EPS-2015-364-LIS, *http*://repub.eur.nl/pub/79355

Pruijssers, J.K., *An Organizational Perspective on Auditor Conduct*, Promotor(s): Prof.dr. J. van Oosterhout & Prof.dr. P.P.M.A.R. Heugens, EPS-2015-342-S&E, http://repub.eur.nl/pub/78192

Retel Helmrich, M.J., *Green Lot-Sizing*, Promotor(s): Prof.dr. A.P.M. Wagelmans, EPS-2013-291-LIS, http://repub.eur.nl/pub/41330

Rietdijk, W.J.R. *The Use of Cognitive Factors for Explaining Entrepreneurship,* Promotor(s): Prof.dr. A.R. Thurik & Prof.dr. I.H.A. Franken, EPS-2015-356-S&E, http://repub.eur.nl/pub/79817

Rietveld, N., *Essays on the Intersection of Economics and Biology*, Promotor(s): Prof.dr. A.R. Thurik, Prof.dr. Ph.D. Koellinger, Prof.dr. P.J.F. Groenen, & Prof.dr. A. Hofman, EPS-2014-320-S&E, http://repub.eur.nl/pub/76907

Rijsenbilt, J.A., *CEO Narcissism: Measurement and Impact*, Promotor(s): Prof.dr. A.G.Z. Kemna & Prof.dr. H.R. Commandeur, EPS-2011-238-STR, http://repub.eur.nl/pub/23554

Rösch, D. *Market Efficiency and Liquidity,* Promotor: Prof.dr. M.A. van Dijk, EPS-2015-353-F&A, http://hdl.handle.net/1765/79121

Roza-van Vuren, M.W., *The Relationship between Offshoring Strategies and Firm Performance: Impact of innovation, absorptive capacity and firm size*, Promotor(s): Prof.dr. H.W. Volberda & Prof.dr.ing. F.A.J. van den Bosch, EPS-2011-214-STR, http://repub.eur.nl/pub/22155

Rubbaniy, G., *Investment Behaviour of Institutional Investors*, Promotor(s): Prof.dr. W.F.C. Verschoor, EPS-2013-284-F&A, http://repub.eur.nl/pub/40068

Schoonees, P. *Methods for Modelling Response Styles*, Promotor: Prof.dr P.J.F. Groenen, EPS-2015-348-MKT, http://repub.eur.nl/pub/79327

Shahzad, K., *Credit Rating Agencies, Financial Regulations and the Capital Markets*, Promotor(s): Prof.dr. G.M.H. Mertens, EPS-2013-283-F&A, http://repub.eur.nl/pub/39655

Sousa, M.J.C. de, *Servant Leadership to the Test: New Perspectives and Insights*, Promotor(s): Prof.dr. D.L. van Knippenberg & Dr. D. van Dierendonck, EPS-2014-313-ORG, http://repub.eur.nl/pub/51537

Spliet, R., *Vehicle Routing with Uncertain Demand*, Promotor(s): Prof.dr.ir. R. Dekker, EPS-2013-293-LIS, http://repub.eur.nl/pub/41513

Staadt, J.L., *Leading Public Housing Organisation in a Problematic Situation: A Critical Soft Systems Methodology Approach*, Promotor(s): Prof.dr. S.J. Magala, EPS-2014-308-ORG, http://repub.eur.nl/pub/50712

Stallen, M., *Social Context Effects on Decision-Making: A Neurobiological Approach*, Promotor(s): Prof.dr.ir. A. Smidts, EPS-2013-285-MKT, http://repub.eur.nl/pub/39931

Tarakci, M., *Behavioral Strategy: Strategic Consensus, Power and Networks*, Promotor(s): Prof.dr. D.L. van Knippenberg & Prof.dr. P.J.F. Groenen, EPS-2013-280-ORG, http://repub.eur.nl/pub/39130

Teixeira de Vasconcelos, M., *Agency Costs, Firm Value, and Corporate Investment*, Promotor(s): Prof.dr. P.G.J. Roosenboom, EPS-2012-265-F&A, http://repub.eur.nl/pub/37265

Troster, C., *Nationality Heterogeneity and Interpersonal Relationships at Work*, Promotor(s): Prof.dr. D.L. van Knippenberg, EPS-2011-233-ORG, http://repub.eur.nl/pub/23298

Tsekouras, D., *No Pain No Gain: The Beneficial Role of Consumer Effort in Decision-Making*, Promotor(s): Prof.dr.ir. B.G.C. Dellaert, EPS-2012-268-MKT, http://repub.eur.nl/pub/37542

Tuijl, E. van, *Upgrading across Organisational and Geographical Configurations*, Promotor(s): Prof.dr. L. van den Berg, EPS-2015-349-S&E, http://repub.eur.nl/pub/78224

Tuncdogan, A., *Decision Making and Behavioral Strategy: The Role of Regulatory Focus in Corporate Innovation Processes*, Promotor(s): Prof.dr.ing. F.A.J. van den Bosch, Prof.dr. H.W. Volberda, & Prof.dr. T.J.M. Mom, EPS-2014-334-S&E, http://repub.eur.nl/pub/76978

Uijl, S. den, *The Emergence of De-facto Standards*, Promotor(s): Prof.dr. K. Blind, EPS-2014-328-LIS, http://repub.eur.nl/pub/77382

Vagias, D., *Liquidity, Investors and International Capital Markets*, Promotor(s): Prof.dr. M.A. van Dijk, EPS-2013-294-F&A, http://repub.eur.nl/pub/41511

Veelenturf, L.P., *Disruption Management in Passenger Railways: Models for Timetable, Rolling Stock and Crew Rescheduling*, Promotor(s): Prof.dr. L.G. Kroon, EPS-2014-327-LIS, http://repub.eur.nl/pub/77155

Venus, M., *Demystifying Visionary Leadership: In search of the essence of effective vision communication*, Promotor(s): Prof.dr. D.L. van Knippenberg, EPS-2013-289-ORG, http://repub.eur.nl/pub/40079

Vermeer, W., *Propagation in Networks:The impact of information processing at the actor level on system-wide propagation dynamics,* Promotor: Prof.mr.dr. P.H.M.Vervest, EPS-2015-373-LIS, http://repub.eur.nl/pub/79325

Visser, V.A., *Leader Affect and Leadership Effectiveness:How leader affective displays influence follower outcomes*, Promotor(s): Prof.dr. D.L. van Knippenberg, EPS-2013-286-ORG, http://repub.eur.nl/pub/40076

Vlam, A.J., *Customer First? The Relationship between Advisors and Consumers of Financial Products*, Promotor(s): Prof.dr. Ph.H.B.F. Franses, EPS-2011-250-MKT, http://repub.eur.nl/pub/30585

Vries, J. de, *Behavioral Operations in Logistics,* Promotor(s): Prof.dr M.B.M de Koster & Prof.dr. D.A. Stam, EPS-2015-374-LIS, http://repub.eur.nl/pub/79705

Waltman, L., *Computational and Game-Theoretic Approaches for Modeling Bounded Rationality*, Promotor(s): Prof.dr.ir. R. Dekker & Prof.dr.ir. U. Kaymak, EPS-2011-248-LIS, http://repub.eur.nl/pub/26564

Wang, T., *Essays in Banking and Corporate Finance*, Promotor(s): Prof.dr. L. Norden & Prof.dr. P.G.J. Roosenboom, EPS-2015-352-F&A, http://repub.eur.nl/pub/78301

Wang, Y., *Information Content of Mutual Fund Portfolio Disclosure*, Promotor(s): Prof.dr. M.J.C.M. Verbeek, EPS-2011-242-F&A, http://repub.eur.nl/pub/26066

Wang, Y., *Corporate Reputation Management: Reaching Out to Financial Stakeholders*, Promotor(s): Prof.dr. C.B.M. van Riel, EPS-2013-271-ORG, http://repub.eur.nl/pub/38675

Weenen, T.C., *On the Origin and Development of the Medical Nutrition Industry*, Promotor(s): Prof.dr. H.R. Commandeur & Prof.dr. H.J.H.M. Claassen, EPS-2014-309-S&E, http://repub.eur.nl/pub/51134

Wolfswinkel, M., *Corporate Governance, Firm Risk and Shareholder Value*, Promotor(s): Prof.dr. A. de Jong, EPS-2013-277-F&A, http://repub.eur.nl/pub/39127

Yang, S., *Information Aggregation Efficiency of Prediction Markets*, Promotor(s): Prof.dr.ir. H.W.G.M. van Heck, EPS-2014-323-LIS, http://repub.eur.nl/pub/77184

Zaerpour, N., *Efficient Management of Compact Storage Systems*, Promotor(s): Prof.dr.ir. M.B.M. de Koster, EPS-2013-276-LIS, http://repub.eur.nl/pub/38766

Zhang, D., *Essays in Executive Compensation*, Promotor(s): Prof.dr. I. Dittmann, EPS-2012-261-F&A, http://repub.eur.nl/pub/32344

Zwan, P.W. van der, *The Entrepreneurial Process: An International Analysis of Entry and Exit*, Promotor(s): Prof.dr. A.R. Thurik & Prof.dr. P.J.F. Groenen, EPS-2011-234-ORG, http://repub.eur.nl/pub/23422

Erasmus Research Institute of Management - ERiM

NETWORK DESIGN AND ROBUST SCHEDULING IN LINER SHIPPING

Liner ships sail according to a published timetable specifying the arrival and departure times in ports. Liner shipping networks consist of cyclic routes that are usually serviced once a week. In this way, customers know when they can pick up or deliver their containers to a port. Each route in the network is sailed using one or more ships with possibly different capacities. The network is used to satisfy the given container demand between ports. Liner companies obtain a revenue for each container that they transport between two ports. Furthermore, they incur costs for operating the network: fixed ship costs, bunker costs, (un)loading and transshipment costs and lost sales costs. The first part of this thesis proposes methods to design liner networks that maximize profit for the liner companies.

Ships will incur delays when executing a route, resulting in uncertain arrival and departure times. Delayed arrivals can be very costly for liner companies, hence they want their schedules to be reliable. The reliability of a route can be increased by including buffer time in the route, which can be used to capture a delay without incurring delay costs. Furthermore, ships can reduce delays by speeding up, which brings about additional costs. The second part of this thesis proposes methods to minimize total costs by reallocating buffer times and determining recovery policies specifying how to react on incurred delays.

Design & layout: B&T Ontwerp en advies (www.b-en-t.nl) Print: Haveka (www.haveka.nl)

ERiM

The Erasmus Research Institute of Management (ERIM) is the Research School (Onderzoekschool) in the field of management of the Erasmus University Rotterdam. The founding participants of ERIM are the Rotterdam School of Management, Erasmus University (RSM), and the Erasmus School of Economics (ESE). ERIM was founded in 1999 and is officially accredited by the Royal Netherlands Academy of Arts and Sciences (KNAW). The research undertaken by ERIM is focused on the management of the firm in its environment, its intra- and interfirm relations, and its business processes in their interdependent connections.

The objective of ERIM is to carry out first rate research in management, and to offer an advanced doctoral programme in Research in Management. Within ERIM, over three hundred senior researchers and PhD candidates are active in the different research programmes. From a variety of academic backgrounds and expertises, the ERIM community is united in striving for excellence and working at the forefront of creating new business knowledge.

ERIM PhD Series

Research in Management

Erasmus Research Institute of Management - ERiM
Rotterdam School of Management (RSM)
Erasmus School of Economics (ESE)
Erasmus University Rotterdam (EUR)
P.O. Box 1738, 3000 DR Rotterdam,
The Netherlands

Tel. +31 10 408 11 82
Fax +31 10 408 96 40
E-mail info@erim.eur.nl
Internet www.erim.eur.nl